

フロントエンドエンジニア 教材資料 1

令和5年度文部科学省委託「専修学校による地域産業中核的人材養成」事業

フロントエンドエンジニア 教材資料 1

目次

Web システム概論	1
はじめに	1
第 1 章 Web システムの構成	2
第 2 章 開発環境の構築手順	15
第 3 章 Web 技術の概要	39
UI (ユーザインターフェイス) 概論	47
はじめに	47
第 1 章 UI と UX	48
第 2 章 デザインプロセス	74
第 3 章 UI 部品	93
第 4 章 UI コンポーネント	103
HTML/CSS	111
はじめに	111
第 1 章 HTML Living Standard	112
第 2 章 文章構造	121
第 3 章 CSS 3	160
第 4 章 CSS 3 の設定	177
プログラミング演習 (第 2 章)	219
プログラミング演習 (第 4 章)	241

フロントエンドエンジニア 養成講座

Webシステム概論

1

はじめに

【学習目標】

本テキストでは、Webシステムの全体像を理解することを目標とします。

目次

第1章	Webシステムの構成	003
第2章	開発環境の構築手順	028
第3章	Web技術の概要	076

2

Webシステム概論

第1章

Webシステムの構成

【本章学習内容】

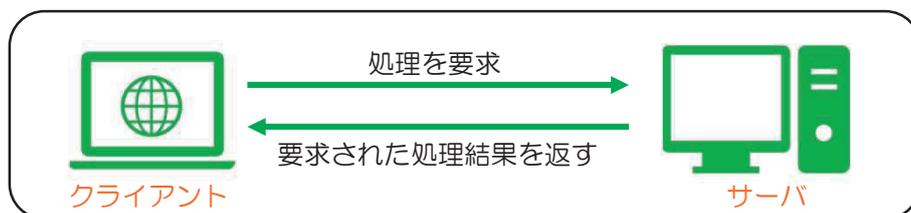
本章では、Webシステムの代表的な構成として「3層クライアントサーバシステム」について学習します。

3

第1章 Webシステムの構成

1.1 3層クライアントサーバシステム

1.1.1 クライアントサーバシステムとは



- ユーザは、クライアントから処理を要求
- サーバは、要求された処理を実行し、その結果をクライアントに返信
- ユーザは、結果を確認
- クライアントに高性能なコンピュータは不要

4

第1章 Webシステムの構成

1.1 3層クライアントサーバシステム

1.1.2 3層クライアントサーバシステムとは



- サーバを分けることで処理の負荷や通信の集中を防ぐことができる
- 現在のWebシステムの大多数はこの**3層クライアントサーバシステム**で動いている
- サーバが増えるので2層クライアントサーバシステムよりコストは高くなる

5

第1章 Webシステムの構成

1.1 3層クライアントサーバシステム

1.1.3 プレゼンテーション層 (クライアント)



クライアント
(プレゼンテーション層)

- クライアントはユーザが操作する端末
- 処理はサーバが行うため高い処理能力は不要
- Webシステムはユーザにとっての使いやすさが必須
- クライアントで表示し、ユーザが使用する画面を**ユーザの立場に立って設計**することがフロントエンドエンジニアの役割
- クライアントはPC以外にスマートフォン、タブレットを使用するケースが増加
- HTML/CSSで画面を作成し、JavaScriptで使い勝手を向上

6

第1章 Webシステムの構成

1.1 3層クライアントサーバシステム

1.1.4 ファンクション層（アプリケーションサーバ）



アプリケーションサーバ
(ファンクション層)

- Webプログラムが稼働するサーバ
- Java/PHP/Pythonなどのプログラム言語でサーバアプリケーションを構築
- アプリケーションサーバという専用のハードウェアがあるのではなく、サーバ上でWebアプリケーションを実行するためのミドルウェア（ソフトウェア）を実行
- 代表的なアプリケーションサーバは「Apache」「Tomcat」等
- クライアントから受け取ったユーザからの要求を処理し、必要に応じてデータベースサーバにアクセス要求を行いデータ処理を実行
- アプリケーションサーバで稼働するプログラムはバックエンドエンジニアが開発

7

第1章 Webシステムの構成

1.1 3層クライアントサーバシステム

1.1.5 データ層（データベースサーバ）



データベースサーバ
(データ層)

- データベースが稼働するサーバ
- Webシステムにとってデータベースの処理が最も負荷が高い
- アプリケーションサーバからデータベースの処理を独立させアプリケーションサーバの負荷を軽減する
- データベースサーバではDBMS（Data Base Management System）が稼働

8

第1章 Webシステムの構成

1.2 通信プロトコル

1.2.1 通信プロトコルとは



- ネットワーク通信を行う上で様々なルールや規格・規約が存在しており、それらの総称が通信プロトコル
- 通信する機器同士が同じルールに従ってやり取りしなければならない
- 通信の種類により、使用できるプロトコルは異なる
- インターネット上では世界中の通信機器が同じプロトコルを使用しているので異なる国同士でも簡単に通信可能
- プロトコルが異なる通信機器同士で通信を行う場合、プロトコルコンバータというプロトコル変換のための機器が必要

第1章 Webシステムの構成

1.2 通信プロトコル

1.2.2 HTTP



- Hyper Text Transfer Protocolの略
- WebサーバとWebブラウザの間でデータの送受信を行うために用いるプロトコル
- Webページを構成するために必要なHTMLファイル、スタイルシート (CSS)、スクリプト、画像、動画などのファイルをやり取り
- もともとはWeb用ではなく、研究用の情報を研究の関係者間で共有するシステムが始まり

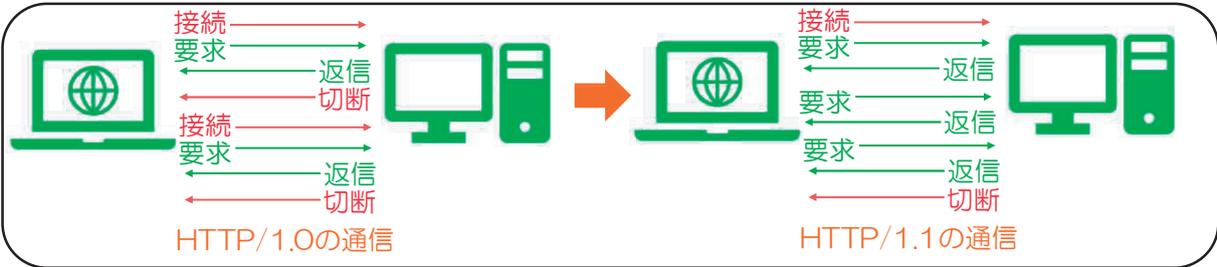
1996年	HTTP/1.0
1997年	HTTP/1.1
2015年	HTTP/2
2022年	HTTP/3

HTTPバージョンの推移

第1章 Webシステムの構成

1.2 通信プロトコル

1.2.3 HTTP/1.1



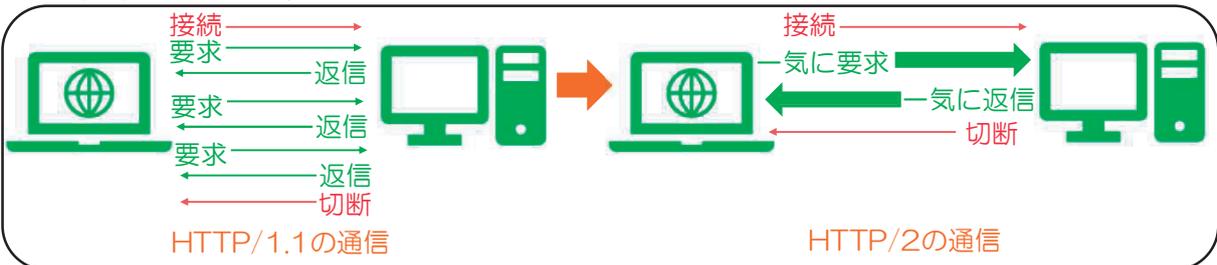
- 不明確な箇所の多かったHTTP/1.0を改良し明確化
- 1セッションごとにサーバ接続を切断する仕様を改め、サーバ接続を維持したまま通信が通信効率が格段に向上
- 拡張性の高いプロトコルで、その後15年以上拡張されながら継続使用
- Webページの見た目や動きがよりリッチ(豪華)になってきたことにより、ページの表示に必要な通信の数が増えたことで問題が発生

11

第1章 Webシステムの構成

1.2 通信プロトコル

1.2.4 HTTP/2



- 同一のサーバ接続で複数のHTTPリクエストを並行して扱うことができ、HTTP/1.1の問題点を解決
- HTTP/1.1の上で動く追加プロトコル
- Googleが中心になって開発したSPDYというプロトコルがベース
- テキスト形式での送受信ではなくバイナリ形式での送受信になり処理効率が更に向上

12

第1章 Webシステムの構成

1.2 通信プロトコル

1.2.5 HTTPS (HTTP Secure)

- HTTPとHTTPSの違いは通信が暗号化されているかどうか
- HTTP/2の通信はHTTPS必須
- TLSプロトコルを使用し「通信内容の暗号化」「通信相手の確認」「通信内容の改ざん防止」を実装
- サーバ認証（サーバ証明書）とクライアント認証（クライアント証明書）を用いて接続認証を行い、許可されていない通信を遮断



13

第1章 Webシステムの構成

1.2 通信プロトコル

1.2.6 サーバ認証（サーバ証明書）



- サーバ証明書という電子証明書を用いて「通信の暗号化」「実在証明」を実施
- サーバ証明書は「認証局」と呼ばれる組織が発行
- 発行時に申請者の情報を調査し、問題が無ければ認証局によって発行されるためWebサイトの訪問者に認証局によって認証されていることを証明
- サーバ証明書を発行してもらうためには認証局が提供しているサービスを契約
- TLSプロトコルを用いてWebサーバとWebクライアント間のデータの送受信を暗号化

14

第1章 Webシステムの構成

1.2 通信プロトコル

1.2.7 クライアント認証（クライアント証明書）



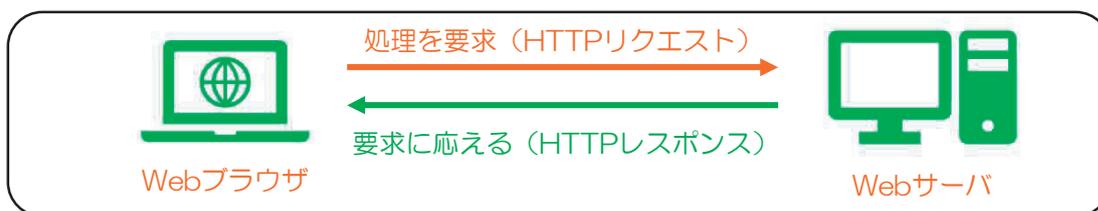
- クライアント証明書という電子証明書を用いてクライアント（ユーザ）の信用を保証
- クライアント認証をおこなうとネットワークやサーバに接続できるクライアントを制限できるため、万が一ID/パスワードが漏洩してもクライアント証明書をインストールしたクライアントでなければ接続不可
- クライアント証明書には有効期限が存在するため盗難・紛失の際に証明書を失効させれば同じクライアントでも接続不可
- 証明書を失効させても元の端末が手元にありID/パスワードを知っていれば証明書の再発行ができるシステムが多く、端末とID/パスワードの取り扱いは厳重注意

15

第1章 Webシステムの構成

1.2 通信プロトコル

1.2.8 HTTPリクエスト



- Webクライアント（Webブラウザ）からWebサーバへ要求を伝えるメッセージ
- Webサーバはその要求に応じてファイルをWebクライアントに送信
- HTTPリクエストメソッドは、主に「GETメソッド」「POSTメソッド」を利用

16

第1章 Webシステムの構成

1.2 通信プロトコル

1.2.9 GETメソッド



- WebサーバにHTTPリクエストヘッダ経由で情報 (パラメータ) を渡すHTTPリクエスト
- 「<https://www.example.co.jp?input=123>」のように【?パラメータ名=値】をURLの末尾につけて情報 (パラメータ) をWebサーバに渡すHTTPリクエスト
- 送信するパラメータがURLに残るためセキュリティを考慮しなければならない情報の送信は不可

17

第1章 Webシステムの構成

1.2 通信プロトコル

1.2.10 POSTメソッド



- WebサーバにHTTPリクエストボディ経由で情報 (パラメータ) を渡すHTTPリクエスト
- URLに情報 (パラメータ) が付加されないためセキュリティを考慮する情報の送信に不可欠

18

第1章 Webシステムの構成

1.2 通信プロトコル

1.2.1.1 HTTPレスポンス



- Webクライアント (Webブラウザ) が出した要求 (HTTPリクエスト) に対するWebサーバからの返答
- 返答の種類を表すステータスコードが付属
- HTTPレスポンスヘッダはHTTPレスポンスで送られてくるデータの種類の各種情報
- HTTPレスポンスボディはHTTPレスポンスヘッダで指定された種類のデータ

第1章 Webシステムの構成

1.2 通信プロトコル

1.2.1.2 複数のファイルで構成されたWebページのHTTPレスポンス



第1章 Webシステムの構成

1.2 通信プロトコル

1.2.1.3 HTTPレスポンスのステータスコード



- 主なステータスコードの種類は次の通り

200	OK	正常処理
401	Unauthorized	認証失敗
403	Forbidden	アクセス権限無し
404	Not Found	HTTPリクエストしたURLのページが存在せず
500	Internal Server Error	プログラム等でエラーが発生
503	Service Unavailable	Webサーバに負荷がかかりすぎて応答無し
504	Gateway Timeout	レスポンスが制限時間内に得られない

21

第1章 Webシステムの構成

1.3 Webアプリケーションの構成

1.3.1 Webライブラリ



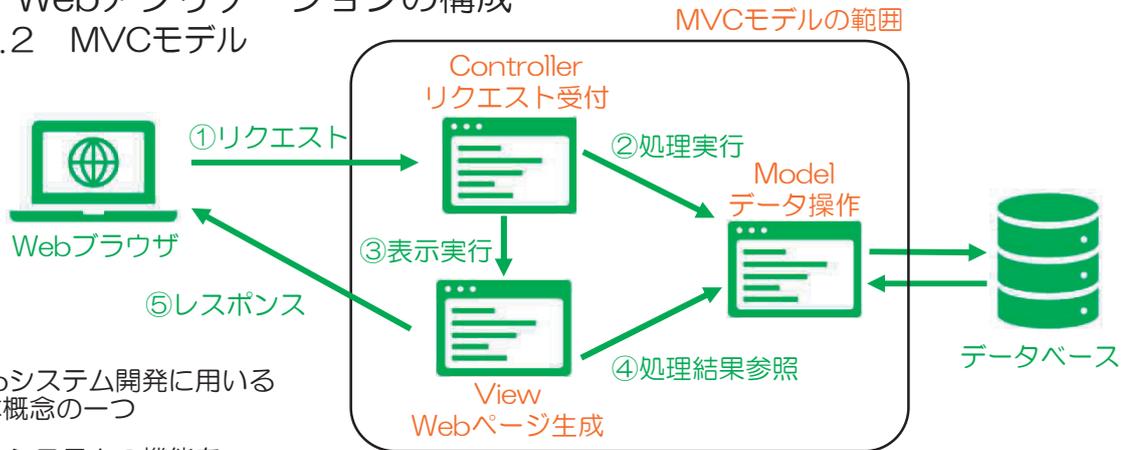
- Webアプリケーションの開発において、**開発効率を向上**
- 共通した作業に必要なものがあらかじめ用意されており、その分の**時間・労力を軽減**
- 大規模なプロジェクトだけでなく、規模に関係なくWebアプリケーション開発で活用
- 複数のエンジニアで同時開発する際の、各エンジニアの癖によるプログラムの**バグやミスの発生を軽減**
- Webライブラリは世界中で開発されており、非常に高機能なものも存在
- **開発に必要なWebライブラリを見極めて採用することが重要**

22

第1章 Webシステムの構成

1.3 Webアプリケーションの構成

1.3.2 MVCモデル

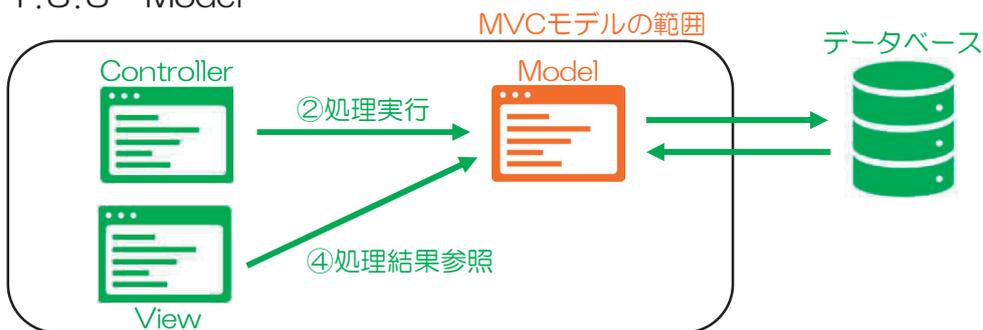


- Webシステム開発に用いる基本概念の一つ
- Webシステムの機能を「データと処理」「表示」「制御」に分割し実装
- Webアプリケーションの設計作業を整理し、どこに何を記述するか明確化

第1章 Webシステムの構成

1.3 Webアプリケーションの構成

1.3.3 Model

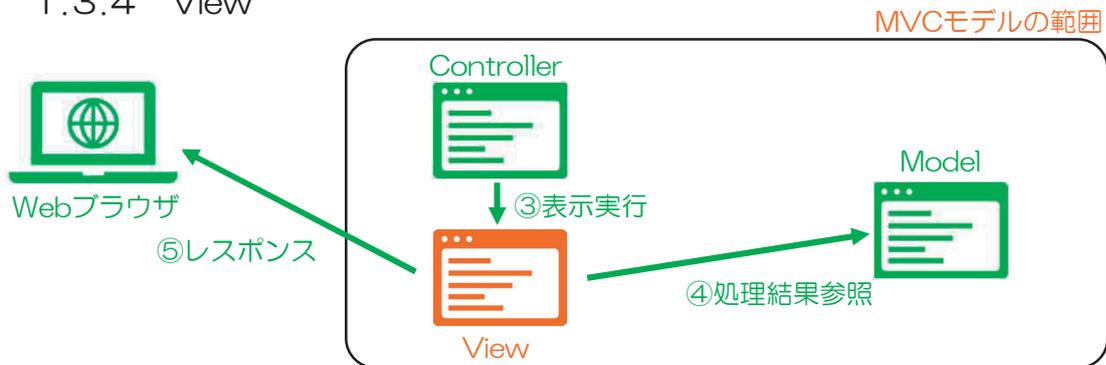


- データベースに関するプログラムを**一か所に集約**
- Controllerからのデータベースアクセス要求に応じてデータや処理結果を返却
- データベースの扱うデータ型を表示に適した形式に加工
- データベースの変更に伴う影響範囲を限定することで**メンテナンス性向上**

第1章 Webシステムの構成

1.3 Webアプリケーションの構成

1.3.4 View



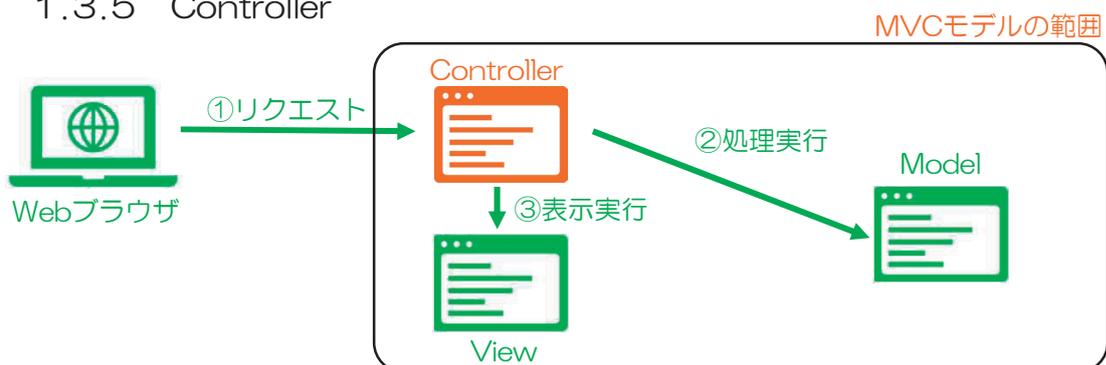
- Webブラウザに表示するHTMLファイルを生成する部分
- アクセスするたびに内容が変わる動的ページを生成
- フロントエンドエンジニアはHTML・CSS・JavaScriptを使用し動的ページを作成

25

第1章 Webシステムの構成

1.3 Webアプリケーションの構成

1.3.5 Controller



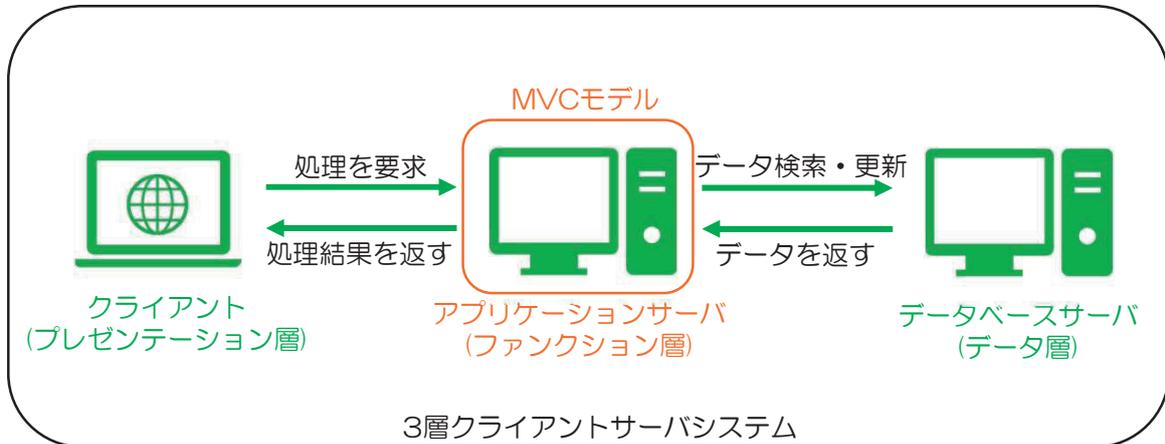
- Webブラウザからリクエストを受付
- Modelにデータベース操作の処理を依頼
- ViewにWebブラウザの表示内容を作成する処理を依頼

26

第1章 Webシステムの構成

1.3 Webアプリケーションの構成

1.3.6 3層クライアントサーバシステムとMVCモデル



Webシステム概論

第2章

開発環境の構築

【本章学習内容】

本章では「開発環境のインストール」「開発環境の使用方法」「Chromeのインストール」について学習します。

28

第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.1 開発環境とは

代表的なテキストエディタ（入力機能のみ）



VSCode



Sublime Text



Vim



サクラエディタ

代表的な統合開発環境（入力と実行機能）



Eclipse



Visual Studio



Xcode



Android Studio



Unity

29

第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.2 Eclipseとは



Eclipse

- Webアプリケーション開発に広く用いられる統合開発環境
- 主にJavaプログラムの開発に用いられるが、HTML・CSS・JavaScriptファイルの開発も可能
- プラグインのインストールで他の多くの言語にも対応可能
- 強力な入力支援機能を実装
- Webサーバ機能やデバッグ機能など便利な実行環境を実装
- Pleiadesプロジェクトにより日本語化されている
- 開発に必要なパッケージをまとめた「Pleiades All in One」が便利

30

第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.3 Pleiades All in Oneのダウンロード

- PleiadesプロジェクトのWebサイト (<https://mergedoc.osdn.jp/>) にアクセス
- Eclipse最新バージョンのボタンをクリック

Eclipseバージョン	付属 JDK バージョン							
	1.4	1.5	6	7	8	11	17	21
2023								
2022								
2021								
2020								

31

第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.3 Pleiades All in Oneのダウンロード

- 開発環境（Windows・Mac）に応じたJava列の「Download」ボタンをクリック



32

第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.3 Pleiades All in Oneのダウンロード

- 自動的にダウンロードが開始
- 自動でダウンロードが開始されない場合、画面中段のダウンロードリンクをクリック
- 大きく表示されている「Download Now」等の画像は広告なのでクリックしない



33

第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.4 Pleiades All in Oneのインストール (Windows)

- ダウンロード先は、特に変更していなければ「ダウンロード」フォルダ
- ダウンロードしたPleiadesインストーラをダブルクリック



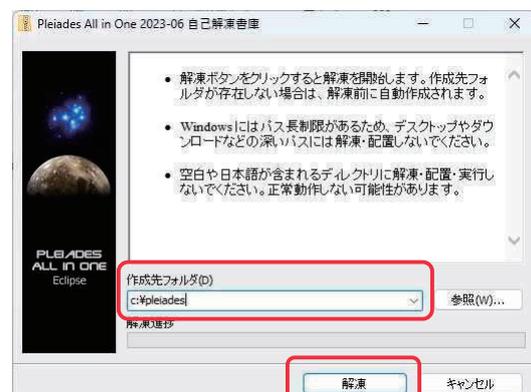
34

第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.4 Pleiades All in Oneのインストール (Windows)

- 作成先フォルダを「C:\pleiades」に変更
- 解凍ボタンをクリック



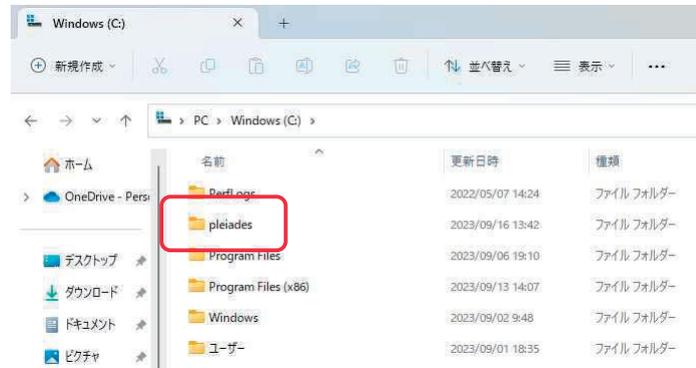
35

第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.4 Pleiades All in Oneのインストール (Windows)

- インストールが完了すると
Cドライブ直下に
「pleiades」フォルダがある



36

第2章 開発環境の構築

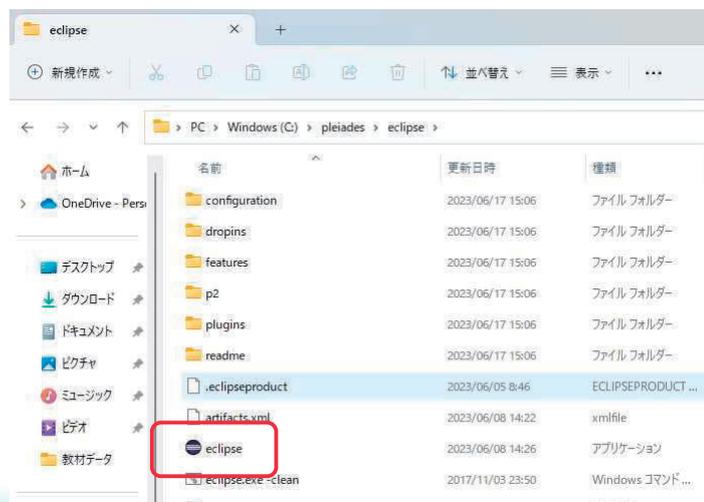
2.1 Webアプリケーション開発環境の構築

2.1.4 Pleiades All in Oneのインストール (Windows)

- C:\pleiades\eclipse\ eclipse.exeをダブルクリック
- 今後、使いやすいように
デスクトップ等にショート
カットを作成すると良い

※Windows10の場合
アイコンを右クリック→
送る→デスクトップ

※Windows11の場合
アイコンを右クリック→
その他のオプションを確認→
送る→デスクトップ



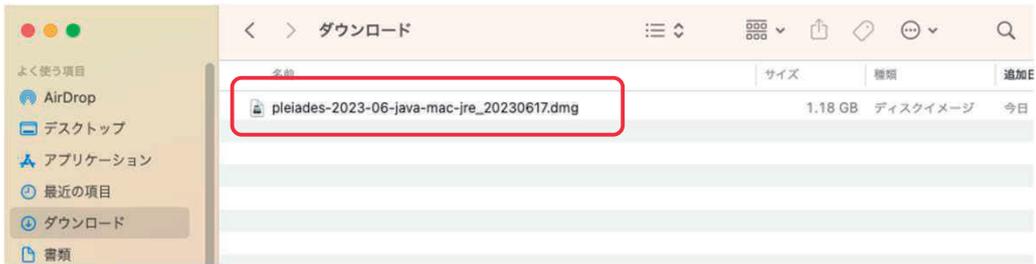
37

第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.5 Pleiades All in Oneのインストール (Mac)

- ダウンロード先は、特に変更していなければFinderの「ダウンロード」
- ダウンロードしたPleiadesインストーラをダブルクリック



38

第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.5 Pleiades All in Oneのインストール (Mac)

- インストーラが起動
「Eclipse_yyyy-mm.app」を
「Applications」へドラッグ



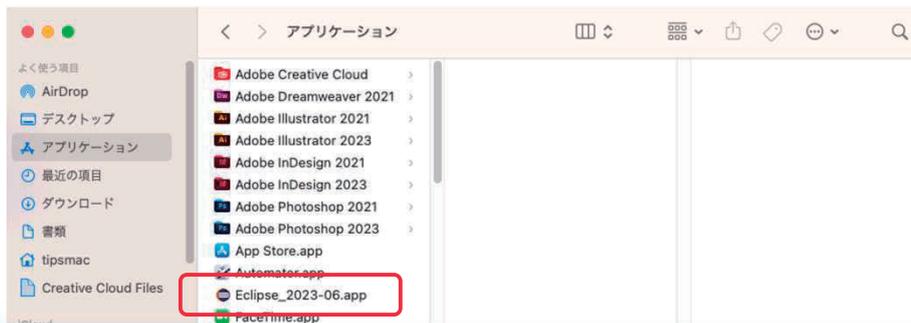
39

第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.5 Pleiades All in Oneのインストール (Mac)

- インストールが完了するとFinderの「アプリケーション」に追加される
- Eclipse_YYYY-MM.appをダブルクリック



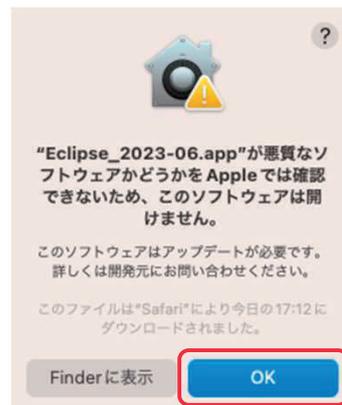
40

第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.5 Pleiades All in Oneのインストール (Mac/エラーの場合)

- ダブルクリックで起動時、右の図のようなエラーが表示される場合がある
- いったん「OK」をクリックしてこのウィンドウを非表示



41

第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.5 Pleiades All in Oneのインストール (Mac/エラーの場合)

- キーボードの「Option」キーを押しながらクリックしてメニューを表示し「開く」をクリック



42

第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.5 Pleiades All in Oneのインストール (Mac/エラーの場合)

- 「開く」をクリックをクリックするとEclipseが起動
- Eclipseはセキュリティ設定の例外として保存され、今後は普通に起動する



43

第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.6 Eclipseの起動と使い方

- Eclipseを起動するとワークスペースを選択する「Eclipse IDE ランチャー」が起動
- ワークスペース保存場所を変更する場合は「参照」ボタンをクリックし保存場所を指定
- 変更の必要が無ければそのまま「起動」をクリック



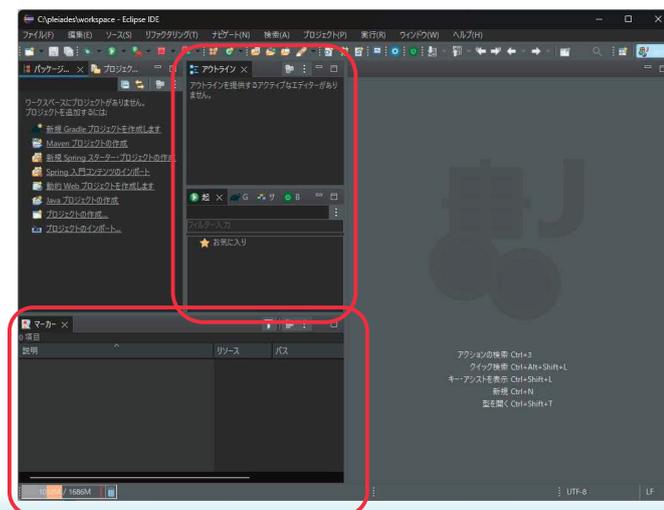
44

第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.6 Eclipseの起動と使い方

- Eclipseが起動
- 不要な表示エリアを非表示



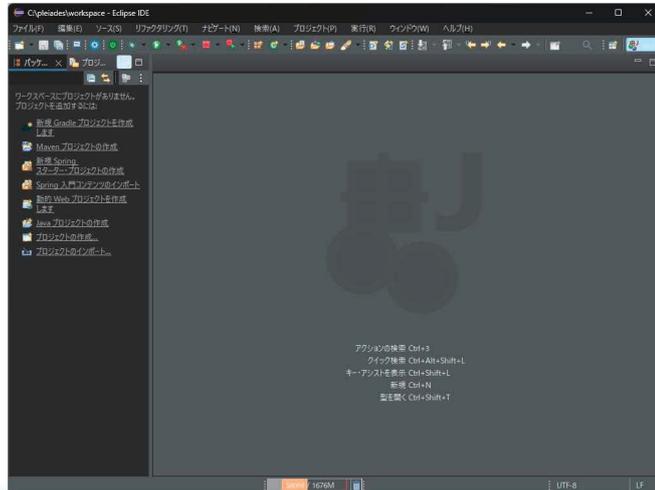
45

第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.6 Eclipseの起動と使い方

- 右の図が非表示にした状態



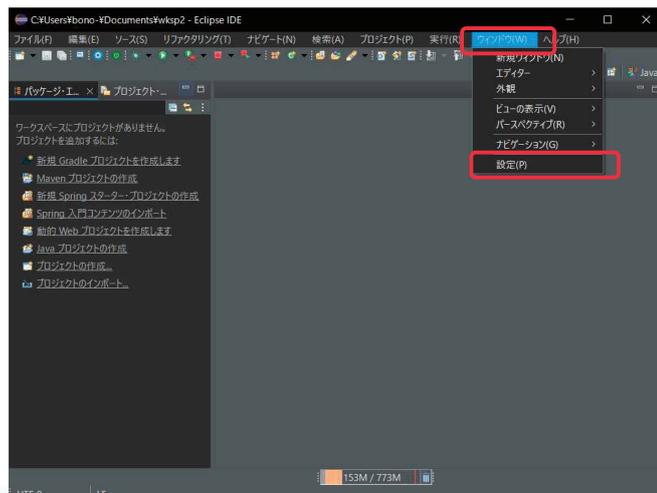
46

第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.6 Eclipseの起動と使い方

- Eclipseのデフォルトの配色は暗めの色（ダーク）
- 明るめの色に変更するにはメニューの「ウィンドウ」→「選択」をクリック



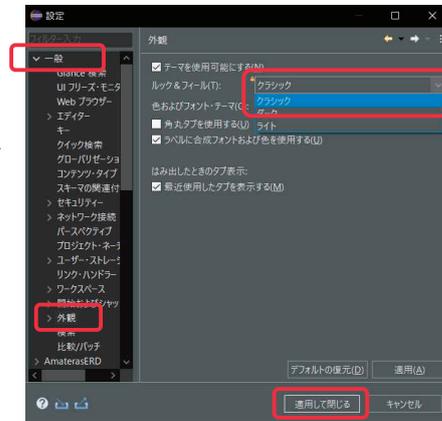
47

第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.6 Eclipseの起動と使い方

- 「一般」→「外観」をクリックし、外観メニューを表示
- 「ルック&フィール」から「クラシック」を選択
- 「色およびフォント・テーマ」から「クラシック・テーマ」を選択
- 「適用して閉じる」ボタンをクリック
- 「テーマの変更を完全に有効にするために再起動しますか」の確認ダイアログが表示された場合は、「再開」ボタンをクリック



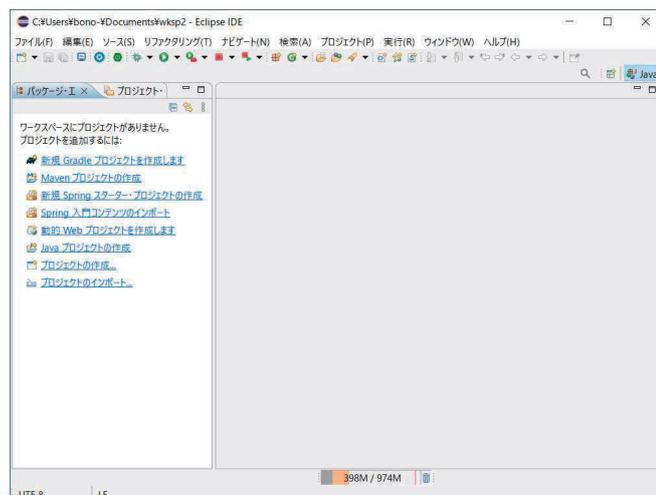
48

第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.6 Eclipseの起動と使い方

- Eclipse再起動後に配色が変更



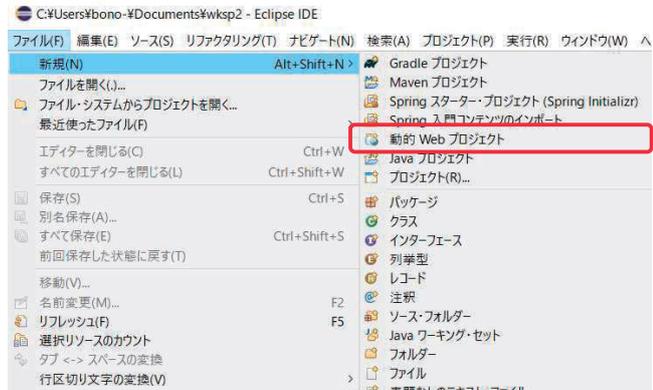
49

第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.6 Eclipseの起動と使い方

- Webアプリケーションはプロジェクト単位で管理
- Webアプリケーションの作成時はメニューの「ファイル」→「新規」→「動的Webプロジェクト」を選択



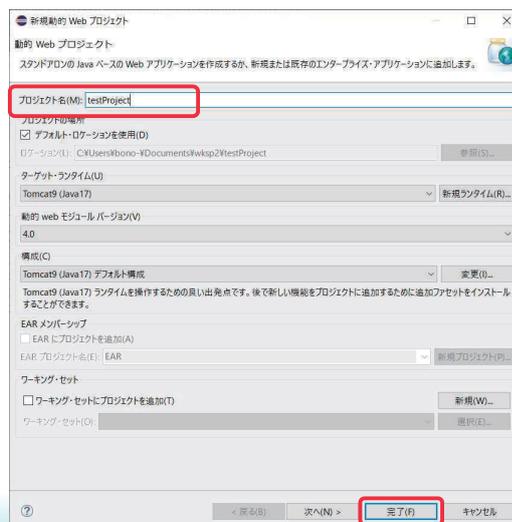
50

第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.6 Eclipseの起動と使い方

- 「プロジェクト名」欄に任意のプロジェクト名を入力
- プロジェクト名を入力後、「完了」ボタンをクリック



51

第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.6 Eclipseの起動と使い方

- ファイルを新規作成するにはメニューの「ファイル」→「新規」→「HTMLファイル」を選択
- 作成するプログラムの種類を選択し、ファイルを新規作成



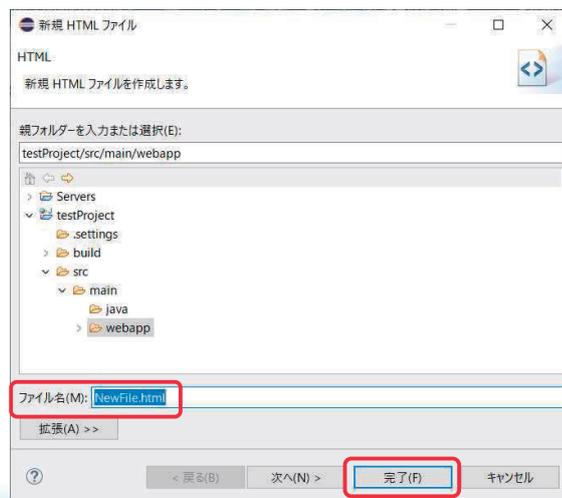
52

第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.6 Eclipseの起動と使い方

- 作成するHTMLファイル名を入力し「完了」ボタンをクリック



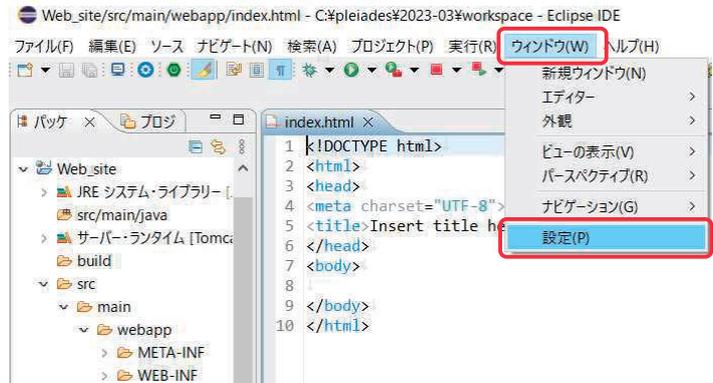
53

第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.6 Eclipseの起動と使い方

- HTMLファイルが新規作成される
- プログラムの文字サイズは変更可能
- メニューの「ウィンドウ」→「設定」をクリック



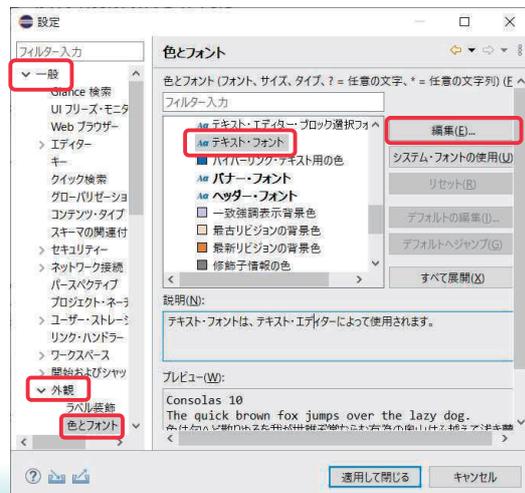
54

第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.6 Eclipseの起動と使い方

- 「一般」→「外観」→「色とフォント」をクリックし、色とフォントの項目一覧を表示
- 「テキスト・フォント」を選択し「編集」ボタンをクリック



55

第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.6 Eclipseの起動と使い方

- 「フォント名」「スタイル」「サイズ」を好みのものに変更
- 「OK」ボタンをクリック
- 「色とフォント」ダイアログに戻り「適用して閉じる」をクリック



56

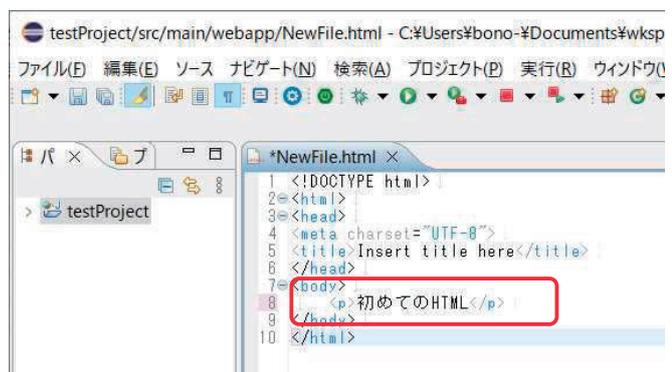
第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.6 Eclipseの起動と使い方

- 8行目に空行があるので、次の内容を入力

<p>はじめてのHTML</p>



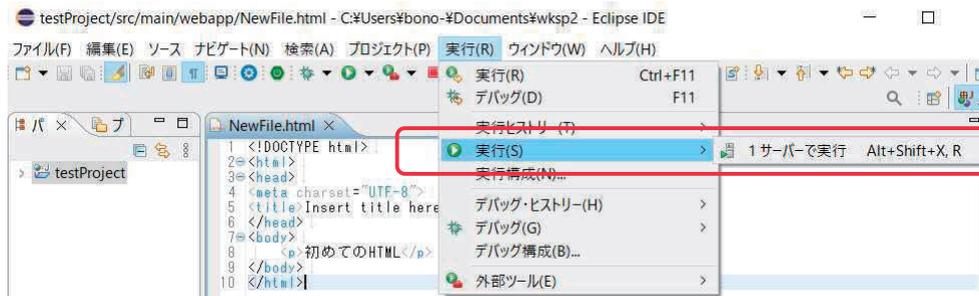
57

第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.6 Eclipseの起動と使い方

- HTMLファイルを表示するにはメニューの「実行」→「実行」→「サーバで実行」を選択



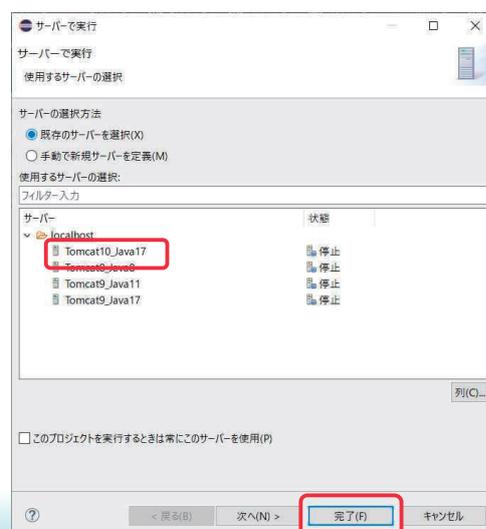
58

第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.6 Eclipseの起動と使い方

- 表示に使用するWebサーバ (Tomcat) を選択して「完了」ボタンをクリック
- 最もバージョンが新しいものを選択



59

第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.6 Eclipseの起動と使い方

- Windows版の初回のみ、右のセキュリティ警告が表示
- 「アクセスを許可する」ボタンをクリック



60

第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.6 Eclipseの起動と使い方

- Windows版の初回のみ、右のセキュリティ警告が表示
- 「アクセスを許可する」ボタンをクリック



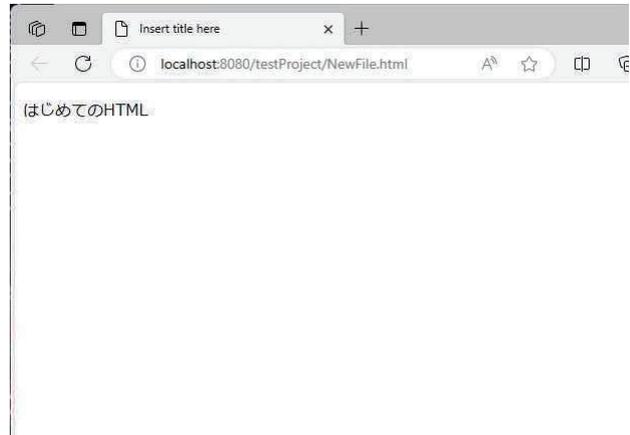
61

第2章 開発環境の構築

2.1 Webアプリケーション開発環境の構築

2.1.6 Eclipseの起動と使い方

- HTMLファイルの記述内容がWebブラウザで表示
- HTMLファイルを変更し、変更後のHTMLファイルを表示するには、HTMLファイルを保存した後にWebブラウザの再読み込みボタンを押すだけで変更内容が反映



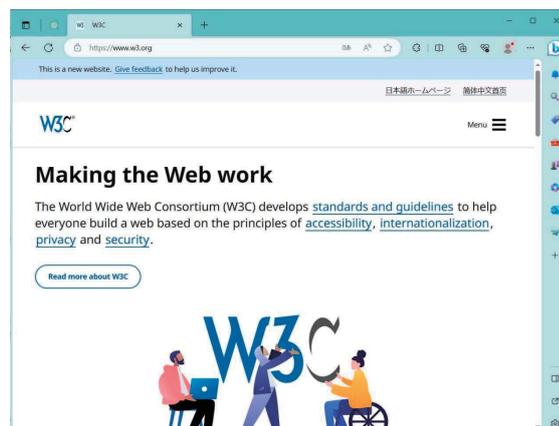
62

第2章 開発環境の構築

2.2 Webブラウザのインストール

2.2.1 Webブラウザとは

- Webブラウザとは、Webページを閲覧するためのソフトウェア
- PC、スマートフォン、タブレット等には必ず何らかのWebブラウザが標準インストール済
- 自分好みのWebブラウザを追加でインストールすることも可能
- 機能拡張を追加でインストールしWebブラウザ自体のカスタマイズも可能



63

第2章 開発環境の構築

2.2 Webブラウザのインストール

2.2.2 Webブラウザの種類



Microsoft Edge
Windows標準搭載
Chromiumベースで生まれ変わった
対話型AI (GPT-4) 搭載



Opera
Chromiumベースで生まれ変わった
機能拡張が無くても広告ブロックや
VPNなど様々な機能を持つ



Google Chrome
Android・Chromebook標準搭載
シェアトップ(2023年時点約50%)
Googleの各サービスと連携



Vivaldi
Operaから分離
新生Operaになる前のOperaの思想を
引き継ぎ、カスタマイズ性が高い



Apple Safari
iOS・Mac標準搭載
Apple製品間での連携がスムーズ
Windows版サポート終了



Mozilla Firefox
オープンソースによる開発
高いカスタマイズ性とセキュリティ性
ラビッドリリースによる新機能搭載

64

第2章 開発環境の構築

2.2 Webブラウザのインストール

2.2.2 Webブラウザの種類



Microsoft Edge
Windows標準搭載
Chromiumベースで生まれ変わった
対話型AI (GPT-4) 搭載



Opera
Chromiumベースで生まれ変わった
機能拡張が無くても広告ブロックや
VPNなど様々な機能を持つ



Google Chrome
Android・Chromebook標準搭載
シェアトップ(2023年時点約50%)
Googleの各サービスと連携



Vivaldi
Operaから分離
新生Operaになる前のOperaの思想を
引き継ぎ、カスタマイズ性が高い



Apple Safari
iOS・Mac標準搭載
Apple製品間での連携がスムーズ
Windows版サポート終了



Mozilla Firefox
オープンソースによる開発
高いカスタマイズ性とセキュリティ性
ラビッドリリースによる新機能搭載

65

第2章 開発環境の構築

2.2 Webブラウザのインストール

2.2.2 Webブラウザの種類



Microsoft Edge
Windows標準搭載
Chromiumベースで生まれ変わった
対話型AI (GPT-4) 搭載



Opera
Chromiumベースで生まれ変わった
機能拡張が無くても広告ブロックや
VPNなど様々な機能を持つ



Google Chrome
Android・Chromebook標準搭載
シェアトップ(2023年時点約50%)
Googleの各サービスと連携



Vivaldi
Operaから分離
新生Operaになる前のOperaの思想を
引き継ぎ、カスタマイズ性が高い



Apple Safari
iOS・Mac標準搭載
Apple製品間での連携がスムーズ
Windows版サポート終了



Mozilla Firefox
オープンソースによる開発
高いカスタマイズ性とセキュリティ性
ラビッドリリースによる新機能搭載

66

第2章 開発環境の構築

2.2 Webブラウザのインストール

2.2.2 Webブラウザの種類



Microsoft Edge
Windows標準搭載
Chromiumベースで生まれ変わった
対話型AI (GPT-4) 搭載



Opera
Chromiumベースで生まれ変わった
機能拡張が無くても広告ブロックや
VPNなど様々な機能を持つ



Google Chrome
Android・Chromebook標準搭載
シェアトップ(2023年時点約50%)
Googleの各サービスと連携



Vivaldi
Operaから分離
新生Operaになる前のOperaの思想を
引き継ぎ、カスタマイズ性が高い



Apple Safari
iOS・Mac標準搭載
Apple製品間での連携がスムーズ
Windows版サポート終了



Mozilla Firefox
オープンソースによる開発
高いカスタマイズ性とセキュリティ性
ラビッドリリースによる新機能搭載

67

第2章 開発環境の構築

2.2 Webブラウザのインストール

2.2.2 Webブラウザの種類



Microsoft Edge
Windows標準搭載
Chromiumベースで生まれ変わった
対話型AI（GPT-4）搭載



Opera
Chromiumベースで生まれ変わった
機能拡張が無くても広告ブロックや
VPNなど様々な機能を持つ



Google Chrome
Android・Chromebook標準搭載
シェアトップ(2023年時点約50%)
Googleの各サービスと連携



Vivaldi
Operaから分離
新生Operaになる前のOperaの思想を
引き継ぎ、カスタマイズ性が高い



Apple Safari
iOS・Mac標準搭載
Apple製品間での連携がスムーズ
Windows版サポート終了



Mozilla Firefox
オープンソースによる開発
高いカスタマイズ性とセキュリティ性
ラビッドリリースによる新機能搭載

68

第2章 開発環境の構築

2.2 Webブラウザのインストール

2.2.2 Webブラウザの種類



Microsoft Edge
Windows標準搭載
Chromiumベースで生まれ変わった
対話型AI（GPT-4）搭載



Opera
Chromiumベースで生まれ変わった
機能拡張が無くても広告ブロックや
VPNなど様々な機能を持つ



Google Chrome
Android・Chromebook標準搭載
シェアトップ(2023年時点約50%)
Googleの各サービスと連携



Vivaldi
Operaから分離
新生Operaになる前のOperaの思想を
引き継ぎ、カスタマイズ性が高い



Apple Safari
iOS・Mac標準搭載
Apple製品間での連携がスムーズ
Windows版サポート終了



Mozilla Firefox
オープンソースによる開発
高いカスタマイズ性とセキュリティ性
ラビッドリリースによる新機能搭載

69

第2章 開発環境の構築

2.2 Webブラウザのインストール

2.2.3 Google Chromeのインストール

- 検索サイトで「Chrome」を検索
- 検索結果からChromeのWebサイトへアクセス
- 右図はMicrosoft Edgeで検索した結果



70

第2章 開発環境の構築

2.2 Webブラウザのインストール

2.2.3 Google Chromeのインストール

- 「Chromeをダウンロード」ボタンをクリック



71

第2章 開発環境の構築

2.2 Webブラウザのインストール

2.2.3 Google Chromeのインストール

- ダウンロード先は、特に変更していなければ「ダウンロード」フォルダ
- ダウンロードしたChromeインストーラをダブルクリック
- インストーラ起動時に許可を求めるダイアログが表示された場合はそれを許可
- 少し待つとChromeのインストール完了



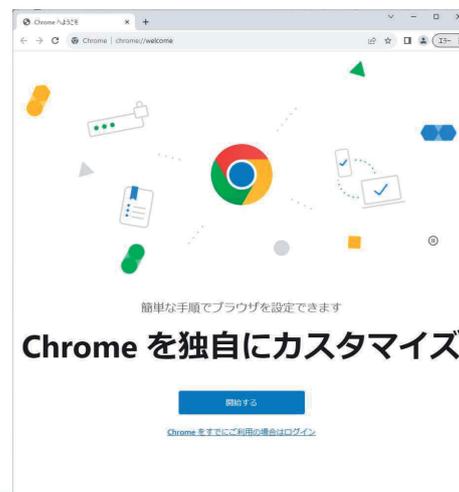
72

第2章 開発環境の構築

2.2 Webブラウザのインストール

2.2.3 Google Chromeのインストール

- インストールが完了すると、自動的にGoogle Chromeが起動



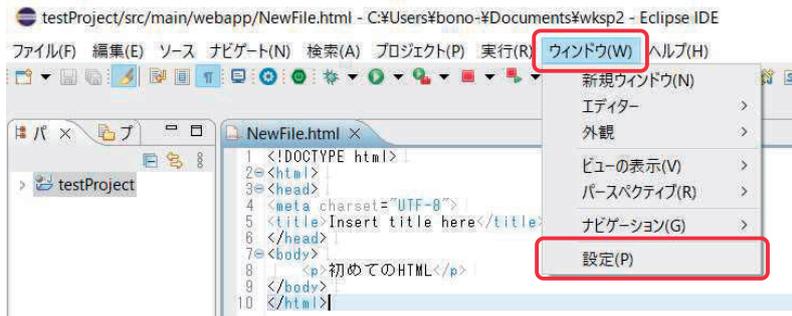
73

第2章 開発環境の構築

2.2 Webブラウザのインストール

2.2.4 Eclipseの設定変更

- Eclipseで作成したWebページを開く既定のWebブラウザをChromeに変更
- Eclipseを起動し、メニューから「ウィンドウ」→「設定」をクリック



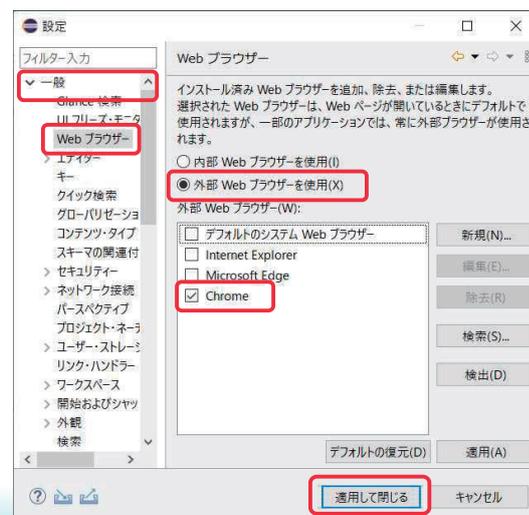
74

第2章 開発環境の構築

2.2 Webブラウザのインストール

2.2.4 Eclipseの設定変更

- 「一般」→「Webブラウザ」→「外部Webブラウザを使用」をクリック
- 外部Webブラウザ欄の「Chrome」をチェック
- 「適用して閉じる」をクリックするとEclipseの既定ブラウザが変更



75

Webシステム概論

第3章

Web技術の概要

【本章学習内容】

本章ではWebアプリケーションの開発で利用される「HTML」「CSS」「JavaScript」「各種ライブラリ」の概要について学習します。

76

第3章 Web技術の概要

3.1 HTML Living Standard

3.1.1 HTMLとは



- HTML (HyperText Markup Language) はWebページに表示する内容を記述するプログラム言語
- Markupとは文書の構成や文書の役割という意味を表す言葉
- HTMLはすべて文字のみで構成
- タグの指定により、文字や文章に意味を付与
(例：段落を表す場合は<p>タグ、大見出しを表す場合は<h1>タグ)
- タグは基本的に、開始タグと終了タグをペアで使用
(例：<p>段落は「p」で囲む</p>)
- デザインや細かい表示の設定はCSSを利用

77

第3章 Web技術の概要

3.1 HTML Living Standard

3.1.1 HTMLとは

The image shows a code editor on the left and a browser window on the right. The code editor displays the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>ページのタイトルは「title」で囲む</title>
6 </head>
7 <body>
8 <h1>大見出しは「H1」で囲む</h1>
9 <p>段落は「p」で囲む</p>
10 </body>
11 </html>
```

The browser window shows the rendered output of this code. The page title is "ページのタイトルは「title」で囲む". The main heading is "大見出しは「H1」で囲む" and the paragraph is "段落は「p」で囲む". Red arrows point from the code editor to the browser window, highlighting the mapping between the code and the rendered output.

78

第3章 Web技術の概要

3.1 HTML Living Standard

3.1.2 HTMLの歴史

1993年	HTML1.0公開	IETFよりドラフト（草稿版）の状態公開
1995年	HTML2.0公開	IETFより正式版として公開
1997年	HTML3.2公開	W3Cより公開（1月）
	HTML4.0公開	W3Cより公開（12月）
1999年	HTML4.01公開	W3Cより公開
2004年	WHATWG発足	
2011年	WHATWGによりHTML Living Standard開始	
2014年	HTML5	W3Cより勧告
2016年	HTML5.1	W3Cより勧告
2019年	WHATWGとW3Cに分かれていた規格を一本化することに合意	
2021年	W3CのHTML規格が廃棄され、WHATWGが設定したHTML Living Standardに一本化	

79

第3章 Web技術の概要

3.1 HTML Living Standard

3.1.3 HTML Living StandardとHTML5の違い



- 両方の内容はほぼ同じ
- 一部のタグの扱いが変更
- HTML5はページの文書構造の適正化に重点を置いていたが、HTML Living StandardはHTML5にWebブラウザの動作に直結する要素を追加

追加要素（抜粋）

利用できるWebブラウザの機能を指定
先に読み込む外部リソースを指定
動画をインラインで再生
入力時のソフトウェアキーボードの種類を指定

など

80

第3章 Web技術の概要

3.2 CSS

3.2.1 CSSとは



- CSS (Cascading Style Sheets) はWebページの表示やレイアウトを指定するためのプログラム言語
- HTMLと組み合わせて使用
- Webブラウザで表示される色、サイズ、レイアウトなどの表示スタイルを指定
- プリンタで印刷するときのみ適用される設定も可能
- 音声で読み上げられるときのみ適用される（読み飛ばしなどの再生スタイル）設定も可能

81

第3章 Web技術の概要

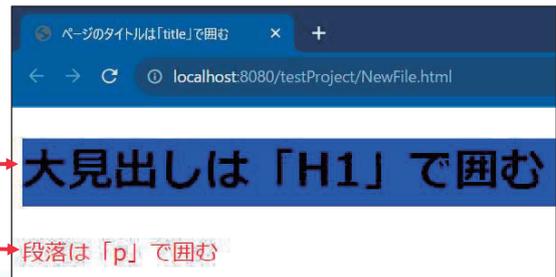
3.2 CSS

3.2.1 CSSとは

```
NewFile.html × style.css
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>ページのタイトルは「title」で囲む</title>
6 <link rel="stylesheet" type="text/css" href="css/style.css" />
7 </head>
8 <body>
9 <h1>大見出しは「H1」で囲む</h1>
10 <p>段落は「p」で囲む</p>
11 </body>
```

参照

```
NewFile.html × style.css
1 @charset "UTF-8";
2
3 h1 {
4     background-color: blue;
5 }
6
7 p {
8     color: red;
9 }
```



82

第3章 Web技術の概要

3.2 CSS

3.2.2 CSSの歴史

1994年	CSSの草案を提唱（10月）
	Netscape Navigator（後のFirefox）発売（12月）
1995年	Microsoft Internet Explorer発売によりブラウザ戦争勃発
1996年	CSS level1勧告
1998年	CSS level2勧告（5月）
	Windows98にInternet Explorerが標準搭載されブラウザ戦争終結（8月）
2011年	CSS level2 Revision1勧告
2016年以降	CSS level3徐々に勧告

83

第3章 Web技術の概要

3.3 JavaScript

3.3.1 JavaScriptとは



- 動きのあるWebページを作成できるプログラム言語
- 文法はJavaと似ているが、全く別もの
- Webブラウザ上で動作（クライアントサイドJavaScript）
- ライブラリが豊富で高い拡張性を持つ
- HTMLファイルまたはJavaScriptファイルにプログラムを記述

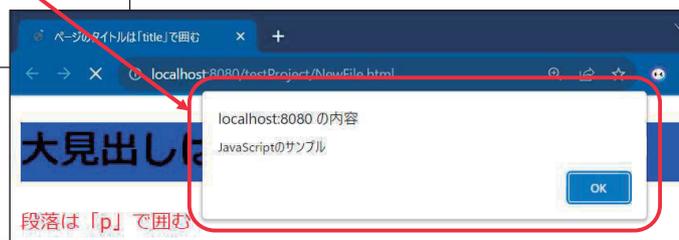
84

第3章 Web技術の概要

3.3 JavaScript

3.3.1 JavaScriptとは

```
*NewFile.html x style.css
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <script>
6   window.onload = function(){
7     alert("JavaScriptのサンプル")
8   }
9 </script>
10 <title>ページのタイトルは「title」で囲む</title>
11 <link rel="stylesheet" type="text/css" href="css/style.css" />
12 </head>
13 <body>
14   <h1>大見出しは「H1」で囲む</h1>
15   <p>段落は「p」で囲む</p>
16 </body>
```



85

第3章 Web技術の概要

3.3 JavaScript

3.3.2 JavaScriptの歴史

1995年	Netscape Navigator2.0に初めてJavaScript搭載
1996年	Microsoft Internet Explorer3.0にJScript搭載
1997年	ECMA InternationalがJavaScript仕様の中核部分を標準化 ECMAScript 1st Edition公開
1998年	ECMAScript 2nd Edition公開
1999年	ECMAScript 3rd Edition公開
2005年	JavaScriptのAjax技術を採用したGoogleマップ公開
2009年	ECMAScript 5th Edition公開
2015年以降	ECMAScriptを毎年改訂、ECMAScript 2015のように名称に年号を付与

86

第3章 Web技術の概要

3.4 CSSフレームワーク

3.4.1 Bootstrapとは



- CSSフレームワークは、ボタン・フォーム・レイアウトなどWebページに適用するCSSの集合体
- 比較的短期間でWebサイトを作成できる
- **デザイン実装の手間を軽減**
- 目的によってフレームワークを使い分ける
- **Bootstrap**は、カスタマイズ可能なCSSフレームワーク
- **ユーザの端末に合わせて自動的にレイアウトを最適化（レスポンシブWebデザイン）**
- 画面の制御にJavaScript使用

87

第3章 Web技術の概要

3.4 CSSフレームワーク

3.4.2 Bootstrap5



- Bootstrapは、Twitter（現X）が開発し、現在はオープンソースとして公開されているCSSフレームワーク
- 2022年にBootstrap5がリリース、現在も新機能の追加が続けられている
- Bootstrap5と旧バージョンとの大きな違いは、外部のJavaScriptライブラリから内部のJavaScriptプログラムを利用するように変更し、Webページの読込時間が大幅短縮
- Bootstrap5でInternet Explorerサポート対象外

88

第3章 Web技術の概要

3.4 CSSフレームワーク

3.4.3 レスポンシブWebデザイン

- 端末やWebブラウザのウィンドウサイズに合わせて自動的にレイアウトを最適化



PC用



タブレット用



スマートフォン用

※出典：内閣府ホームページ (<https://www.cao.go.jp/>)

89

第3章 Web技術の概要

3.5 JavaScriptライブラリ

3.5.1 JavaScriptライブラリとは



- JavaScriptの使用頻度が高いプログラム部品をまとめたもの
- Webアプリケーション開発時にJavaScriptプログラムをすべて書くのではなく、JavaScriptライブラリを活用して開発
- JavaScriptライブラリの活用実績により、それ自体に**バグが含まれる可能性は低い**
- **jQuery**は、使いやすくまとめられたJavaScriptライブラリ
- Webブラウザの互換性が保たれる

90

第3章 Web技術の概要

3.5 JavaScriptライブラリ

3.5.2 jQueryとは



- 2016年にjQuery3がリリース、現在も新機能の追加が続けられている
- 公式キャッチコピー「**write less, do more**（少ない記述でもっと多くのことをする）」
- 数十行にわたるコードが必要だった処理が、わずか数行のコードで実現
- jQuery本体にない機能を実装したjQueryプラグインが豊富
- jQueryプラグインの代表例として、日付選択時のカレンダー表示機能、郵便番号入力時の住所表示機能、表形式データの並べ替え機能など
- jQuery本体のAjax技術を使用すると、Googleマップのような**非同期通信が実現可能**
- jQuery本体とjQueryプラグインの導入は比較的容易

91

フロントエンドエンジニア 養成講座

UI（ユーザインターフェイス）概論

1

はじめに

【学習目標】

本テキストでは、UI（ユーザインターフェイス）を理解することを目的とします。

目次

第1章	UIとUX	003
第2章	デザインプロセス	054
第3章	UI部品	091
第4章	UIコンポーネント	111

2

UI概論

第1章

UIとUX

【本章学習内容】

本章では、UI（ユーザインタフェース）とは何なのか、またUIとUX（ユーザエクスペリエンス）との関係性などについて学習します。

3

第1章 UIとUX

1.1 UIとは

1.1.1 UIとはユーザインタフェースの略称

ユーザとWebアプリケーションや製品
またはサービスの接地面を表す言葉



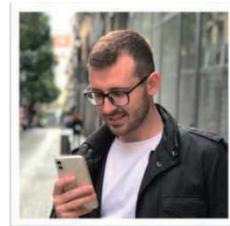
4

第1章 UIとUX

1.1 UIとは

1.1.2 UI（ユーザインタフェース）の分類

- ① GUI：グラフィカルユーザインタフェース
- ② CUI：キャラクターベースユーザインタフェース
- ③ NUI：ナチュラルユーザインタフェース
- ④ VUI：ボイスユーザインタフェース



5

第1章 UIとUX

1.1 UIとは

1.1.3 GUI、グラフィカルユーザインタフェースとは

コンピュータ上の画面がグラフィック表示され、マウスを使ったマウスポインタの移動やクリックなどの操作を使った入力方式を採用しているユーザインタフェースのこと。



6

第1章 UIとUX

1.1 UIとは

1.1.4 CUI、キャラクターベースユーザインタフェースとは

キーボードなどの操作を介して、
文字のコマンドによりコンピュータの
操作を行う方式のこと。



7

第1章 UIとUX

1.1 UIとは

1.1.5 NUI、ナチュラルユーザインタフェースとは

人間が行う日常的な動作を通じて
コンピュータや電化製品、スマートフォンなどを
操作する方式のこと。



8

第1章 UIとUX

1.1 UIとは

1.1.6 VUI、ボイスユーザインタフェースとは

人間が行う日常的な動作の一つ声や音を使ってコンピュータや電化製品、スマートフォンなどを操作する方式のこと。NUIの一つとして認識されることもある



9

第1章 UIとUX

1.2 良いUI（ユーザインタフェース）とは

1.2.1 UI（ユーザインタフェース）を良くするには

アクセシビリティが確保されたUI（ユーザインタフェース）



ユーザビリティの重要性

良いUI

ユーザビリティに優れた
UI（ユーザインタフェース）



悪いUI

ユーザビリティに問題がある
UI（ユーザインタフェース）

10

第1章 UIとUX

1.2 良いUI（ユーザインタフェース）とは

1.2.2 ヤコブ・ニールセンによる5分類（ユーザビリティ特性）

UI（ユーザインタフェース）評価の基準として
提唱された「ヤコブ・ニールセンによる5分類」

- 学習しやすさ（Learnability）
- 効率性（Efficiency）
- 記憶しやすさ（Memorability）
- 間違いにくさ（エラー（Errors））
- 主観的満足度（Satisfaction）



出展：Wikipedia

第1章 UIとUX

1.2 良いUI（ユーザインタフェース）とは

1.2.3 学習しやすさ（Learnability）とは

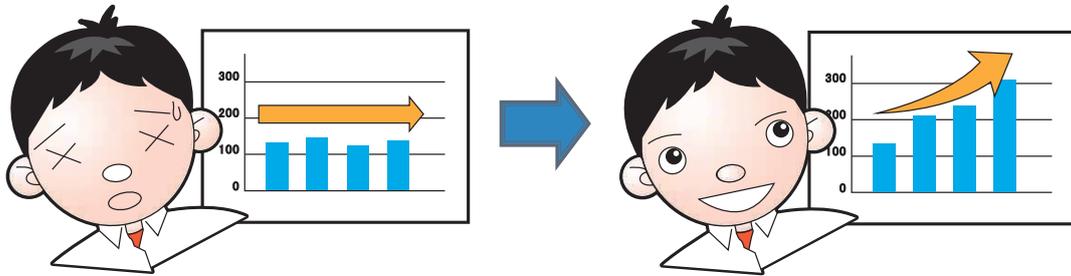


ユーザがすぐ使い始められるよう、簡単に学習できるように
マニュアルを見ずとも簡単な操作方法が自然に行えるようなWebシステムのインタ
フェースは理想的

第1章 UIとUX

1.2 良いUI（ユーザインタフェース）とは

1.2.4 効率性（Efficiency）とは



一度学習すれば、あとは高い生産性を上げられるよう、効率的に使用可能
一度操作を覚えてしまえば、効率良く操作が可能。業務効率がアップすることで高い生産性を維持することができる。

13

第1章 UIとUX

1.2 良いUI（ユーザインタフェース）とは

1.2.5 記憶しやすさ（Memorability）とは



ユーザがしばらく使わなくても、また使うときにはすぐ使えるように覚えやすくする
忘れにくいUIのWebシステムは、すぐに使うことができるため、ユーザにとって必要不可欠なWebシステムとなる

14

第1章 UIとUX

1.2 良いUI（ユーザインタフェース）とは

1.2.6 エラー（Errors）、いわゆる「間違えにくさ」とは



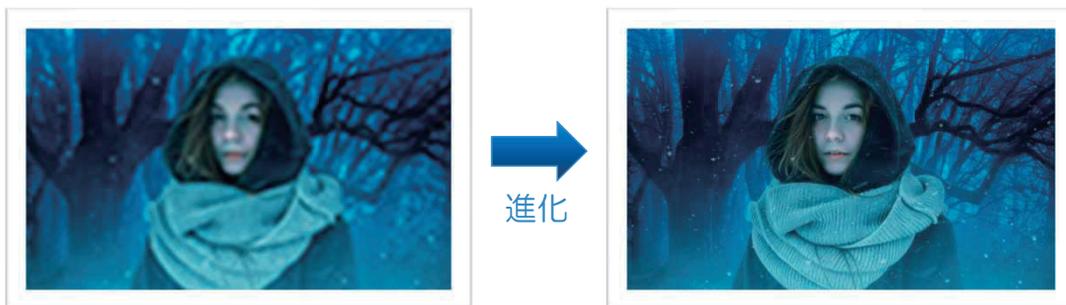
間違いの発生率を低くし、間違いが起こっても回復、致命的な間違いを起こさない
ユーザの間違いの発生を極力抑え、間違いが起っても致命的な間違いを起こさずに、
発生しても復旧することが可能であるならば、ユーザは安心して操作可能

15

第1章 UIとUX

1.2 良いUI（ユーザインタフェース）とは

1.2.7 主観的満足度（Satisfaction）とは



ユーザが個人的に満足できるよう、楽しく利用できなければならない
サイトを利用したユーザがその操作を通して、楽しいと感じる満足度

16

第1章 UIとUX

1.3 UX（ユーザエクスペリエンス）とは

1.3.1 UXとはユーザ体験のこと

UX：User Experience（ユーザ体験）

- 製品やサービスなどを利用したユーザの体験を示す言葉
- この体験が良いものであればあるほど、それが利用された製品やサービスの評価が上がる



17

第1章 UIとUX

1.3 UX（ユーザエクスペリエンス）とは

1.3.2 UXとUIの違い

UX（ユーザエクスペリエンス）とUI（ユーザインタフェース）の違いをしっかりと理解することでUX・UIの理解を深める

UX（ユーザエクスペリエンス）

ユーザが体験を通じて得られるものの総称

UI（ユーザインタフェース）

- ユーザと機器やサービスなどをつなぐもの

UIとはUXの中の一つの要素、UXを高める手段の一つ

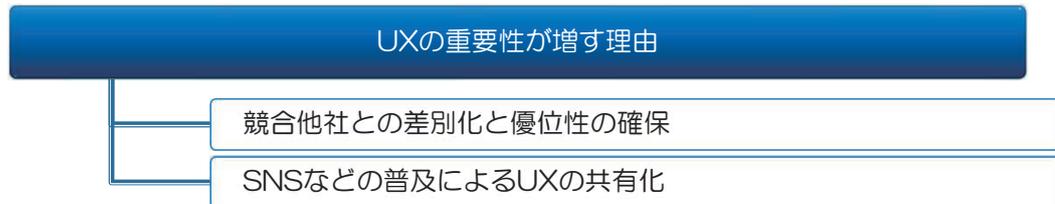
18

第1章 UIとUX

1.3 UX（ユーザエクスペリエンス）とは

1.3.3 なぜUXは重要なのか？

市場が成熟することで製品などの性能の差別化が図りにくくなったことで、ユーザの価値判断基準が単なる性能表などから変化し、その機器やサービスを利用した体験への評価に代わってきた。

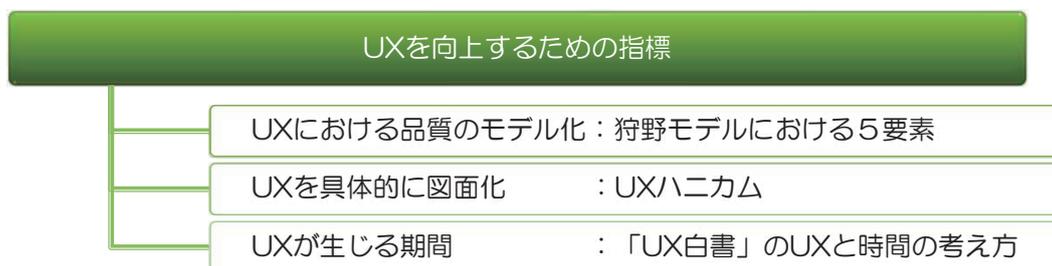


19

第1章 UIとUX

1.3 UX（ユーザエクスペリエンス）とは

1.3.4 良いUXを提供するには



20

第1章 UIとUX

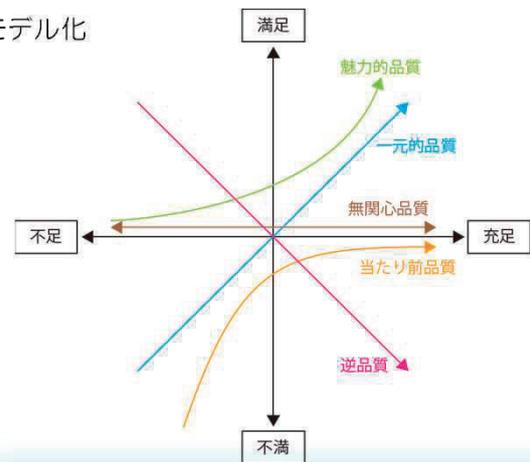
1.3 UX（ユーザエクスペリエンス）とは

1.3.5 狩野モデルにおける5要素

- 顧客満足度を左右する品質の5要素としてモデル化

1. 当たり前品質（Must-Be）
2. 一元的品質（One-dimensional）
3. 魅力的品質（Attractive）
4. 無関心品質（Indifferent）
5. 逆品質（Reverse）

※優先順位を守ることで、
ユーザのニーズに応じた
製品やサービスを構築



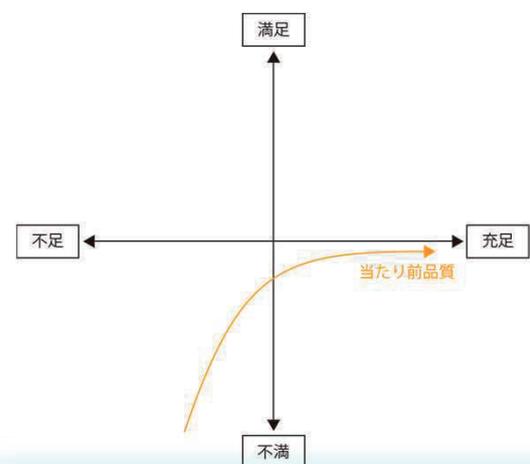
21

第1章 UIとUX

1.3 UX（ユーザエクスペリエンス）とは

1.3.6 優先順位1：当たり前品質（Must-Be）

当然満たされているである品質のこと



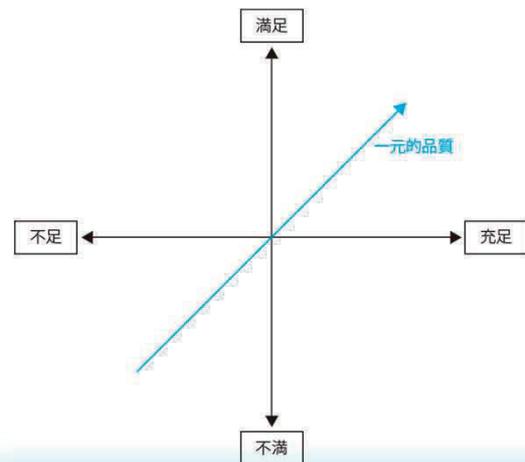
22

第1章 UIとUX

1.3 UX（ユーザエクスペリエンス）とは

1.3.7 優先順位2：一元的品質（One-dimensional）

品質が満たされればそれに伴い
ユーザの満足度が上がる品質のこと



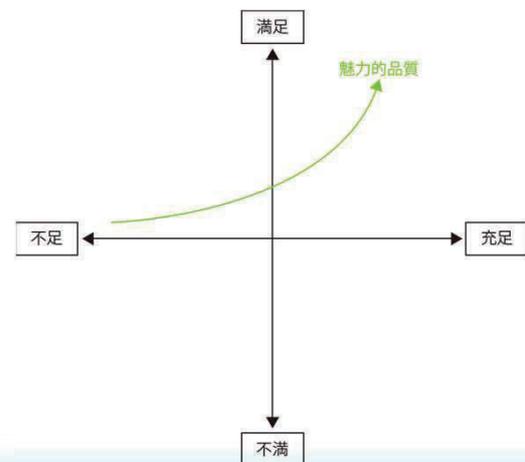
23

第1章 UIとUX

1.3 UX（ユーザエクスペリエンス）とは

1.3.8 優先順位3：魅力的品質（Attractive）

品質が満たされてなくてもユーザの満足度は
低下しない。また満たされれば満足度が上がる
品質のこと



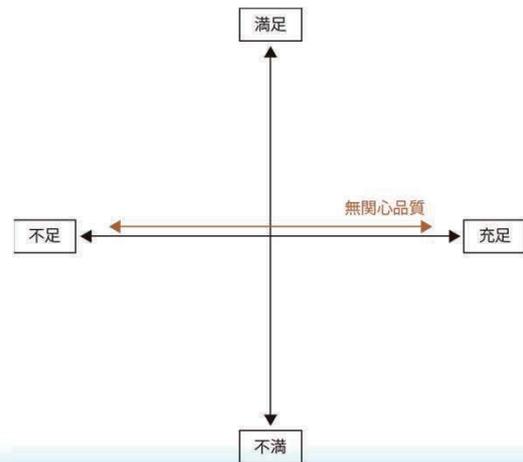
24

第1章 UIとUX

1.3 UX（ユーザエクスペリエンス）とは

1.3.9 優先順位4：無関心品質（Indifferent）

ユーザの満足度に関係しない品質のこと



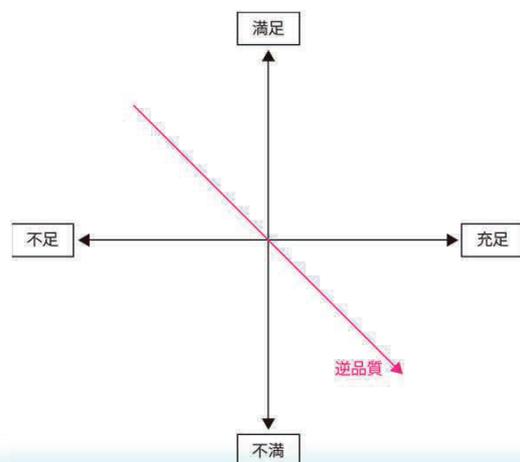
25

第1章 UIとUX

1.3 UX（ユーザエクスペリエンス）とは

1.3.10 優先順位5：逆品質（Reverse）

品質が満たされることで顧客満足度を下げ
てしまう品質のこと



26

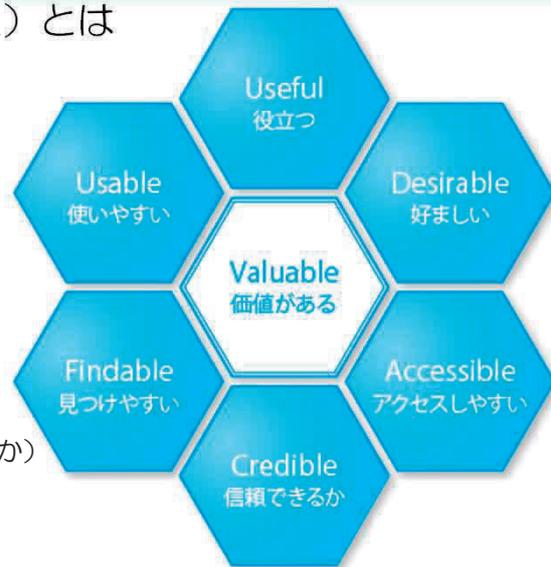
第1章 UIとUX

1.3 UX（ユーザエクスペリエンス）とは

1.3.11 UXハニカム

ピーターモービル氏（UXデザイナー）が提唱
「UXに必要な要素をわかりやすく図式化」

- Valuable（価値があるかどうか）
 1. Useful（役立つかどうか）
 2. Usable（使いやすいかどうか）
 3. Desirable（好ましいかどうか）
 4. Findable（見つけやすいかどうか）
 5. Accessible（アクセスしやすいかどうか）
 6. Credible（信頼できるかどうか）



27

第1章 UIとUX

1.3 UX（ユーザエクスペリエンス）とは

1.3.12 UXと時間「UXの期間」モデルとは

UX白書の中でまとめられたUXと時間の考え方

※UX白書：2010年9月15～18日にドイツで行われたUXセミナーの成果をまとめられたもの

- UXの時間はUXの期間によって分類



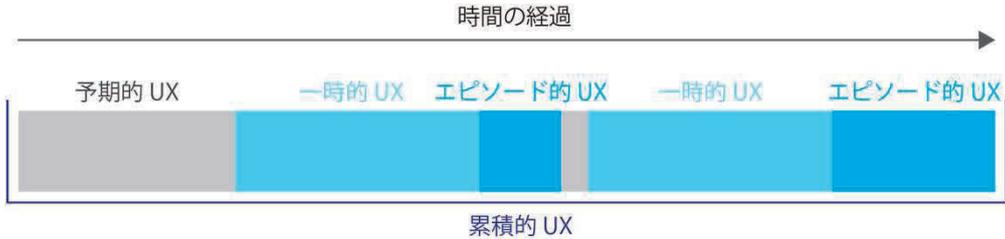
28

第1章 UIとUX

1.3 UX（ユーザエクスペリエンス）とは

1.3.13 時間軸上の「UXの分類」とは

ユーザが獲得するUXの期間を時間軸に置き換えて見てみると



- 予期的UX : UXを実体験する前の情報を得る期間を指す
- 一時的UX : 実際に機器やサービスを利用、体験する期間を指す
- エピソード的UX : 体験後そのUXの振り返りなどで得られ追体験期間を指す
- 累積的UX : 上記の3つの期間が繰返し行われるそのすべての期間を指す

第1章 UIとUX

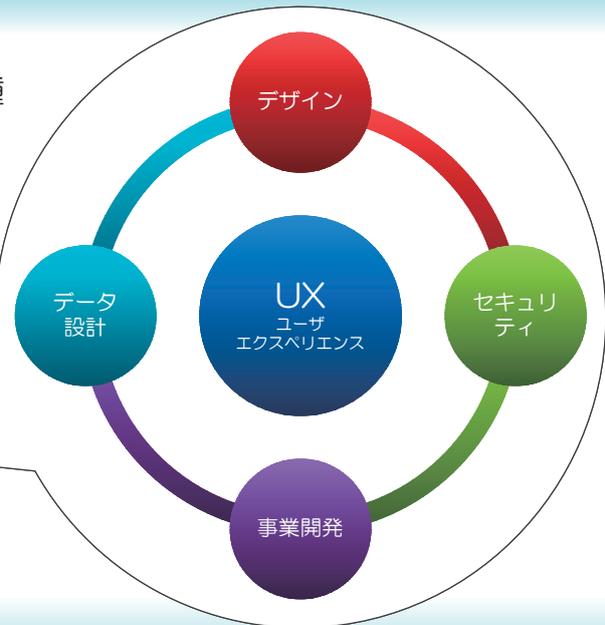
1.4 UX/UIデザイナー

1.4.1 UX/UIデザイナーという職種

Q UX（ユーザエクスペリエンス）を必要とされるのはどんな業務？



A ビジネスにかかわる全ての業務で必要



第1章 UIとUX

1.4 UX/UIデザイナー

1.4.2 ビジネスシーンにおけるUX/UIデザイナーの役割

重要と考えられているUX・UIをビジネスシーンで活用するにはどのような人材が必要？

経済産業省が策定したデジタル人材の育成の中の「デジタルスキル標準」



経済産業省HPより

第1章 UIとUX

1.4 UX/UIデザイナー

1.4.3 DX推進スキル標準の人材類型とロールの構成

人材類型	ビジネスアーキテクト			デザイナー			データサイエンティスト			ソフトウェアウェアエンジニア			サイバーセキュリティ		
ロール (DX推進において担う責任、 主な業務、必要なスキルにより定義)	ビジネスアーキテクト (新規事業開発)	ビジネスアーキテクト (既存事業の高度化)	ビジネスアーキテクト (社内業務の高度化・効率化)	サービスデザイナー	CX/UIデザイナー	グラフィックデザイナー	データビジネスストラテジスト	データサイエンスプロフェッショナル	データエンジニア	フロントエンドエンジニア	バックエンドエンジニア	クラウドエンジニア/SRE	フィジカルコンピューティングエンジニア	サイバーセキュリティマネージャー	サイバーセキュリティエンジニア

<デジタル標準ver1.1：DX推進スキル標準の構成内グラフ（一部抜粋）>

第1章 UIとUX

1.4 UX/UIデザイナー

1.4.4 DX推進スキル標準の構成：言葉の定義

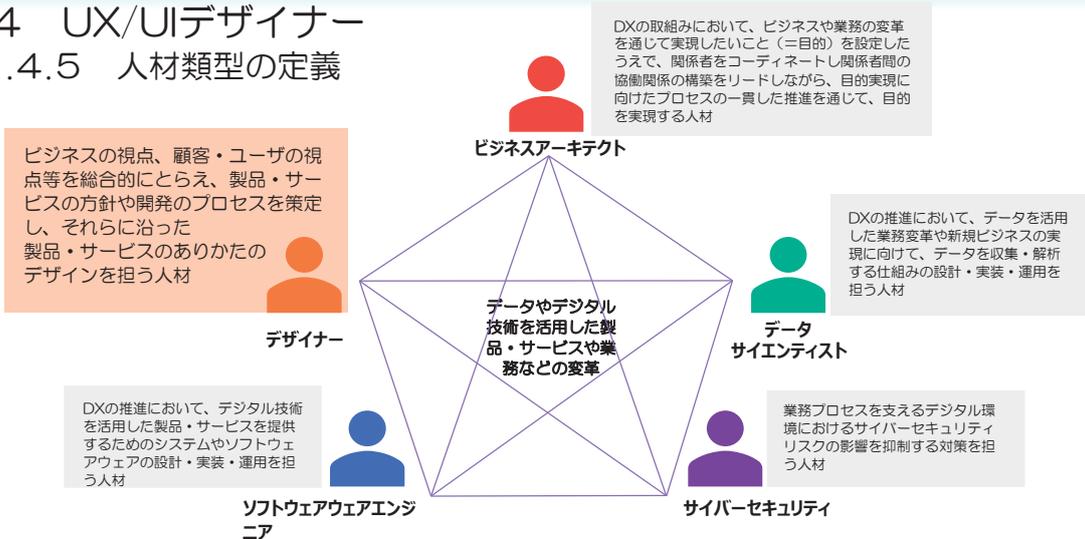
用語	定義
人材類型	企業や組織のDXの推進において必要とされる人材を5つの類型に区分したもの (ビジネスアーキテクト/デザイナー/データサイエンティスト/ソフトウェアエンジニア/サイバーセキュリティ)
ロール	ある人材類型を、業務の違いによりさらに詳細に区分したもの ※1人の人材が複数のロールを兼ねることもある
DXの推進において担う責任	各ロールがDXを推進する上で果たすべきミッション
主な業務	各ロールが上記責任を果たすために実施すべきと考えられる仕事・作業・タスク
スキル項目	企業や組織のDX推進における業務遂行に必要とされる知識や能力などの項目
学習項目例	あるスキル項目の習得に向けて学習が期待される項目の一例 ※学習項目例はあくまでも例であり、全ての項目の学習を必須とするものではない

<デジタル標準ver1.1：DX推進スキル標準の構成 | 言葉の定義より>

第1章 UIとUX

1.4 UX/UIデザイナー

1.4.5 人材類型の定義



<デジタル標準ver1.1：人材類型の定義よりイメージを抜粋>

第1章 UIとUX

1.4 UX/UIデザイナー

1.4.6 人材類型間の連携

	ビジネスアーキテクト	デザイナー	データサイエンティスト	ソフトウェアエンジニア	サイバーセキュリティ
ビジネスアーキテクト					
デザイナー	<ul style="list-style-type: none"> 顧客・ユーザ調査の結果から導出されたインサイトを踏まえた製品・サービスのアイデアの検討 				
データサイエンティスト	<ul style="list-style-type: none"> データ分析結果から得られる示唆を踏まえた製品・サービスのアイデアの検討 	<ul style="list-style-type: none"> 顧客・ユーザ理解や製品・サービス検証のための調査、データ取得、分析、および分析結果の見せ方に関する検討 			
ソフトウェアエンジニア	<ul style="list-style-type: none"> 新技術・ツールを起点とした製品・サービスのアイデアの検討 顧客ニーズに基づく開発要件の定義やソフトウェアアーキテクチャの設計 開発の優先順位の決定 	<ul style="list-style-type: none"> デザインガイドライン、ユーザビリティ、倫理的妥当性を考慮した製品・サービスの開発、評価、検証 	<ul style="list-style-type: none"> 新たなデータ収集・蓄積・解析・可視化の仕組みと既存のシステム等との連携・接続の仕組みの検討 		
サイバーセキュリティ	<ul style="list-style-type: none"> コストとリスクのバランスを考慮した、製品・サービスのリスクへの最適な対応策の検討 リスクに応じた新たなルールの検討 	<ul style="list-style-type: none"> セキュリティ強化によるユーザの負担感を低減させるUIの検討 	<ul style="list-style-type: none"> データ管理やプライバシー保護に関するポリシーの検討 	<ul style="list-style-type: none"> 新製品・サービスのリスクに応じたセキュリティルールや対策の策定 	

<デジタル標準ver1.1：人材類型間の連携より表を抜粋>

第1章 UIとUX

1.4 UX/UIデザイナー

1.4.7 人材タイプのローラー一覧（DX推進において担う責任）

人材類型	ロール	DX推進において担う責任
ビジネスアーキテクト		紙面の都合上省略
デザイナー	サービスデザイナー	社会、顧客・ユーザ、製品・サービス提供における社内外関係者の課題や行動から顧客価値を定義し製品・サービスの方針（コンセプト）を策定するとともに、それを継続的に実現するための仕組みのデザインを行う
	UX/UIデザイナー	バリュープロポジションに基づき製品・サービスの顧客・ユーザ体験を設計し、製品・サービスの情報設計や、機能、情報の配置、外観、動的要素のデザインを行う
	グラフィックデザイナー	ブランドのイメージを具現化し、ブランドとして統一感のあるデジタルグラフィック、マーケティング媒体等のデザインを行う
データサイエンティスト		紙面の都合上省略
ソフトウェアエンジニア		紙面の都合上省略
サイバーセキュリティ		紙面の都合上省略

<デジタル標準ver1.1：ローラー一覧より表を抜粋>

第1章 UIとUX

1.4 UX/UIデザイナー

1.4.8 共通スキルリストの全体像

カテゴリー	サブカテゴリー	スキル項目	カテゴリー	サブカテゴリー	スキル項目		
ビジネス変革	戦略・マネジメント・システム	ビジネス戦略策定・実行	テクノロジー	ソフトウェア開発	コンピュータサイエンス		
		プロダクトマネジメント			チーム開発		
		変革マネジメント			ソフトウェアウェア設計手法		
		システムズエンジニアリング			ソフトウェアウェア開発プロセス		
		エンタープライズアーキテクチャ			Webアプリケーション基本技術		
	プロジェクトマネジメント	フロントエンドシステム開発					
	ビジネスモデル・プロセス	ビジネス調査			バックエンドシステム開発		
		ビジネスモデル設計			クラウドインフラ活用		
		ビジネスアナリシス			SFDCプロセス		
		検証（ビジネス視点）			サービス活用		
		マーケティング			フィジカルコンピューティング		
	デザイン	顧客・ユーザ理解			その他先端技術	デジタルテクノロジー	テクノロジートレンド
		価値発見・定義			セキュリティマネジメント		セキュリティ体制構築・運営
		設計			セキュリティマネジメント		セキュリティマネジメント
		検証（顧客・ユーザ視点）			インシデント対応と事業継続		プライバシー保護
その他デザイン技術		セキュリティ技術	セキュア設計・開発・構築				
データ活用	データ・AIの戦略的活用	データ理解・活用	セキュリティ	セキュリティ運用・保守・監視	リーダーシップ		
		データ・AI活用戦略			コラボレーション		
		データ・AI活用業務の設計・事業案策定・評価			パーソナルスキル	コンセプチュアルスキル	
	AI・データサイエンス	数理統計・多変量解析・データ可視化	パーソナルスキル	ヒューマンスキル	コンセプチュアルスキル	ゴール設定	
		機械学習・深層学習				創造的な問題解決	
		データ活用基礎設計				批判的思考	
	データエンジニアリング	データ活用基礎策定・運用	パーソナルスキル	ヒューマンスキル	コンセプチュアルスキル	適応力	

<デジタル標準ver1.1：共通スキルリストの全体像より表を抜粋>

第1章 UIとUX

1.4 UX/UIデザイナー

1.4.9 ビジネス変革、デザインの共通スキル項目詳細

カテゴリー	サブカテゴリー	スキル項目	内容（スキル項目ごとの説明）
ビジネス変革	デザイン	顧客・ユーザ理解	<ul style="list-style-type: none"> ユーザ調査（顧客満足度・利用データ等の調査やインタビュー等）や市場・競合調査の設計、実施を行うスキル ユーザ調査の結果から、顧客の期待や不満、新たなニーズや競合、トレンドを把握・分析し、インサイトを導き出すスキル
		価値発見・定義	<ul style="list-style-type: none"> ステークホルダーをファシリテートしながら、顧客・ユーザのニーズを基にアイデアを発散させ、バリュープロポジションを定義するスキル
		設計	<ul style="list-style-type: none"> 顧客・ユーザのニーズを踏まえて、必要な機能やコンテンツを明確化するスキル 顧客・ユーザにとってのわかりやすさや見つけやすさを考慮して、機能・コンテンツの構造や骨格をデザインするスキル ユーザにとって好ましい外観や動的要素（Look & Feel）をデザインするスキル
		検証（顧客・ユーザ視点）	<ul style="list-style-type: none"> 定義したバリュープロポジションを、実装した製品やサービスを通じて実際に顧客が体験できるか、顧客にとって有用な体験になっているかどうかを検証するスキル
		その他デザイン技術	<ul style="list-style-type: none"> マーケティングに関わるデジタル媒体のグラフィックをデザインするスキル 電子書籍・カタログ等の誌面を読みやすい誌面にレイアウトしまとめるスキル

<デジタル標準ver1.1：ビジネス変革 | デザインより表を抜粋>

第1章 UIとUX

1.4 UX/UIデザイナー

1.4.10 人材類型：デザイナーとは（1）

定義

ビジネスの視点、顧客・ユーザの視点等を総合的にとらえ、製品・サービスの方針や開発のプロセスを策定し、それらに沿った製品・サービスのありかたのデザインを担う人材

「デザイナー」を定義する理由

DXを推進する人材として、データやデジタル技術の活用の先にあるビジネスそのものの変革を、ビジネスの視点だけでなく顧客・ユーザの視点を起点として実現する人材が必要であると考え、本類型を定義することにした。

経済産業省の「デザイン政策ハンドブック2020」によると、市場や技術、社会の大きな変化により、デザインに求められる役割は、単なる造形を美しくする役割から、人を起点とした価値創造・問題解決の手段へと変化している。このようなデザインに期待される役割の変化を踏まえ、顧客・ユーザの視点からビジネスの変革を実現する人材として「デザイナー」を定義した。

<デジタル標準ver1.1：デザイナーとはより抜粋>

39

第1章 UIとUX

1.4 UX/UIデザイナー

1.4.10 人材類型：デザイナーとは（2）

「デザイナー」が活躍する場面

デザイナーの活躍場面として、組織ケイパビリティの強化に関する取組み（デザイン思考を全社的に浸透させるような組織作り、人材育成）は除外しており、あくまでも個別のデータやデジタル技術を活用した取組み（個別の製品・サービス・業務の単位を想定）を対象としている。

- ・DX推進スキル標準で想定する人材のレベルとして、全社的な取組みの責任を担うような経営層レベルを想定していないため
- ・ただし、プロジェクトの性質によっては組織ケイパビリティ強化が求められる場合もあるため、これに関するスキルをデザイナー類型が持っておくべきものとして定義した

個別の取組みの中には、製品・サービスの提供先が社内である場合も含まれる。

個別の取組みの中では、DXを進めていくためのあらゆるプロセス（例：構想、実装、仮説検証、導入後の効果検証等）においてデザイナーが活躍するものと想定。デザインに期待される役割の変化を踏まえ、単なる外観のデザインだけでなく、新たな製品・サービスの構想においてもデザイナーが活躍するものとする。

<デジタル標準ver1.1：デザイナーとはより抜粋>

40

第1章 UIとUX

1.4 UX/UIデザイナー

1.4.11 人材類型：期待される役割



顧客・ユーザ視点でのアプローチを、取組みの関係者が常に意識できるように導く



倫理的観点を踏まえた顧客・ユーザとの接点（製品・サービスと顧客・ユーザとが関わるポイント）のデザインを行う

<デジタル標準ver1.1：デザイナーとは | 期待される役割1/2より抜粋>

41

第1章 UIとUX

1.4 UX/UIデザイナー

1.4.12 顧客・ユーザ視点でのアプローチを関係者に

顧客・ユーザ視点でのアプローチを、取組みの関係者が常に意識できるように導く

見落とされがちな顧客・ユーザ視点のアプローチが欠落しないよう、DXの取組みのあらゆる場面において、顧客・ユーザ視点で関係者が取組みを進められるようにサポートすることが求められる。例えば以下のような内容が想定される。

- ・ 製品・サービスの構想において、収益性やコスト削減などの企業視点だけになっていないか確認し、顧客・ユーザ視点の検討をファシリテートする
- ・ アプリケーション等の開発の場面においては、必要な機能が実装できているかだけでなく、顧客・ユーザにとってのユーザビリティ（分かりやすさ、見つけやすさ、使いやすさ）が実現できているかを確認する

<デジタル標準ver1.1：デザイナーとは | 期待される役割1/2より抜粋>

42

第1章 UIとUX

1.4 UX/UIデザイナー

1.4.13 倫理的な観点を踏まえた顧客・ユーザとの接点を

倫理的観点を踏まえた顧客・ユーザとの接点
(製品・サービスと顧客・ユーザとが関わるポイント)のデザインを行う

顧客・ユーザとの接点をデザインするにあたっては、顧客・ユーザにとってその製品・サービスが分かりやすいか、見つけやすいか、好ましいかといった要素だけでなく、倫理的な妥当性(例:非倫理的な誘導を行っていないか)も踏まえることが求められる。

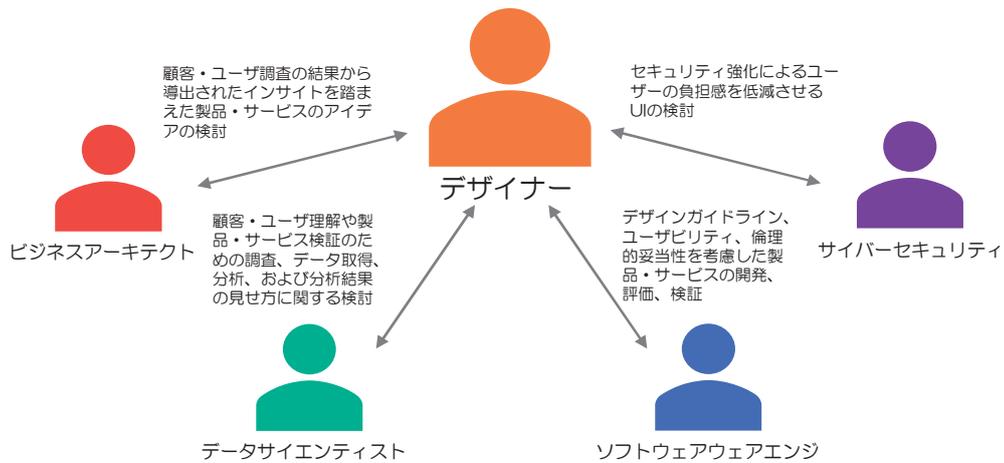
人の行動原理や心理学を基にしてデザインを行うことや、でき上がった製品・サービスについて倫理的観点からのチェックを行い、非倫理的な要素が見つかった場合は差し戻すことが求められる。

<デジタル標準ver1.1:デザイナーとは | 期待される役割1/2より抜粋>

第1章 UIとUX

1.4 UX/UIデザイナー

1.4.14 人材類型:デザイナーと他の人材類型との関係性



<デジタル標準ver1.1:デザイナーとは | 期待される役割2/2より抜粋>

第1章 UIとUX

1.4 UX/UIデザイナー

1.4.15 人材類型：デザイナーのロール | ロールの定義（1）

デザイナー		
サービスデザイナー	UX/UIデザイナー	グラフィックデザイナー

DX推進プロセスによるロール区分

- 「デザイナー」という類型を、業務の違いによって区分したものが「ロール」である
- デザイナーは、DX推進のあらゆるプロセスにおいて活躍することが想定される
 - ① バリュープロポジションの定義、製品・サービスのビジネスモデルやビジネスプロセスのデザイン、方針（コンセプト）の策定
 - ② 製品・サービスにおける顧客・ユーザ体験の検討、情報設計や機能や情報の配置、外観、動的要素のデザイン
 - ③ ブランドイメージの具現化、デジタルグラフィック、マーケティング媒体等のデザイン

<デジタル標準ver1.1：デザイナーのロール | ロールの定義より抜粋>

第1章 UIとUX

1.4 UX/UIデザイナー

1.4.15 人材類型：デザイナーのロール | ロールの定義（2）

デザイナー		
サービスデザイナー	UX/UIデザイナー	グラフィックデザイナー

DX推進プロセスによるロール区分

- 前スライド②について、設計はUXを踏まえて行う必要があり、UXデザインとUIデザインは一体のロールとして定義した。
- 企業のDXを進めていくうえでは、ここに示すロールの担い方はそれぞれである。
- また、これらのロールのうち、DX推進をこれから始める企業が特に優先的に揃えた方がよいと思われるロールは、UX/UIデザイナーであると考えられる。
- ただし、より規模の大きな企業において全社的な変革を進める場合はサービスデザイナーも重宝される場合がある。

<デジタル標準ver1.1：デザイナーのロール | ロールの定義より抜粋>

第1章 UIとUX

1.4 UX/UIデザイナー

1.4.16 デザイナーのロール | スキルマッピングの考え方

サービスデザイナー

- ・社会、顧客・ユーザ、製品・サービス提供における社内外関係者の課題や行動からバリュープロポジションを定義し製品・サービスの方針（コンセプト）を策定

UX/UIデザイナー

- ・バリュープロポジションに基づき製品・サービスの顧客・ユーザー体験を設計し、製品・サービスの情報設計や、機能、情報の配置、外観、動的要素のデザインを行う

グラフィックデザイナー

- ・ブランドのイメージを具現化し、ブランドとして統一感のあるデジタルグラフィック、マーケティング媒体等のデザインを行う

<デジタル標準ver1.1：デザイナーのロール | スキルマッピングの考え方より抜粋>

47

第1章 UIとUX

1.4 UX/UIデザイナー

1.4.17 デザイナーのロール | 担う責任・主な業務・スキル（1）

人材類型	デザイナー
ロール	UX/UIデザイナー
DXの推進において担う責任	バリュープロポジションに基づき製品・サービスの顧客・ユーザー体験を設計し、製品・サービスの情報設計や、機能、情報の配置、外観、動的要素のデザインを行う
主な業務	<ul style="list-style-type: none">・バリュープロポジションに基づき、顧客・ユーザーが製品・サービスとの接点においてとる行動や、行動に至る経緯・思考・感情を可視化し、製品・サービスの顧客・ユーザー体験を設計する・製品・サービスの方針（コンセプト）を、仕様・ガイドライン・デザインプリンシプル等の形に具体化し、顧客・ユーザーにとって心地よい体験を実現するための、製品・サービスにおける情報設計や、機能や情報の配置、外観、動的要素（Look&Feel）のデザインを行う・PoCや本格導入、導入後のそれぞれの段階において、ブランディング、マーケティング施策と連動したWebやアプリケーション等のプロトタイプ作成を行う・PoCや本格導入、導入後のそれぞれの段階において、ユーザービリティ評価（顧客・ユーザーが迷わず目的の情報までたどり着けたかの検証）を行う

<デジタル標準ver1.1：デザイナーのロール | 担う責任・主な業務・スキルより抜粋>

48

第1章 UIとUX

1.4 UX/UIデザイナー

1.4.17 デザイナーのロール | 担う責任・主な業務・スキル (2)

必要なスキル	カテゴリ	サブカテゴリ	スキル項目	重要度
	ビジネス変革	戦略・マネジメント・システム	ビジネス戦略策定・実行	D
			プロダクトマネジメント	C
			変革マネジメント	D
			システムズエンジニアリング	D
			エンタープライズアーキテクチャ	D
			プロジェクトマネジメント	C
		ビジネスモデル・プロセス	ビジネス調査	D
			ビジネスモデル設計	D
			ビジネスアナリシス	D
			検証 (ビジネス視点)	D
			マーケティング	B
		デザイン	ブランディング	C
			顧客・ユーザー理解	A
			価値発見・定義	A
			設計	A
			検証 (顧客・ユーザ視点)	A
			その他デザイン技術	C

<デジタル標準ver1.1：デザイナーのロール | 担う責任・主な業務・スキルより抜粋>

第1章 UIとUX

1.4 UX/UIデザイナー

1.4.17 デザイナーのロール | 担う責任・主な業務・スキル (3)

必要なスキル	カテゴリ	サブカテゴリ	スキル項目	重要度
	データ活用	データ・AIの戦略的活用	データ理解・活用	D
			データ・AI活用戦略	D
			データ・AI活用業務の設計・事業実装・評価	D
		AI・データサイエンス	数理統計・多変量解析・データ可視化	D
			機械学習・深層学習	D
			データ活用基盤設計	D
		データエンジニアリング	データ活用基盤実装・運用	D
			コンピュータサイエンス	C
			チーム開発	B
	ソフトウェア設計手法		C	
	ソフトウェア開発プロセス		C	
	テクノロジー	ソフトウェア開発	Webアプリケーション基本技術	C
			フロントエンドシステム開発	C
			バックエンドシステム開発	C
			クラウドインフラ活用	C
			SREプロセス	C
			サービス活用	C

<デジタル標準ver1.1：デザイナーのロール | 担う責任・主な業務・スキルより抜粋>

第1章 UIとUX

1.4 UX/UIデザイナー

1.4.17 デザイナーのロール | 担う責任・主な業務・スキル (4)

必要なスキル	カテゴリー	サブカテゴリー	スキル項目	重要度
	テクノロジー	デジタルテクノロジー	フィジカルコンピューティング	C
			その他先端技術	D
			テクノロジートレンド	C
	セキュリティ	セキュリティマネジメント	セキュリティ体制構築・運営	D
			セキュリティマネジメント	D
			インシデント対応と事業継続	D
		セキュリティ技術	プライバシー保護	C
			セキュア設計・開発・構築	D
			セキュリティ運用・保守・監視	D
	パーソナルスキル	ヒューマンスキル	リーダーシップ	Z
			コラボレーション	Z
		コンセプチュアルスキル	ゴール設定	Z
			創造的な問題解決	Z
			批判的思考	Z
	適応力	Z		

【重要度凡例】 A：高い実践力と専門性が必要 B：一定の実践力と専門性が必要 C：説明可能なレベルで理解が必要
 D：位置づけや関連性の理解が必要 Z：役割や状況に応じた実践力が必要

<デジタル標準ver1.1：デザイナーのロール | 担う責任・主な業務・スキルより抜粋>

第1章 UIとUX

1.4 UX/UIデザイナー

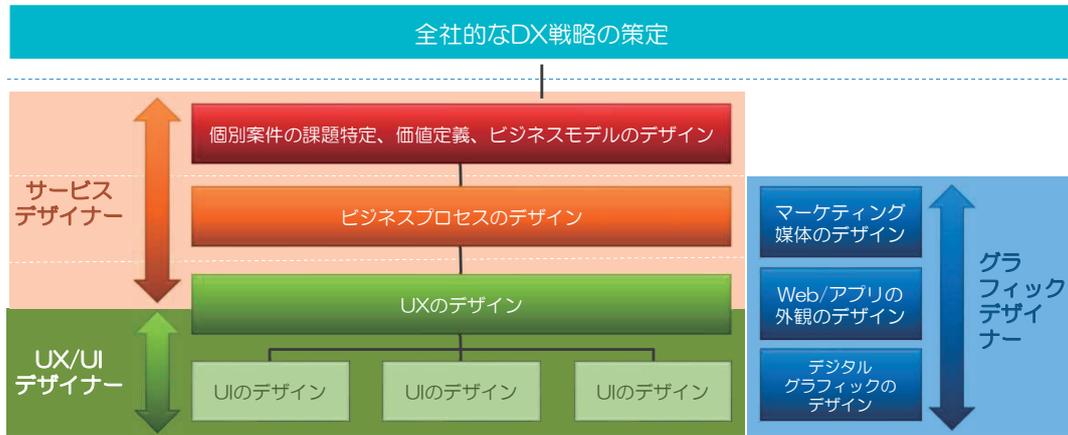
1.4.18 補足資料 | デザインに期待される役割の変化



第1章 UIとUX

1.4 UX/UIデザイナー

1.4.19 補足資料 | 各ロールがデザインする対象



UI概論

第2章

デザインプロセス

【本章学習内容】

本章では、UXを意識したWebサイトの構築に必要なプロセスである「UXデザイン」や、Webサイト全体を把握するための「ナビゲーション設計」などを学習します。

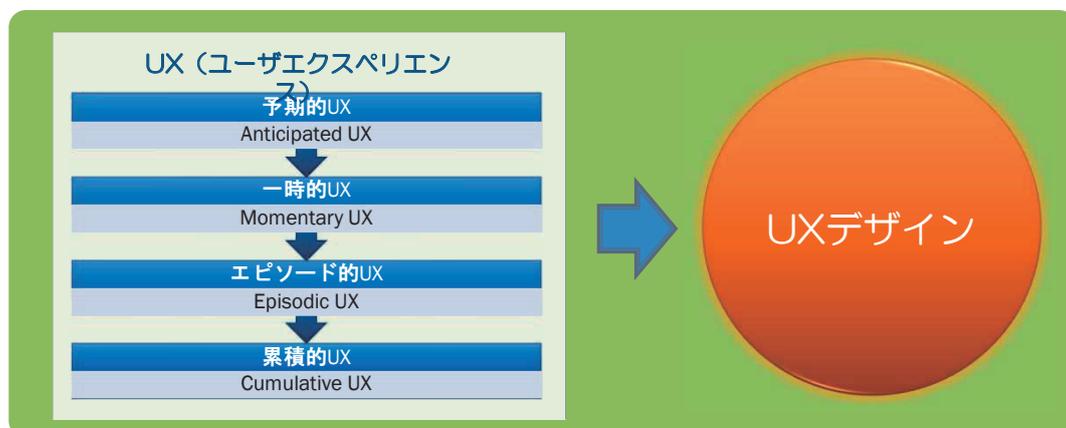
54

第2章 デザインプロセス

2.1 UXデザイン

2.1.1 UXデザインとは

UX（ユーザエクスペリエンス）を重視したデザインのこと



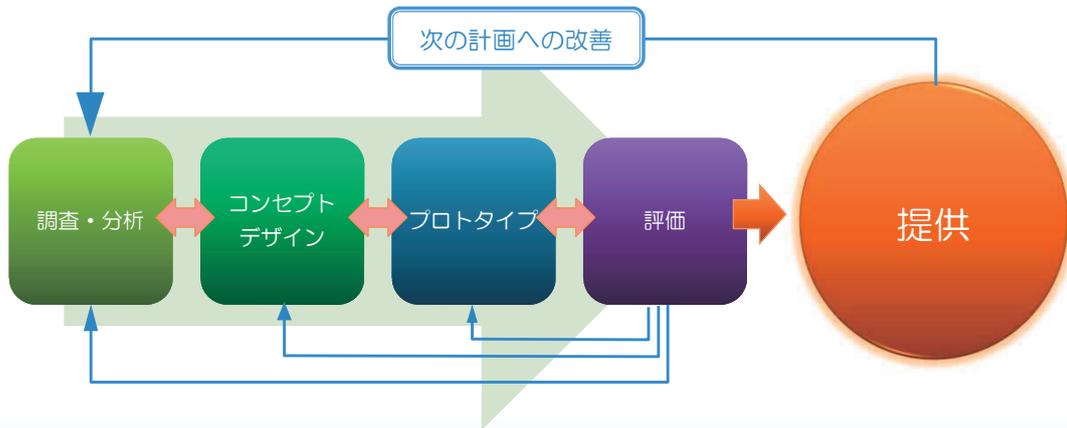
55

第2章 デザインプロセス

2.1 UXデザイン

2.1.2 UXデザインのプロセス

UXデザインは①調査・分析②コンセプトデザイン③プロトタイプ④評価⑤提供



56

第2章 デザインプロセス

2.1 UXデザイン

2.1.3 プロセス①調査・分析の役割(1)

UXデザインにとってユーザやユーザのニーズを知ることは大切



ユーザの情報を知る

属性

・〇〇の属性の人が

心理

・〇〇の心理状態で

行動

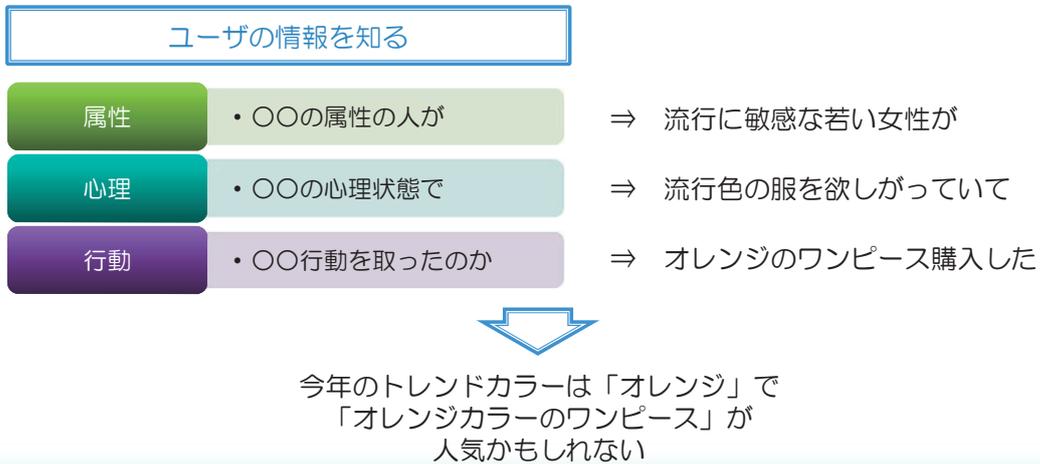
・〇〇行動を取ったのか

57

第2章 デザインプロセス

2.1 UXデザイン

2.1.3 プロセス①調査・分析の役割 (2)



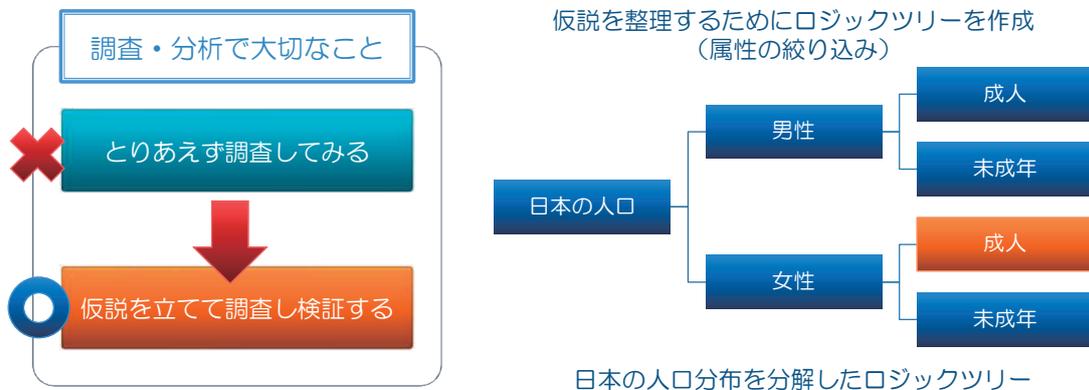
58

第2章 デザインプロセス

2.1 UXデザイン

2.1.4 調査・分析で大切なこと

調査・分析の対象が決まらないから、とりあえず調査を行ってみるはダメ！



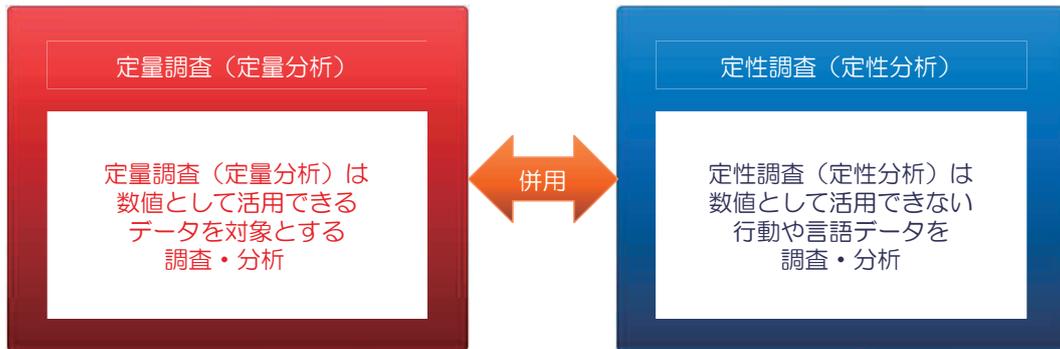
59

第2章 デザインプロセス

2.1 UXデザイン

2.1.5 調査・分析の分類（1）

一般的に調査・分析方法は2つの方法に分けることが可能



60

第2章 デザインプロセス

2.1 UXデザイン

2.1.5 調査・分析の分類（2）

一般的に調査・分析方法は2つの方法に分けることが可能



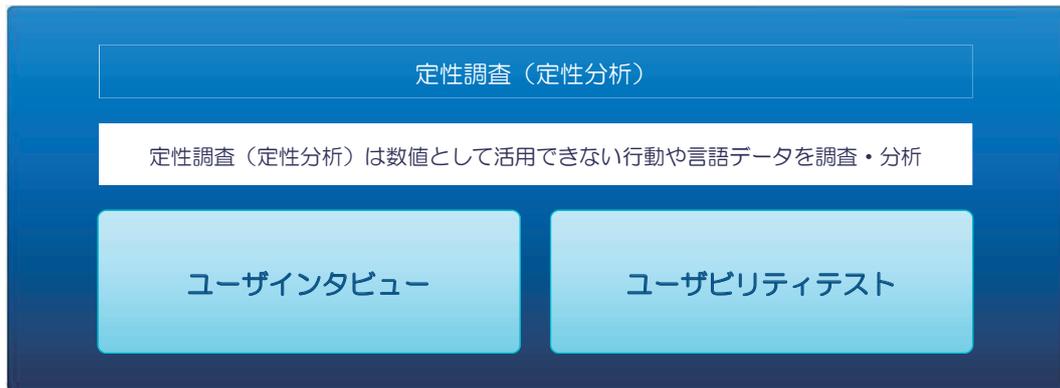
61

第2章 デザインプロセス

2.1 UXデザイン

2.1.5 調査・分析の分類 (3)

一般的に調査・分析方法は2つの方法に分けることが可能



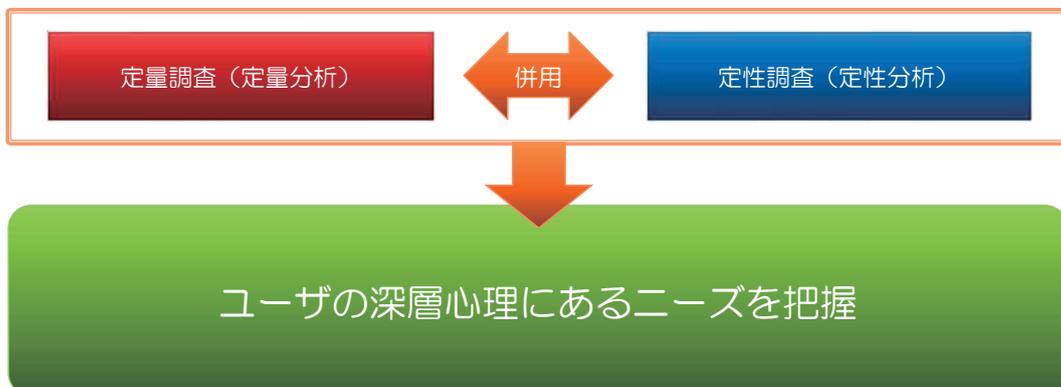
62

第2章 デザインプロセス

2.1 UXデザイン

2.1.5 調査・分析の分類 (4)

両方の「調査・分析」を行うことが重要



63

第2章 デザインプロセス

2.1 UXデザイン

2.1.6 定量調査（定量分析）の種類として（1）



64

第2章 デザインプロセス

2.1 UXデザイン

2.1.6 定量調査（定量分析）の種類として（2）



65

第2章 デザインプロセス

2.1 UXデザイン

2.1.6 定量調査（定量分析）の種類として（3）



アクセス分析

数値を扱う定量情報を得るための手法

Google
アナリティクス

Google
サーチコンソール

66

第2章 デザインプロセス

2.1 UXデザイン

2.1.6 定量調査（定量分析）の種類として（4）



ABテスト

数値を扱う定量情報を得るための手法

A
パターン

B
パターン

67

第2章 デザインプロセス

2.1 UXデザイン

2.1.7 定性調査（定性分析）の種類として（1）

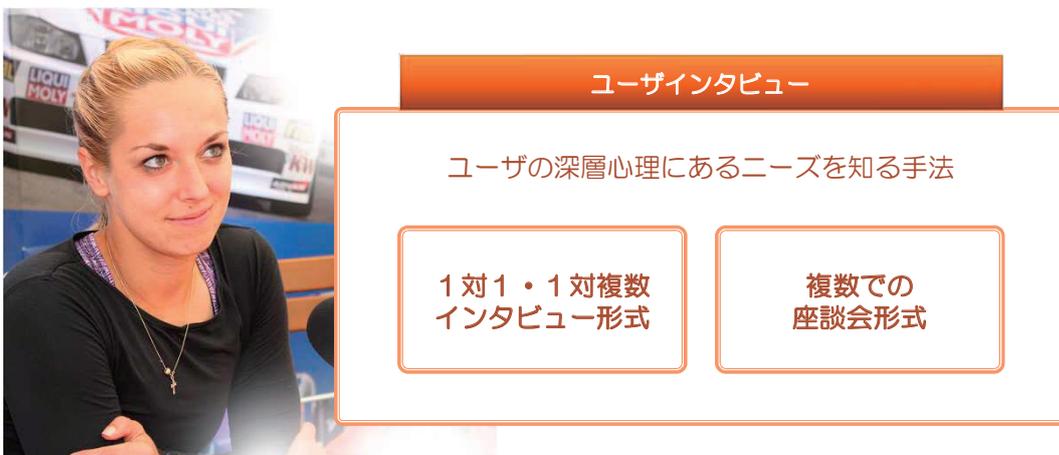


68

第2章 デザインプロセス

2.1 UXデザイン

2.1.7 定性調査（定性分析）の種類として（2）

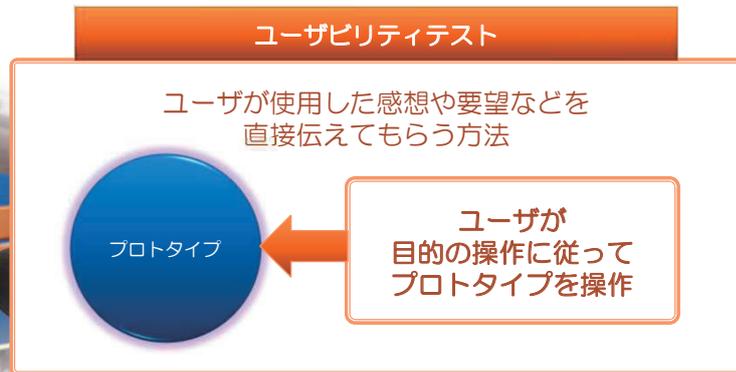
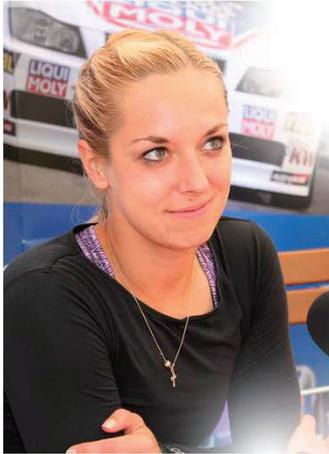


69

第2章 デザインプロセス

2.1 UXデザイン

2.1.7 定性調査（定性分析）の種類として（3）

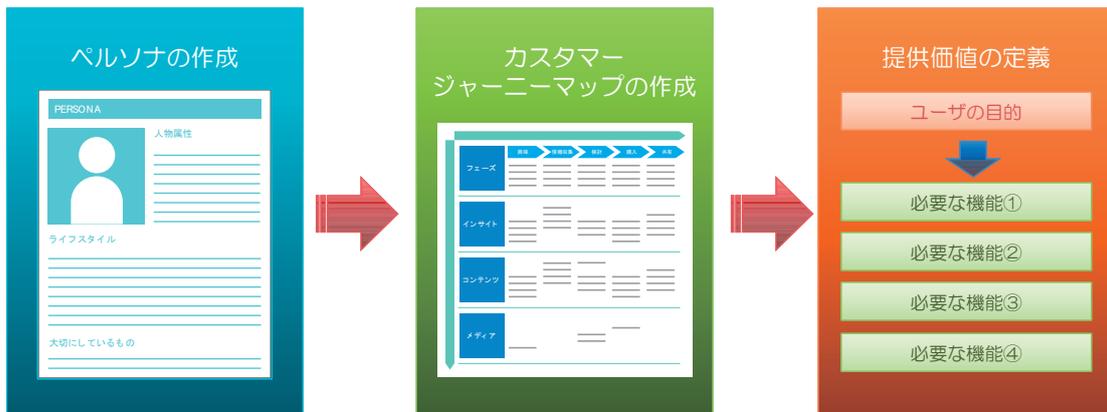


70

第2章 デザインプロセス

2.1 UXデザイン

2.1.8 プロセス②コンセプトデザイン



71

第2章 デザインプロセス

2.1 UXデザイン

2.1.9 ペルソナの作成



PERSONA

人物属性

ライフスタイル

大切にしているもの

72

第2章 デザインプロセス

2.1 UXデザイン

2.1.10 カスタマージャーニーマップの作成 (1)



ユーザの UX (ユーザエクスペリエンス) プロセス

	調査	情報収集	検討	導入	共有
フェーズ	=====	=====	=====	=====	=====
インサイト	=====	=====	=====	=====	=====
行動	=====	=====	=====	=====	=====

ユーザの旅程と行動

73

第2章 デザインプロセス

2.1 UXデザイン

2.1.10 カスタマージャーニーマップの作成 (2)

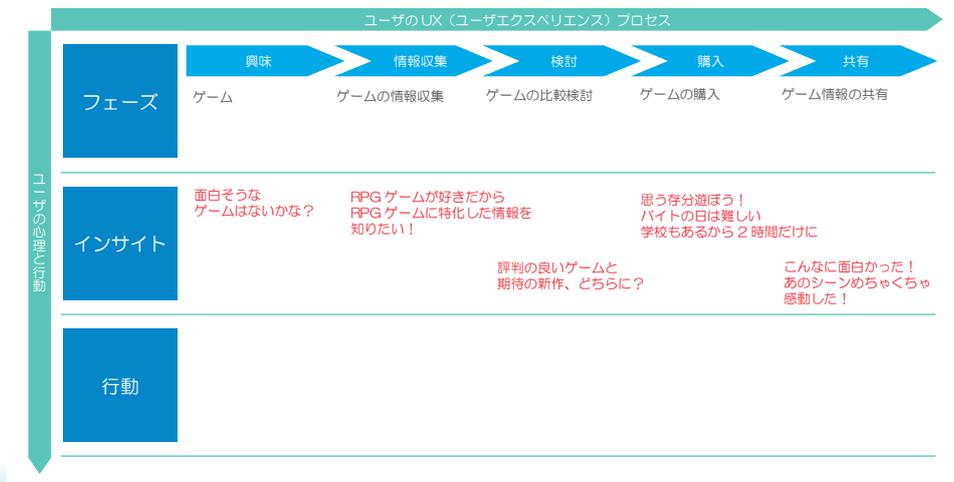


74

第2章 デザインプロセス

2.1 UXデザイン

2.1.10 カスタマージャーニーマップの作成 (3)



75

第2章 デザインプロセス

2.1 UXデザイン

2.1.11 提供価値の定義 (2)



78

第2章 デザインプロセス

2.1 UXデザイン

2.1.12 プロセス③プロトタイプ

ペーパープロトタイプ



デジタルプロトタイプ



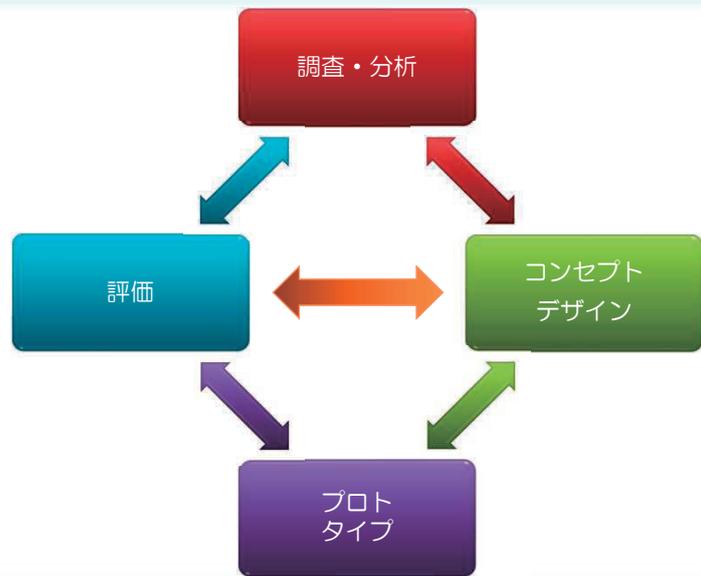
79

第2章 デザインプロセス

2.1 UXデザイン

2.1.13 プロセス④評価

評価でその目的が達成されなければこのプロセスが繰り返されていく

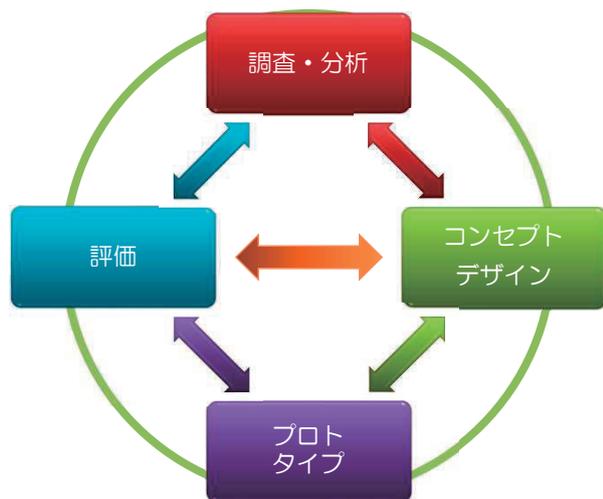


80

第2章 デザインプロセス

2.1 UXデザイン

2.1.14 プロセス⑥提供



81

第2章 デザインプロセス

2.2 ナビゲーション設計

2.2.1 Webサイト全体の構成を把握するため

Webサイトのページ数、ユーザの動き、個々の画面の遷移を把握することが重要



- 1 サイトマップの作成 [全体構造の把握]
- 2 流れ図の作成 [ユーザ行動の定義]
- 3 画面遷移図の作成 [画面関係の定義]

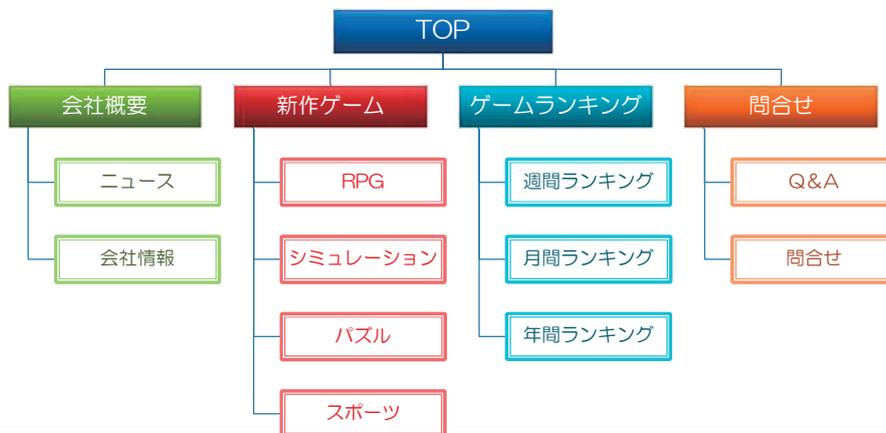
82

第2章 デザインプロセス

2.2 ナビゲーション設計

2.2.2 サイトマップの作成 [全体構造の把握]

Webサイトにて「必要なページ数」「カテゴリごとの整理」を行う



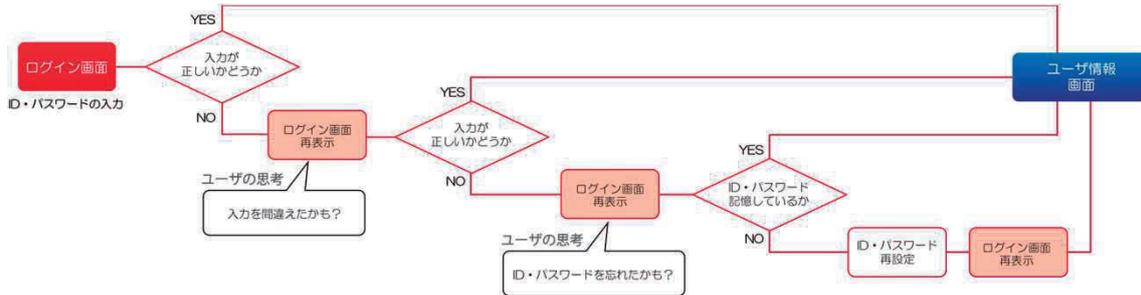
83

第2章 デザインプロセス

2.2 ナビゲーション設計

2.2.3 流れ図の作成 [ユーザ行動の定義]

ID・パスワードを入力してユーザがWebシステムに入る際の流れ図

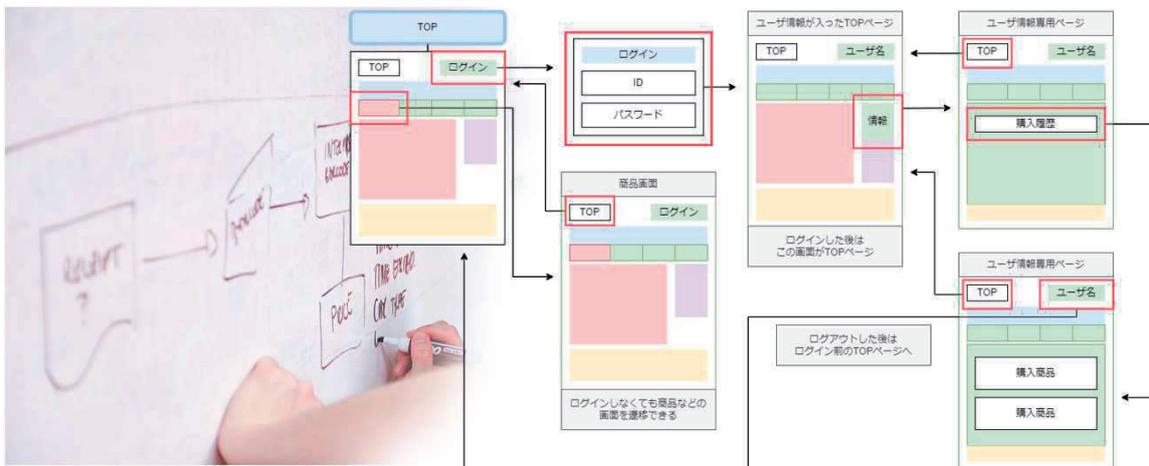


84

第2章 デザインプロセス

2.2 ナビゲーション設計

2.2.4 画面遷移図の作成 [画面関係の定義]



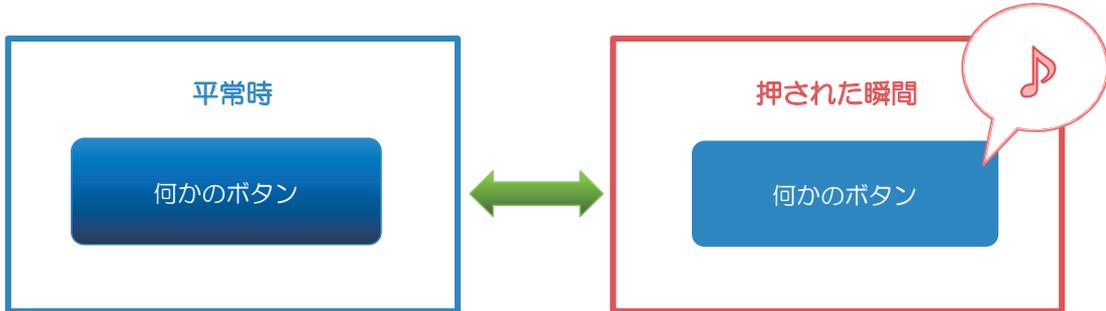
85

第2章 デザインプロセス

2.3 インタラクション（フィードバック）

2.3.1 インタラクション（フィードバック）の必要性

フィードバックがあることでユーザは操作を認識



86

第2章 デザインプロセス

2.3 インタラクション（フィードバック）

2.3.2 フィードバックの種類

フィードバックは3種類

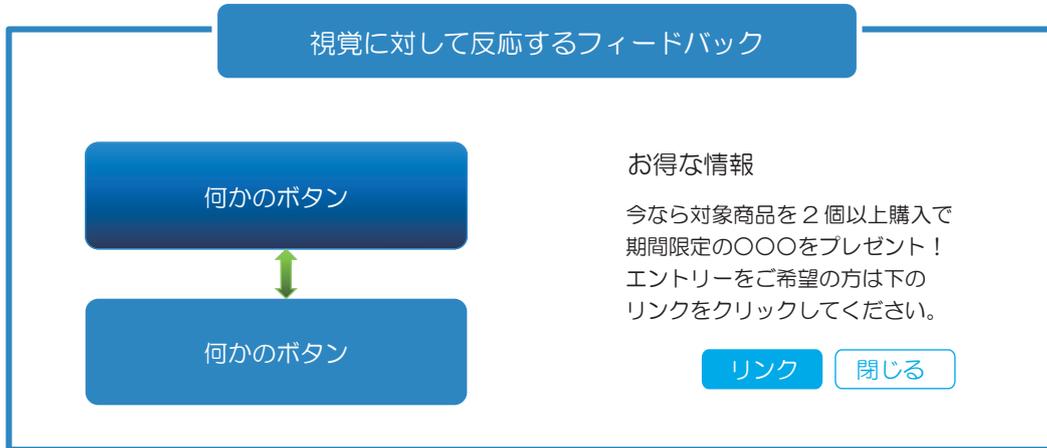


87

第2章 デザインプロセス

2.3 インタラクション（フィードバック）

2.3.3 視覚的フィードバック



88

第2章 デザインプロセス

2.3 インタラクション（フィードバック）

2.3.4 聴覚的フィードバック



89

第2章 デザインプロセス

2.3 インタラクション（フィードバック）

2.3.4 触覚的フィードバック



UI概論

第3章

UI部品

【本章学習内容】

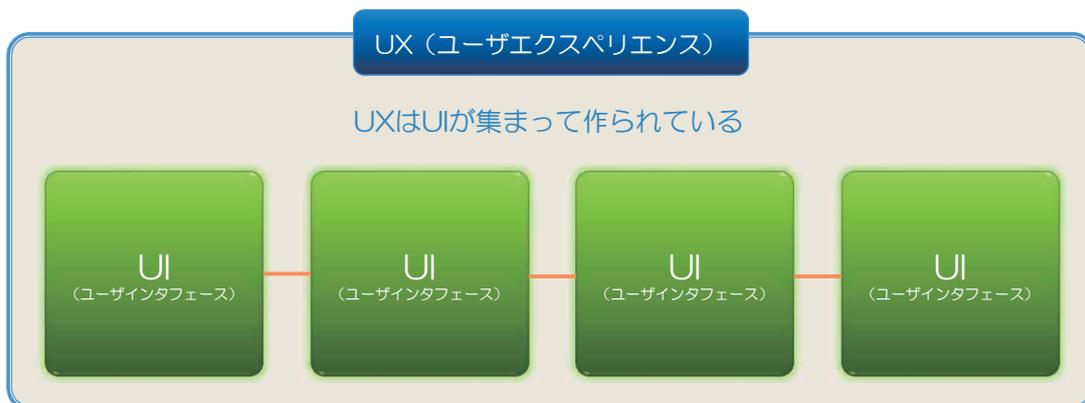
本章では、WebサイトのUIを構築するのに必要なUI部品について学習します。

91

第3章 UI部品

3.1 UXからUIへ

3.1.1 UXとUIの関係を再確認



92

第3章 UI部品

3.2 HTMLで作成できるUI部品（基本）

3.2.1 Webページで利用できるUI部品

HTMLの構造を作るUI部品

- <h2>要素
- <p>要素
- 要素
- <table>要素

商品その①

適当な商品の解説です。商品自体は何を想像していただいてもかまいません。どのような UI 部品があるのかを想像してください。適当な商品の解説です。商品自体は何を想像していただいてもかまいません。どのような UI 部品があるのかを想像してください。

CPU	AMD Ryzen 5 7530U 2GHz/6コア
搭載メモリ	16GB DDR4 PC4-25600
ハードディスク	M.2 SSD 512GB
価格	

93

第3章 UI部品

3.2 HTMLで作成できるUI部品（基本）

3.2.2 イメージ

動物紹介

動物紹介

最新 1 2 3 4 5 >

94

第3章 UI部品

3.2 HTMLで作成できるUI部品（基本）

3.2.3 テーブル

基本スペック			
CPU	AMD Ryzen 5 7530U	画面サイズ	14型(インチ)
	20核/6コア		
画面種類	IPSディスプレイ	解像度	フルHD (1920x1080)
アスペクト比	16:9	タッチパネル	
2in1タイプ		ワイド画面	○
液晶駆動	ノングレア(非光沢)	リフレッシュレート	
メモリ容量	16GB	メモリ規格	DDR4 PC4-25600
メモリスロット (空き)		ストレージ容量	M.2 SSD : 512GB
HDD搭載数			

基本スペック			
CPU	第13世代 インテル Core i9 13900H(Raptor Lake)	画面サイズ	15.6型(インチ)
	14コア		
画面種類		解像度	フルHD (1920x1080)
アスペクト比	16:9	タッチパネル	
2in1タイプ		ワイド画面	○
液晶駆動	ノングレア(非光沢)	リフレッシュレート	60 Hz
メモリ容量	32GB	メモリ規格	DDR4
メモリスロット (空き)	2x1	ストレージ容量	M.2 SSD : 1TB
HDD搭載数			



第3章 UI部品

3.3 HTMLで利用できるUI部品（form）

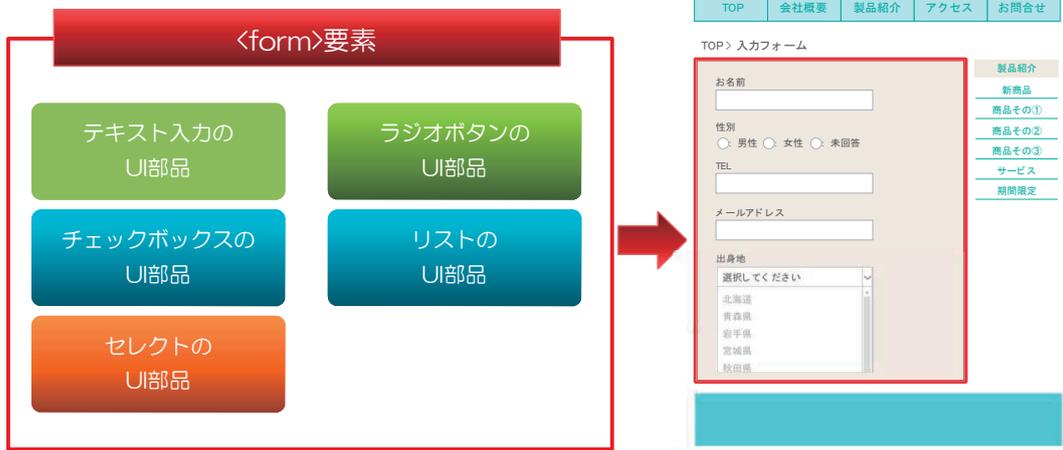
3.3.1 <form>要素内で利用できるUI部品



第3章 UI部品

3.3 HTMLで利用できるUI部品 (form)

3.3.2 入力などのUI部品をまとめる<form>要素



97

第3章 UI部品

3.3 HTMLで利用できるUI部品 (form)

3.3.3 テキスト入力

「form」要素内に記述できる要素の中で、
代表的なテキスト入力を指定するUI部品



98

第3章 UI部品

3.3 HTMLで利用できるUI部品 (form)

3.3.4 ラジオ (選択肢)

掲載されている選択肢のどれか一つしか選ぶことのできないUI部品



The screenshot shows a web form titled 'TOP > 入力フォーム'. It includes a navigation menu with 'TOP', '会社概要', '製品紹介', 'アクセス', and 'お問合せ'. The form fields are: 'お名前' (name), '性別' (gender) with radio buttons for '男性' (male), '女性' (female), and '未回答' (no answer), 'TEL', 'メールアドレス' (email address), and '出身地' (birthplace) with a dropdown menu labeled '選択してください' (please select). A red box highlights the gender section.

99

第3章 UI部品

3.3 HTMLで利用できるUI部品 (form)

3.3.5 チェックボックス (選択肢)

掲載されている選択肢の中から該当する箇所を複数選択可のUI部品



The screenshot shows a web form titled 'TOP > 入力フォーム'. It includes a navigation menu with 'TOP', '会社概要', '製品紹介', 'アクセス', and 'お問合せ'. The form fields are: 'お名前' (name), '性別' (gender) with radio buttons for '男性' (male), '女性' (female), and '未回答' (no answer), and 'お小遣いの使い道' (purpose of pocket money) with checkboxes for '飲食代' (meal), '趣味' (hobby), 'ギャンブル' (gambling), and 'その他' (other). Below this is the '出身地' (birthplace) dropdown menu labeled '選択してください' (please select). A red box highlights the 'お小遣いの使い道' section.

100

第3章 UI部品

3.3 HTMLで利用できるUI部品 (form)

3.3.6 セレクトとリスト (選択肢)

選択肢から該当内容を選ぶUI部品

UI部品の種類は2種類

<select size= “数字” >「数字」の違いで変わる



101

第3章 UI部品

3.4 高度な入力のUI部品 (form)

3.4.1 パスワード入力

入力された文字を

「●」で表示することが可能



102

第3章 UI部品

3.4 高度な入力のUI部品 (form)

3.4.2 数値入力

スピナーで
数値を入力することが可能



TOP 会社概要 製品紹介 アクセス お問い合わせ

TOP > 入力フォーム

お小遣いアンケート

お名前:
山田 太郎

数値を入力:

製品紹介
新商品
商品その①
商品その②
商品その③
サービス
期間限定

The screenshot shows a web form with a navigation bar at the top. The main form area is titled 'TOP > 入力フォーム'. It contains a section 'お小遣いアンケート' with a text input for 'お名前' (filled with '山田 太郎') and a spinner input for '数値を入力'. A red box highlights the spinner input. On the right, there is a sidebar with links for '製品紹介', '新商品', '商品その①', '商品その②', '商品その③', 'サービス', and '期間限定'.

103

第3章 UI部品

3.4 高度な入力のUI部品 (form)

3.4.3 レンジ入力

スライダーで
数値を入力することが可能



TOP 会社概要 製品紹介 アクセス お問い合わせ

TOP > 入力フォーム

お小遣いアンケート

お名前:
山田 太郎

お小遣いの金額:
 25,000

製品紹介
新商品
商品その①
商品その②
商品その③
サービス
期間限定

The screenshot shows a web form with a navigation bar at the top. The main form area is titled 'TOP > 入力フォーム'. It contains a section 'お小遣いアンケート' with a text input for 'お名前' (filled with '山田 太郎') and a range input for 'お小遣いの金額' (set to 25,000). A red box highlights the range input. On the right, there is a sidebar with links for '製品紹介', '新商品', '商品その①', '商品その②', '商品その③', 'サービス', and '期間限定'.

104

第3章 UI部品

3.4 高度な入力のUI部品 (form)

3.4.4 色入力

色選択ダイアログボックスで
色を入力することが可能



105

第3章 UI部品

3.4 高度な入力のUI部品 (form)

3.4.5 日付入力

カレンダー形式ダイアログボックスで
日付を入力することが可能



106

第3章 UI部品

3.4 高度な入力のUI部品 (form)

3.4.6 ファイル選択

ファイル選択ダイアログボックスで
ファイルを入力することが可能



107

第3章 UI部品

3.5 よく使われるUI部品

3.5.1 スクロールバー

決められた大きさから内容がはみ出す場合
「スクロールバー」を表示



108

第3章 UI部品

3.5 よく使われるUI部品

3.5.2 ボタン

The diagram on the left illustrates four types of buttons:

- ボタン** (Buttons): A red header box.
- プライマリボタン** (Primary Button): A white button with a blue border.
- 注文確定** (Order Confirmation): A solid blue button.
- セカンダリボタン** (Secondary Button): A white button with a light blue border.
- キャンセル** (Cancel): A solid light blue button.

The screenshot on the right shows a web form titled "TOP > 入力フォーム". It includes a navigation menu (TOP, 会社概要, 製品紹介, アクセス, お問い合わせ), a "製品紹介" sidebar, and a main form with fields for "お名前", "性別" (radio buttons for 男性, 女性, 未回答), and "お問合せ内容". A "送信" (Send) button is highlighted with a red box, and a red arrow points from the "注文確定" button in the diagram to this button.

109

第3章 UI部品

3.6 UXを高めるUI部品の配置

3.6.1 UI部品の配置で見やすさは変わる

The diagram shows three stages of form layout optimization for a form with the following fields: "お名前", "性別" (radio buttons for 男性, 女性, 未回答), "お小遣いの使い道" (checkboxes for 飲食代, 趣味, ギャンブル, その他), and "出身地" (dropdown menu).

- Stage 1:** The form is in its original state with standard spacing.
- Stage 2:** Red arrows indicate adjustments to the vertical spacing between the "性別" and "お小遣いの使い道" sections, and between "お小遣いの使い道" and "出身地".
- Stage 3:** Red boxes highlight the "お名前" and "お小遣いの使い道" sections, indicating that their vertical spacing has been increased to make them more prominent.

110

UI概論 第4章 UIコンポーネント

【本章学習内容】

本章では、WebサイトのUIを構築するのに必要なUIコンポーネントについて学習します。

111

第4章 UIコンポーネント

4.1 UIコンポーネント

4.1.1 UI部品との違い



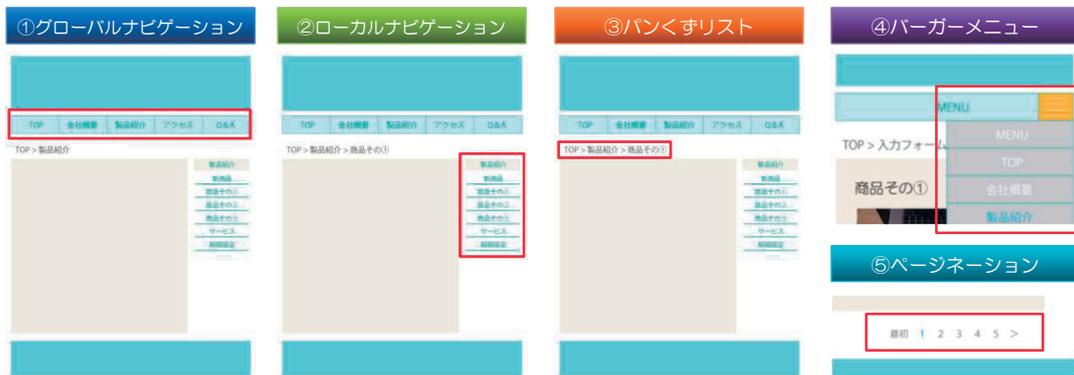
112

第4章 UIコンポーネント

4.2 Webページの構成パーツ（UIコンポーネント）

4.2.1 ナビゲーションの種類

ナビゲーション設計を効果的に行う、5つのナビゲーション



113

第4章 UIコンポーネント

4.2 Webページの構成パーツ（UIコンポーネント）

4.2.2 グローバルナビゲーション



114

第4章 UIコンポーネント

4.2 Webページの構成パーツ (UIコンポーネント)

4.2.3 ローカルナビゲーション

規模の大きなWebサイトでは必須

大きなカテゴリの下位カテゴリ

各カテゴリ内にのみ存在

カテゴリ分けの際に注意が必要



115

第4章 UIコンポーネント

4.2 Webページの構成パーツ (UIコンポーネント)

4.2.4 パンくずリスト

ユーザが見ているページの位置を把握

ユーザの閲覧ページの位置把握

TOPページ以外には必要

ユーザが見ている箇所の近くに設置



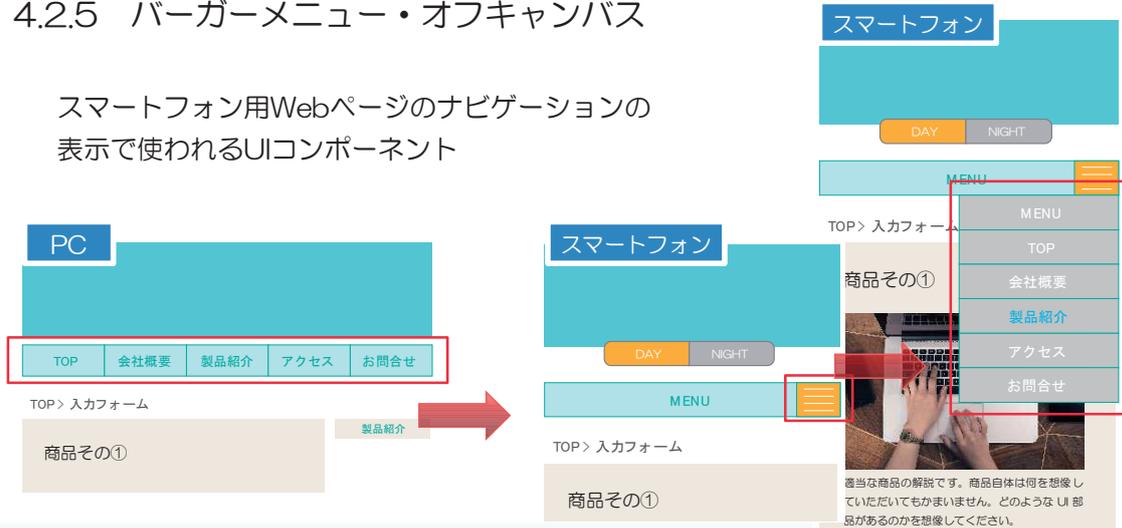
116

第4章 UIコンポーネント

4.2 Webページの構成パーツ (UIコンポーネント)

4.2.5 バーガーメニュー・オフキャンバス

スマートフォン用Webページのナビゲーションの表示で使われるUIコンポーネント



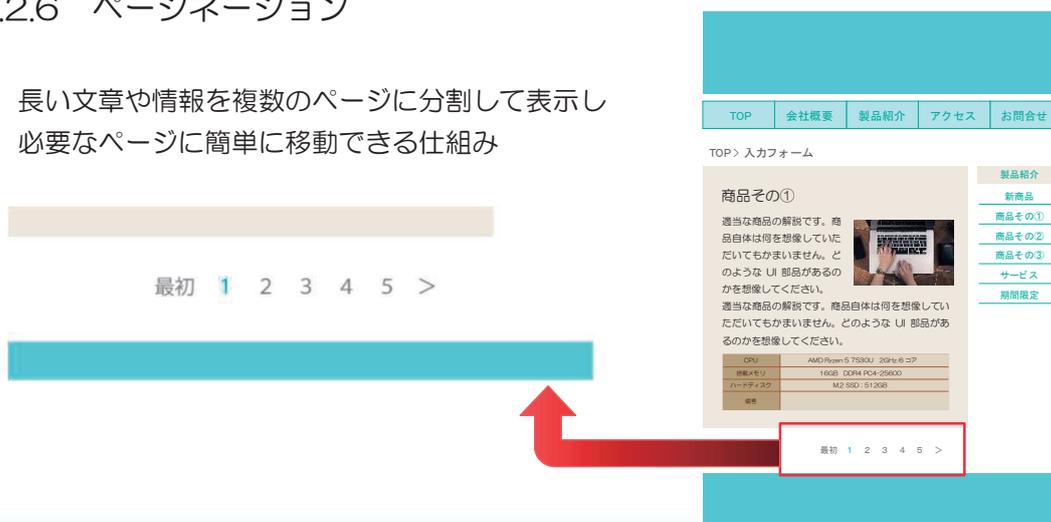
117

第4章 UIコンポーネント

4.2 Webページの構成パーツ (UIコンポーネント)

4.2.6 ページネーション

長い文章や情報を複数のページに分割して表示し
必要なページに簡単に移動できる仕組み



118

第4章 UIコンポーネント

4.2 Webページの構成パーツ (UIコンポーネント)

4.2.7 アコーディオン

ナビゲーションの
ボタンが開閉する
ことで、
目的のページへの
遷移が容易に！



119

第4章 UIコンポーネント

4.2 Webページの構成パーツ (UIコンポーネント)

4.2.8 スイッチ・トグル

ON・OFFの切替など
ボタン一つで切替が可能



120

第4章 UIコンポーネント

4.2 Webページの構成パーツ (UIコンポーネント)

4.2.9 カード

一つひとつの商品などを一枚のカードとして表示することで、注目度に変化が！



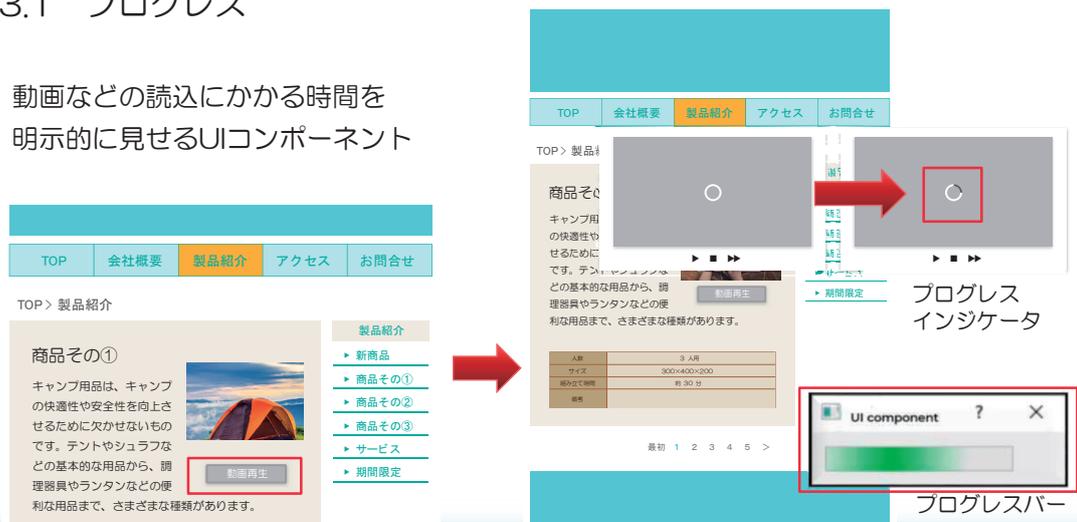
121

第4章 UIコンポーネント

4.3 Webページの動作パーツ (UIコンポーネント)

4.3.1 プログレス

動画などの読込にかかる時間を明示的に見せるUIコンポーネント



122

第4章 UIコンポーネント

4.3 Webページの動作パーツ (UIコンポーネント)

4.3.2 モーダル

操作を行うことで、別のウインドウを立ち上げ、
それ以外の画面を操作できなくしてしまう
UIコンポーネント



123

第4章 UIコンポーネント

4.3 Webページの動作パーツ (UIコンポーネント)

4.3.3 カルousel

Webページの
メイン画像が
推移していく、
UIコンポーネント



124

フロントエンドエンジニア 養成講座

HTML/CSS

1

はじめに

【学習目標】

本テキストでは、Webページを制作するためのプログラムを理解することを目標とします。

目次

第1章	HTML Living Standard·····	003
第2章	文章構造·····	021
第3章	CSS 3·····	098
第4章	CSS 3の設定·····	132

2

HTML/CSS

第1章

HTML Living Standard

【本章学習内容】

本章では、Webページ制作に必要なプログラムHTML Living Standardの記述ルールと基本構文について学習します。

3

第1章 HTML Living Standard

1.1 HTMLファイルの作成方法

1.1.1 HTMLファイルの制作環境

• Webページを制作する際に必要な環境

テキストエディタ：HTMLやCSSのプログラムを作成するソフトウェア
Webブラウザ：HTMLとCSSを解析して画面に表示するソフトウェア
サーバ：HTMLやCSSファイルをインターネットで公開するためのコンピュータ



4

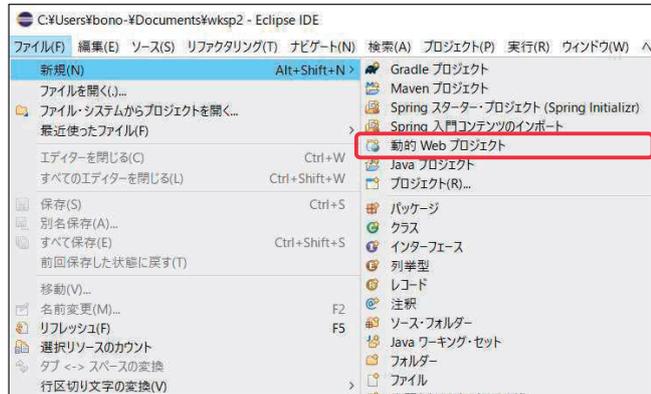
第1章 HTML Living Standard

1.1 HTMLファイルの作成方法

1.1.2 プロジェクト作成

HTMLファイルの作成手順

- Webアプリケーションの作成はプロジェクト単位で行う
- 開発を始める際はメニューの「ファイル」→「新規」→「動的Webプロジェクト」を選択



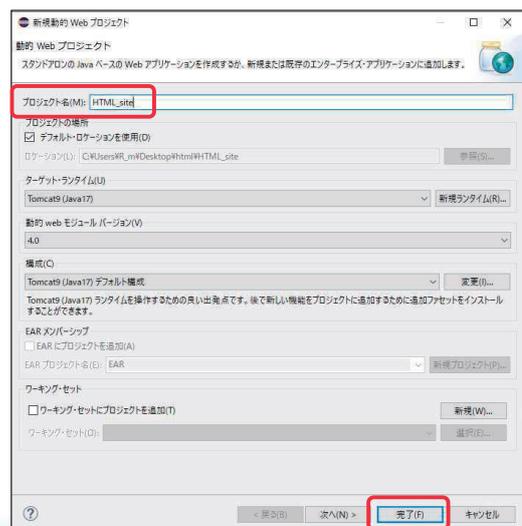
5

第1章 HTML Living Standard

1.1 HTMLファイルの作成方法

1.1.2 プロジェクト作成

- 「プロジェクト名」欄に「HTML_site」とプロジェクト名を入力
- プロジェクト名入力後「完了」ボタンをクリック



6

第1章 HTML Living Standard

1.1 HTMLファイルの作成方法

1.1.3 HTMLファイルの作成

- ファイルを新規作成するにはメニューの「ファイル」→「新規」→「HTMLファイル」を選択
- 作成するプログラムの種類を選択し、ファイルを新規作成



7

第1章 HTML Living Standard

1.1 HTMLファイルの作成方法

1.1.3 HTMLファイルの作成

- 作成するHTMLファイル名欄に「index.html」と入力し完了ボタンをクリック



8

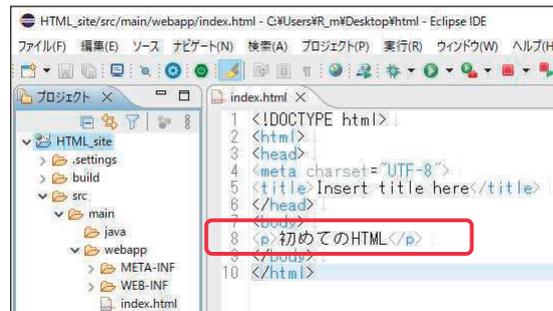
第1章 HTML Living Standard

1.1 HTMLファイルの作成方法

1.1.3 HTMLファイルの作成

- HTMLファイルが新規作成される
- 8行目に空行があるので下記内容を入力

<p>初めてのHTML</p>



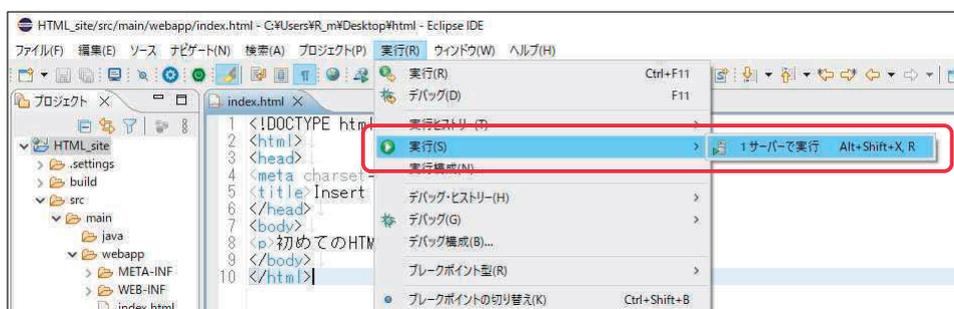
9

第1章 HTML Living Standard

1.1 HTMLファイルの作成方法

1.1.4 HTMLファイルをWebブラウザで表示する

- HTMLファイルを表示するにはメニューの「実行」→「実行」→「サーバで実行」を選択



10

第1章 HTML Living Standard

1.1 HTMLファイルの作成方法

1.1.4 HTMLファイルをWebブラウザで表示する

- 表示に使用するWebサーバ (Tomcat) を選択
- どれでも表示できるが最もバージョンが新しいものを選択



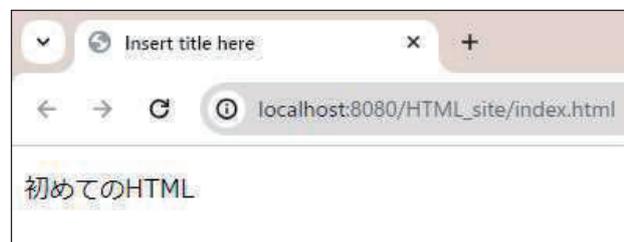
11

第1章 HTML Living Standard

1.1 HTMLファイルの作成方法

1.1.4 HTMLファイルをWebブラウザで表示する

- HTMLファイルの記述内容がWebブラウザで表示
- Eclipse起動中はHTMLファイルを変更し、再度表示させる場合HTMLファイルを保存した後に再読み込みで変更が反映



12

第1章 HTML Living Standard

1.2 HTML Living Standardの概要

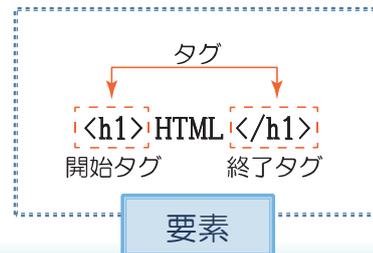
1.2.1 HTMLプログラムの記述ルール

- HTMLはマークアップ言語のひとつ
- <> 不等号で挟まれているマークをHTMLでは、「タグ」という
- テキストに「タグ」をマークアップして文章構造の意味を付加する 例1)
- タグは開始タグと終了タグで一つの要素 例2)
- HTMLの中には、終了タグが省略できるものもある
- HTMLタグで文章の意味付けをした固まりを「要素」(Element)という 例2)

例1)

```
01 <body>
02 <h1>HTML</h1>
03 <p>はじめてのHTML</p>
04 </body>
```

例2)



13

第1章 HTML Living Standard

1.2 HTML Living Standardの概要

1.2.1 HTMLプログラムの記述ルール

- HTMLの要素には、「属性」がある
- 「属性」とは、要素の付加情報を設定する機能
- 「属性」の書き方は、
属性名=「属性値」と書く 例1)
- 「属性」は、要素に対し同じ属性を複数指定できない
- 各要素には、指定できる属性が決まっている
- HTMLの要素に対して複数の属性を指定する場合は、半角スペースを使って区切る 例2)

例1)

```
<html lang="ja"> ~ </html>
```

例2)

```

```

14

第1章 HTML Living Standard

1.2 HTML Living Standardの概要

1.2.2 空要素、親要素・子要素

- HTMLの要素には、終了タグを持たない「**空要素**」という要素もある 例1)
- HTMLの要素は要素の中に違う要素を記述にすることが出来る 例2)
- 例3の様なHTML要素の配置は正しい入れ子状態ではないので間違った記述である
- em要素の外側にあるp要素がem要素から見て「**親要素**」という 例2)
- P要素の中にあるem要素は、p要素から見て「**子要素**」という 例2)

例1) 空要素

```

```

例2) タグの入れ子

```
<P> はじめての <em>HTML</em> </p>
```

子要素

親要素

例3) タグの入れ子の悪い例

```
<em> <P> はじめてのHTML </em> </p>
```

15

第1章 HTML Living Standard

1.2 HTML Living Standardの概要

1.2.3 コメント

- プログラムソースに注釈や説明文を入れる場合は、「**コメントタグ**」をつかう
- <!-- と -->で囲むことでその範囲を「**コメント**」として認識する 例1)
- コメントは、Webブラウザには表示されない
- 一時的にWebページに表示させたくないタグに対してコメントタグで囲むことを「**コメントアウト**」という 例2)

例1) コメントタグ

```
<!-- コメント -->
```

例2) コメントタグ入力

```
01 <article><!-- start_contents -->  
02  
03 <!-- <P>ソースを省略</p> -->  
04 コメントアウト  
05 </article><!-- end_contents -->
```

16

第1章 HTML Living Standard

1.2 HTML Living Standardの概要

1.2.4 HTMLプログラムの基本構文

- 1行目 `<!DOCTYPE html>`
 - DTD宣言
「DTD」とは「Document Type Definition」の略で記述しているHTMLのバージョンを指定する
- 2行目 `<html>` ~ 10行目`</html>`
 - HTMLタグ
HTML文章のルート要素のタグ、このタグが囲っている範囲が、HTMLを記述する場所
Lang属性の設定が推奨されている

```
<html lang="jp">
```

Lang属性

```
01 <!DOCTYPE html>
02 <html>
03 <head>
04 <meta charset="UTF-8">
05 <title>Insert title here</title>
06 </head>
07 <body>
08     <!-- 省略 -->
09 </body>
10 </html>
```

例) 1.1.3で作成したHTMLファイルソース

17

第1章 HTML Living Standard

1.2 HTML Living Standardの概要

1.2.4 HTMLプログラムの基本構文

- 3行目 `<head>` ~ 6行目`</head>`
 - headタグ
ページ内の文章のタイトル・付属情報、技術情報を明示する場所
- 4行目 `<meta charset="UTF-8">`
 - metaタグ
文章の作者・Webページの説明文・エンコーディングの指定などをとするタグ

```
01 <!DOCTYPE html>
02 <html>
03 <head>
04 <meta charset="UTF-8">
05 <title>Insert title here</title>
06 </head>
07 <body>
08     <!-- 省略 -->
09 </body>
10 </html>
```

例) 1.1.3で作成したHTMLファイルソース

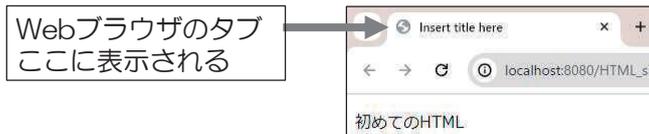
18

第1章 HTML Living Standard

1.2 HTML Living Standardの概要

1.2.4 HTMLプログラムの基本構文

- 5行目 `<title>` ~ `</title>`
titleタグ
Webページ内の文章のタイトルを設定する
Webブラウザのタブに表示される



- 7行目 `<body>` ~ 9行目 `</body>`
 - bodyタグ
文章の内容を表す
Webブラウザに表示される内容

```
01 <!DOCTYPE html>
02 <html>
03 <head>
04 <meta charset="UTF-8">
05 <title>Insert title here</title>
06 </head>
07 <body>
08   <!-- 省略 -->
09 </body>
10 </html>
```

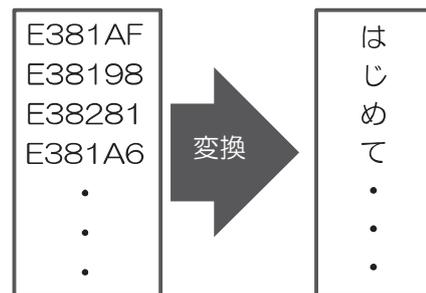
例) 1.1.3で作成したHTMLファイルソース

第1章 HTML Living Standard

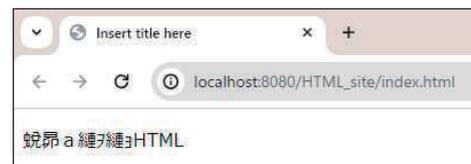
1.2 HTML Living Standardの概要

1.2.5 文字コード・エンコーディング

- **文字コード**とは、コンピュータ上で文字や記号の一文字ごとに割り振られている番号のこと
- 文字コードには、JIS (ジス) コード、Shift_JIS (シフトジス)、EUC、Unicode (ユニコード) UTF-8などがある
- **エンコーディング**とは、文字コードを規則に基づきコンピュータ上に表示できる文字に変換すること
- 世界的に今最も普及しているエンコーディングは、UTF-8である



エンコーディング



例1)文字化け

HTML/CSS

第2章

文章構造

本章学習内容

本章では、HTMLの文章構造を構築していくためのプログラムについて学習します。

21

第2章 文章構造

2.1 文章構成

2.1.1 見出し (h1~h6タグ)

- h1~h6要素は、文章の大見出し・小見出し、小見出しの意味を付加する要素
- h1~h6までの要素があり、h1要素が一番大きい文字サイズで表示され、数字が大きくなるほど文字サイズが小さくなる 例1)
- h1~h6要素で文章の構造（アウトライン）が形成される
- h1要素、前後に1行分の空白行がはいり、最後に段落終了の改行が入る

例1)

```
<h1>見出し 1</h1>  
<h2>見出し 2</h2>  
<h3>見出し 3</h3>  
<h4>見出し 4</h4>  
<h5>見出し 5</h5>  
<h6>見出し 6</h6>
```

22

第2章 文章構造

```
:
08 <!-- ----メインコンテンツ---- -->
09 <h1>見出し 1</h1>
10 <h2>見出し 2</h2>
11 <h3>見出し 3</h3>
12 <h4>見出し 4</h4>
13 <h5>見出し 5</h5>
14 <h6>見出し 6</h6>
15 <!-- ----メインコンテンツ---- -->
:
```

• 実行結果



23

第2章 文章構造

2.1 文章構成

2.1.2 本文(段落) 改行、区切り線 (pタグ brタグ、hrタグ)

- p要素は、「Paragraph」(パラグラフ)の略で文章本文(段落)の意味を付加する要素 例1)
- p要素は、前後に1行分の空白行がはいる、最後に段落終了の改行が入る
- 段落の終わりの改行でない通常の改行は、改行を表す空要素のbr要素を使う 例2)
- hr要素は、段落の区切りの罫線を表す空要素

例1)

```
<p>本文 (段落) テキスト</p>
<hr>
```

例2)

```
<p>本文 (段落) <br>改行を記述</p>
<hr>
```

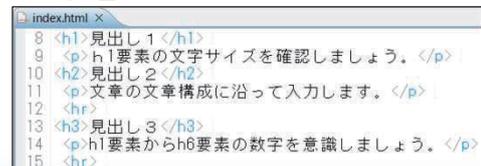
24

第2章 文章構造

:

```
09 <h1>見出し 1</h1>
10 <p>h1要素の文字サイズを確認しましょう。</p>
11 <h2>見出し 2</h2>
12 <p>文章の文章構成に沿って入力します。</p>
13 <hr>
14 <h3>見出し 3</h3>
15 <p>h1要素からh6要素の数字を意識しましょう。</p>
16 <hr>
```

• 入力内容



```
index.html x
8 <h1>見出し 1</h1>
9 <p>h1要素の文字サイズを確認しましょう。</p>
10 <h2>見出し 2</h2>
11 <p>文章の文章構成に沿って入力します。</p>
12 <hr>
13 <h3>見出し 3</h3>
14 <p>h1要素からh6要素の数字を意識しましょう。</p>
15 <hr>
```

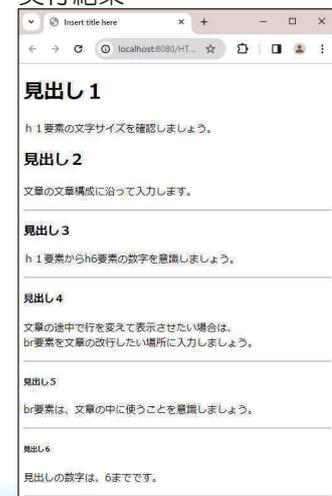
25

第2章 文章構造

:

```
17 <h4>見出し 4</h4>
18 <p>文章を途中で行を変えて表示させたい場合は、<br>
19 br要素を文章の改行したい場所に入力しましょう。</p>
20 <hr>
21 <h5>見出し 5</h5>
22 <p>br要素は、文章の中に使うことを意識しましょう。</p>
23 <hr>
24 <h6>見出し 6</h6>
25 <p>見出しの数字は、6までです。</p>
26 <hr>
```

• 実行結果



見出し 1

h1要素の文字サイズを確認しましょう。

見出し 2

文章の文章構成に沿って入力します。

見出し 3

h1要素からh6要素の数字を意識しましょう。

見出し 4

文章の途中で行を変えて表示させたい場合は、br要素を文章の改行したい場所に入力しましょう。

見出し 5

br要素は、文章の中に使うことを意識しましょう。

見出し 6

見出しの数字は、6までです。

26

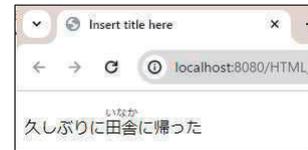
第2章 文章構造

2.1 文章構成

2.1.3 ルビ(フリガナ) (rubyタグ、rtタグ、rpタグ)

- **ruby**要素は、テキストのルビである意味を表す要素 例1)
- **rt**要素は、ruby要素のルビのテキスト部分を表す
- **rt**要素は、ruby要素の子要素として記述する
- **rp**要素は、ルビテキストを囲む括弧を表す要素
ruby要素が対応していないWebブラウザにも対応できるように、
rp要素を設定する 例2)

例1) ルビ



例2) 非対応のルビ



```
<p>久しぶりに<ruby>田舎<rp></rp><rt>いなか</rt><rp></rp></ruby>に帰った</p>
```

ルビテキスト

rp要素(非対応の場合の設定)

27

第2章 文章構造

2.1 文章構成

2.1.3 ルビ(フリガナ) (rubyタグ、rtタグ、rpタグ)

- ルビは、一語ずつに読みを振りつけていく 例1)
- 読みを一語ずつ振り分けるのが難しい場合は、グループルビとしてまとめて指定する 例2)

例1) ルビを一語ずつ割り振る

```
<p><ruby>風<rt>ふう</rt>林<rt>りん</rt>火<rt>か</rt>山<rt>さん</rt></ruby></p>
```

例2) グループルビ(一語ずつ割り振れない場合)

```
<p>久しぶりに<ruby>田舎<rt>いなか</rt></ruby>に帰った</p>
```

28

第2章 文章構造

```
01 <!DOCTYPE html>
02 <html lang="ja">
03 <head>
04 <meta charset="UTF-8">
   .
   .
28 <hr>
29 <!-- ---メインコンテンツ--- -->
30 </body>
31 </html>
```

参考サイト ウィキペディア : <https://ja.wikipedia.org/wiki/>

例1) 入力する要素一覧

```
<ruby> ~ </ruby>
<rp> ~ </rp>
<rt> ~ </rt>
```



• 実行結果



29

第2章 文章構造

2.1 文章構成

2.1.4 上付き文字、下付き文字 (supタグ、subタグ)

- **sup要素**は、上付きテキストを設定するときの要素
 - 例) x^2 を解きなさい。

```
<p>x<sup>2</sup>を解きなさい。</p>
```

sup要素

- **sub要素**は下付きテキストを設定するときの要素
 - 例) CO₂が温暖化の原因です。

```
<p>CO<sub>2</sub>が温暖化の原因です。</p>
```

sub要素

30

第2章 文章構造

:

```
18 <hr>
19 <h4>ラッピング</h4>
20 <p>クラフト紙厚口 70g/m<sup>2</sup></p>
21 <p>クラフト紙 50g/m<sup>2</sup></p>
22 <p>クラフト紙(グレー) 50g/m<sup>2</sup></p>
23 <hr>
24 <h4>地球温暖化対策(SDGs) </h4>
25 <p>「サンパチェンス」は、1株で鉢植えの場合約60cm、地球温暖化の原因と
    される二酸化炭素(CO<sub>2</sub>)や、おもに自動車などの排気ガスに
    あるホルムアルデヒド(HCHO)できわめて高い吸収能力を発揮することが
    実証されている。</p>
26 <hr>
```

:

・実行結果



31

第2章 文章構造

2.1 文章構成

2.1.5 特殊文字

- 文章内で使用する記号は**特殊文字**を使う
- よく使用する特殊文字は、「©(コピーライト)」や「() (括弧)」「&(アンド)」がある
- 特殊文字を使用することで記号の文字化けを防ぐことができる

特殊文字	記号	意味	特殊文字	記号	意味
©	©	著作権記号(コピーライト)	"	“	ダブルクォーテーション
&	&	アンド	'	'	シングルクォーテーション
((左カッコ	×	×	掛け算
))	右カッコ	÷	÷	割り算
<	<	小なり	=	=	イコール
>	>	大なり	¥	¥	円
 		ノーブレイクスペース半角	%	%	パーセント

32

第2章 文章構造

```
01 <!DOCTYPE html>
02 <html lang="ja">
03 <head>
04 <meta charset="UTF-8">
   .
   .
32 </body>
33 </html>
```

• 実行結果



33

第2章 文章構造

2.1 文章構成

2.1.6 リスト (ulタグ、liタグ)

- ul要素は、箇条書きを表す要素
- ul要素の直下に別のul要素を入れることはできない
- ul要素の中には、li要素を複数記述することができる 例1)
- リストを入れ子にすることが出来るが、必ずli要素の子要素にul要素を記述する 例2)

例1) リスト

```
<ul>
  <li>Top</li>
  <li>ビジネスカバン</li>
  <li>トートバッグ</li>
  <li>リュックサック</li>
  <li>ポーチ</li>
</ul>
```

例2) リスト入れ子良い例

```
<ul>
  <li>Top
    <ul><li>カテゴリ</li></ul>
  </li>
  <li>ビジネスカバン</li>
  . . . <省略> . . .
</ul>
```

例3) リスト入れ子悪い例

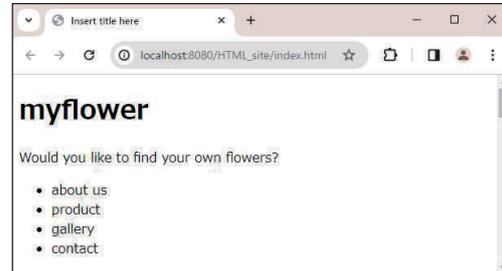
```
<ul>
  <li>Top</li>
  <ul><li>カテゴリ</li></ul>
  <li>ビジネスカバン</li>
  . . . <省略> . . .
</ul>
```

34

第2章 文章構造

```
01 <!DOCTYPE html>
02 <html lang="ja">
03 <head>
04 <meta charset="UTF-8">
   .
   .
50 </body>
51 </html>
```

• 実行結果（ヘッダー）



• 実行結果（フッター）



35

第2章 文章構造

2.1 文章構成

2.1.7 リスト（olタグ、liタグ）

- ol要素は、項目の順序に意味があるリストを表す要素 例1)
- ol要素の直下に別のol要素を入れることはできない
- ol要素の中には、li要素を複数記述することができる
- 箇条書きを入れ子にすることが出来るが、必ずli要素の子要素にol要素を記述する
- ol要素にstart属性で開始番号を指定することができる
- ol要素にreversed属性で番号を降順に指定することができる

例1) リスト

```
<ol>
  <li>小麦粉を入れる</li>
  <li>卵をいれる</li>
  <li>2分混ぜる</li>
  <li>枠に入れる</li>
  <li>オーブンで焼く</li>
</ol>
```

36

第2章 文章構造

```
:
22 <h3>購入方法</h3>
23 <p>購入手順は、下記の通りです。</p>
24 <ol>
25 <li>Webページにアクセス</li>
26 <li>申込フォームに入力</li>
27 <li>購入金額を振込する</li>
28 <li>商品の発送</li>
29 </ol>
30 <hr>
:
```

• 実行結果



37

第2章 文章構造

2.1 文章構成

2.1.8 画像表示 (imgタグ)

- **img要素**は、文章に画像を埋め込むことができる
- img要素の**src属性**の記述は必須
- src属性には、埋め込む画像データの保存先「パス」を設定する
- その他の記述できる属性は、**alt属性**、**width属性**、**height属性**などがある
- alt属性は、画像が表示されなかった際、代わりに表示されるテキストを設定する要素
- width要素は「幅」、height要素は、「高さ」を指定する属性値は整数のみを設定、単位は記述しない

例1)

```

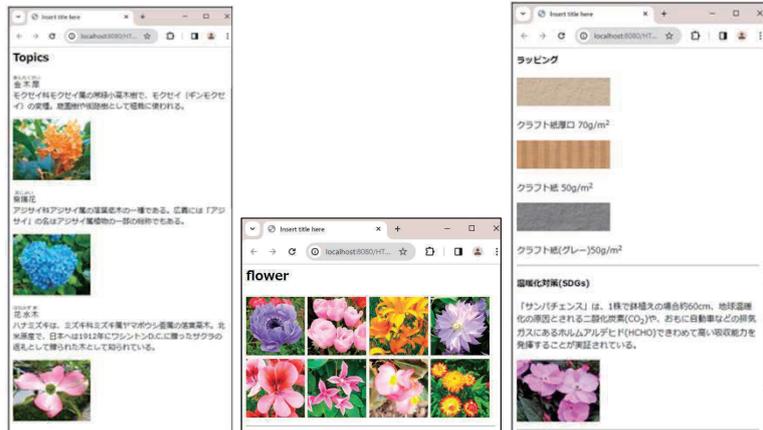
```

38

第2章 文章構造

```
01 <!DOCTYPE html>
02 <html lang="ja">
03 <head>
04 <meta charset="UTF-8">
   .
   .
72 </body>
73 </html>
```

• 実行結果



39

第2章 文章構造

2.1 文章構成

2.1.9 絶対パス、相対パス

- 「パス」とは、経路・道を意味する英単語でファイルが、コンピュータのどの場所に保存されているかを示した文字列のこと
- パスには、「絶対パス」と「相対パス」がある
- 絶対パスは、ディレクトリの階層構造の頂点からすべての道筋を記述したパス 例1)
- 他のWebサイトのリンクは絶対パスで記述する
- 「index」というファイル名は、パスを記述する際は、省略できる

例1) 絶対パス



40

第2章 文章構造

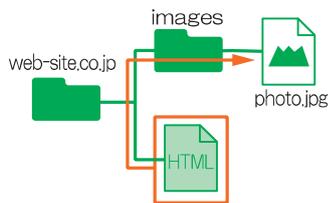
2.1 文章構成

2.1.9 絶対パス、相対パス

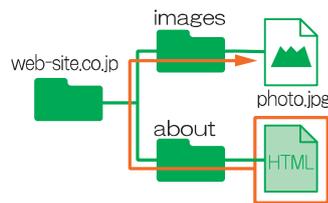
- 相対パスは、現在作業しているファイルの位置からの道筋を記述したパス (例1)
- 自分のWebサイト内のリンクは相対パスで記述する
- 作業対象となっているファイルを起点とし、フォルダを一つ上の階層に上がる場合のパスは、「../」で表す

例1) 相対パス

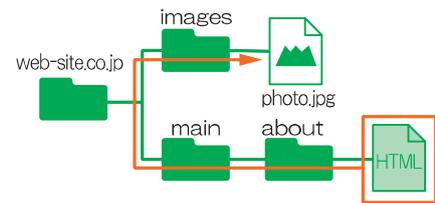
```
../images/photo.jpg
```



例1) images/photo.jpg



例2) ../images/photo.jpg



例3) ../../images/photo.jpg

41

第2章 文章構造

2.1 文章構成

2.1.10 図表、写真、挿絵 (figureタグ、figcaptionタグ)

- figure要素は、図表、写真、挿絵などの集まりを表す要素 (例1)
- figcaption要素は、図表や写真、挿絵などの説明文を表す要素
- figure要素の最初の子要素、または、最後の子要素にfigcaption要素を記述する
- figcaption要素を記述するかは任意である

例1)

```
<figure>  
  
<figcaption>株式会社〇〇〇</figcaption>  
</figure>
```

42

第2章 文章構造

```
01 <!DOCTYPE html>
02 <html lang="ja">
03 <head>
04 <meta charset="UTF-8">
   .
   .
114 </body>
115 </html>
```

• 実行結果



43

第2章 文章構造

2.1 文章構成

2.1.1.1 ハイパーリンク(リンク) (aタグ)

- a要素は、現在のWebページから他のWebページに移動(ハイパーリンク、略して「リンク」ともいう)を設定するための要素
- a要素にhref属性を記述することでハイパーリンクとして設定できる 例1)
- a要素は、href属性を指定しない場合は、ダミーリンクとして設定できる 例2)
- a要素にtarget属性の「_blank」を設定することで新しいウィンドウやタブにリンク先のWebページを表示させることができる 例3)

例1) ハイパーリンク

```
<a href="http://html-site.co.jp">web-site</a>
```

例2) ダミーリンク

```
<a>html-site</a>
```

例3) target属性の設定

```
<a href="http://html2.co.jp" target="_blank">他のサイトへ</a>
```

44

第2章 文章構造

2.1 文章構成

2.1.1.1 ハイパーリンク(リンク) (aタグ)

- a要素にdownload属性を記述するとリンク先のファイルをダウンロードすることを表す 例1)
- a要素にhref属性が省略されている場合、download属性も省略しなければならない
- download属性に値を設定した場合、ダウンロードするファイルのファイル名を指定できる 例2)

例1) download属性の設定

```
<a href="sample.pdf" download>PDFファイルダウンロード</a>
```

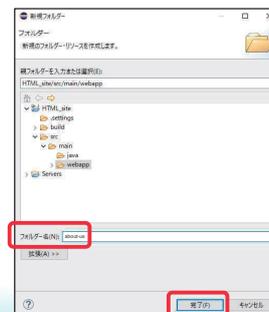
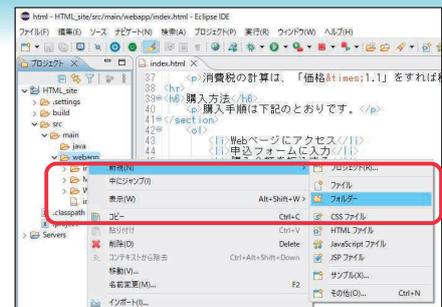
例2) download属性値の設定

```
<a href="sample.pdf" download="マニュアル">PDFファイルダウンロード</a>
```

45

第2章 文章構造

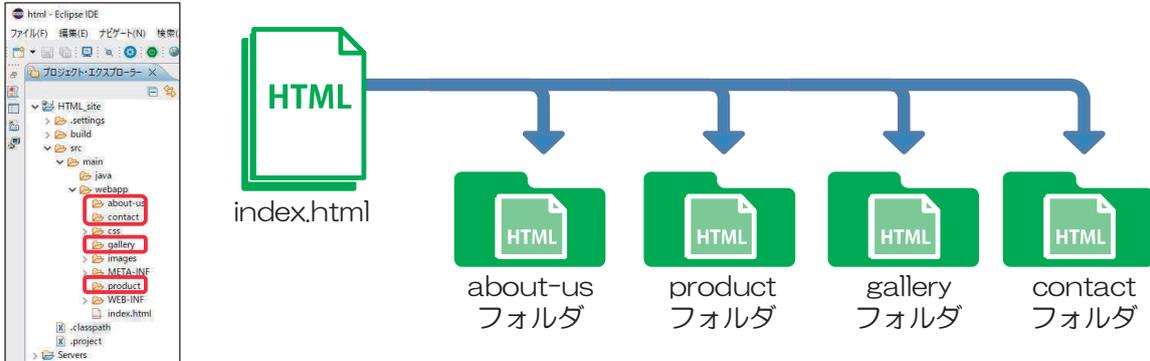
- ① リンク先のWebページを作成
「webapp」フォルダの中に
「about-us」「product」「gallery」「contact」
の4つの新規フォルダを作成
- ② 「HTML_site」プロジェクト内の
「src」フォルダの「main」フォルダの
中の「webapp」フォルダにカーソルを
合わせ右クリック
ショートカットメニューから「新規」の
「フォルダ」をクリック
- ③ 新規フォルダのウィンドウが表示される
フォルダ名欄に「about-us」と入力
「完了」をクリック
同じように他のフォルダも作成



46

第2章 文章構造

④前項で使用しているindex.htmlファイルをそれぞれのフォルダに複製



例1) フォルダー一覧

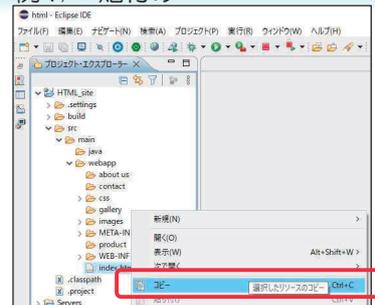
⑤Webページを移動した際、違うWebページだと確認できるように複製したindex.htmlのh1要素のテキストを変更、それぞれのフォルダ名と同じ名前に書き替え
また、h1要素の次のh2要素から、「購入方法」のリストまでを削除

47

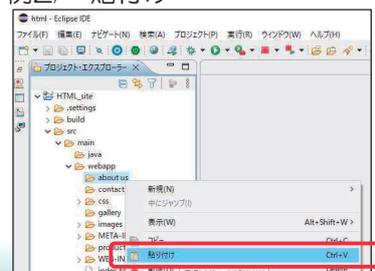
第2章 文章構造

- ① ファイルの複製はプロジェクトエクスプローラーの「index.html」にカーソルを合わせて右クリック
- ② ショートカットメニューから「コピー」をクリック
- ③ 貼り付ける先のフォルダ名にカーソルを合せ右クリック
- ④ ショートカットメニューから「貼り付け」をクリック
- ⑤ 他のフォルダにも同じ手順で「index.html」を複製

例1) 貼付け



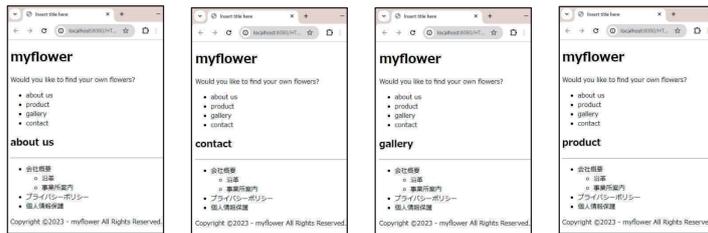
例2) 貼付け



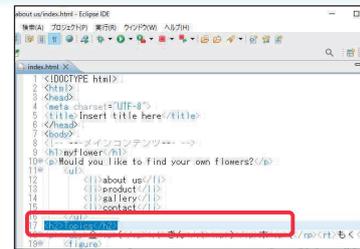
48

第2章 文章構造

- ① 複製した「index.html」を開く
- ② Webページの最初のh2要素のテキストをフォルダ名と同じ名前に打ち換える
- ③ h2要素の次のp要素からh2要素の次のp要素の終了タグまでを削除
- ④ 同じように他のフォルダの「index.html」を変更



例2) 各Webページ表示



<< 省略 >>



例1) プログラム編集

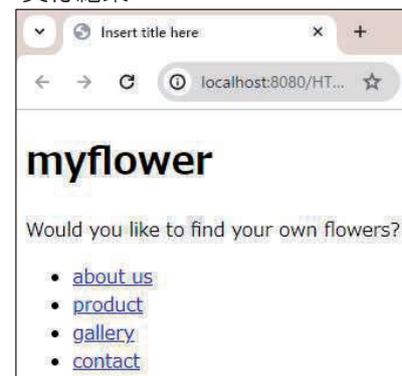
第2章 文章構造

<<ヘッダー>>

```

:
09 <h1>myflower</h1>
10 <p>Would you like to find your own flowers?</p>
11 <ul>
12 <li><a href= "about-us/" >about us</a></li>
13 <li><a href= "product/" >product</a></li>
14 <li><a href= "gallery/" >gallery</a></li>
15 <li><a href= "contact / " >contact</a></li>
16 </ul>
:
    
```

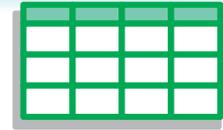
• 実行結果



第2章 文章構造

2.1 文章構成

2.1.1.2 表組（テーブル）



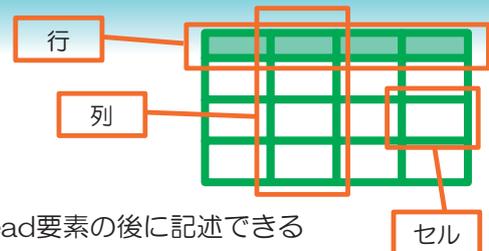
- **table要素**は、表組を表す要素
- **caption要素**は、表組のタイトル、説明文を表す要素でtable要素の最初の子要素として記述する
- **caption要素**は、table要素以外の要素を入れ子にすることができる
- table要素の子要素として、caption要素の後、**thead要素**、**tbody要素**、**tfoot要素**、順に記述する必要がある
- **thead要素**と**tfoot要素**は、任意でtable要素の子要素として1つ設定可能
- **tbody要素**は、table要素の子要素にtr要素を指定している場合は、記述できない
- **thead要素**は、表組の**ヘッダー部分**の行グループを表す
- **tbody要素**は、表組の**本体部分**の行グループを表す
- **tfoot要素**は、表組の**フッター部分**の行グループを表す

51

第2章 文章構造

2.1 文章構成

2.1.1.2 表組（テーブル）



- **tr要素**は、表組内の行を表す
- **tr要素**は、table要素の子要素としてcaption要素、thead要素の後に記述できる
- **tr要素**は、table要素の子要素としてtbody要素が1つも無い場合、記述できる
- **tr要素**は、thead要素、tbody要素、tfoot要素の子要素として、記述できる
- **th要素**は、表組の見出しセルを表す
- **td要素**は、表組内のセル（列）を表す
- **th要素**、**td要素**は、tr要素の子要素として記述する

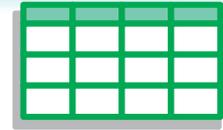
52

第2章 文章構造

2.1 文章構成

2.1.1.2 表組（テーブル）

- td要素、th要素にcolspan属性を記述できる
- td要素、th要素にrowspan属性を記述できる
- colspan属性は、行内のセルと隣のセル（横方向）を結合する属性値は、正の整数のみ指定できる 例1)
- rowspan属性は、行と行（縦方向）を結合する属性値は、「0（ゼロ）」または、正の整数を設定できる 例2)
- rowspan属性に「0」を設定すると指定したセルの最後の行まで結合する



例1) セルの結合（横方向）



例2) セルの結合（縦方向）



53

第2章 文章構造

```
01 <!DOCTYPE html>
02 <html lang="ja">
03 <head>
04 <meta charset="UTF-8">
   .
   .
   .
   .
147 </body>
148 </html>
```

• 実行結果

注文金額	本州・四国	北海道・九州	沖縄
3,000円以上	500円	700円	950円
5,000円以上	700円	850円	
10,000円以上	1,500円		
備考	購入金額が¥20,000円を超える場合は、送料が変わる場合がありますのでお問合せください。		

54

第2章 文章構造

2.1 文章構成

2.1.1.3 表題、コンテンツ、関連記事（headerタグ、mainタグ、footerタグ）

- **header要素**は、文書やセクションのヘッダーを表す
タイトルやロゴ、概要、ナビゲーションなどを記述する場所に使用される
- **main要素**は、ページ内のメインコンテンツであることを表す
1つの文章内でmain要素は、複数使用することはできない
- **footer要素**は、文書やセクションのフッターを表す
著者情報や関連情報へのリンクを記述する場所によく使用される
- **header要素、footer要素**は、子孫要素にheader要素、footer要素を持つことが出来ない



引用書籍：できるポケット Web制作必携 HTML & CSS全事典改定3版
著者：加藤善規&できるシリーズ編集部
出版社名：株式会社インプレス

55

第2章 文章構造

```
01 <!DOCTYPE html>
02 <html lang="ja">
03 <head>
04 <meta charset="UTF-8">
  :
  :
  :
150 </body>
151 </html>
```



例1) 入力する要素一覧

```
<header> ~ </header>
<main> ~ </main>
<footer> ~ </footer>
```

56

第2章 文章構造

2.1 文章構成

2.1.1.4 divタグ、spanタグ

- **div要素**は、意匠や他の設定を付加するためにHTML要素をグルーピングする要素
- **span要素**は文章中の一部分に意匠やその他の設定を付加するための要素
- **div要素**や**span要素**には、文章的な意味がない
- span要素は、Webブラウザで表示する際、画面では変化はない
- div要素は、終了タグの後ろに改行が入る
- div要素を記述する際、例1)の様な間違った記述をしない

例1) タグの入れ子の悪い例

```
<div> <P> はじめてのHTML </div> </p>
```



57

第2章 文章構造

```
01 <!DOCTYPE html>
02 <html lang="ja">
03 <head>
04 <meta charset="UTF-8">
   :
   :
   :
166 </body>
167 </html>
```

例1) タグの入れ子の悪い例

```
<div> <P> はじめてのHTML </div> </p>
```



58

第2章 文章構造

2.2 セクショニングコンテンツ

2.2.1 記事、補足情報、ナビゲーション（articleタグ、asideタグ、navタグ）

- セクショニングコンテンツ要素の中にh1～h6要素の見出しが必須（nav要素は任意）
フッタは任意
- **article要素**は、文章内の独立した記事のセクションを表す
- **article要素**を入れ子にすると、子孫要素となるarticle要素は、先祖要素のarticle要素の文章内容と関連がある内容を表す
- **nav要素**は、文章の主要なナビゲーションのセクションを表す
- **aside要素**は、補足や脚注、用語の説明など、本筋から分離しても問題ない内容（広告も含む）を含んだセクションを表す



引用書籍：できるポケット Web制作必携 HTML & CSS全事典改定3版
著者：加藤善規&できるシリーズ編集部
出版社名：株式会社インプレス

59

第2章 文章構造

```
01 <!DOCTYPE html>
02 <html lang="ja">
03 <head>
04 <meta charset="UTF-8">
   .
   .
   .
   .
172 </body>
173 </html>
```

• 実行結果



60

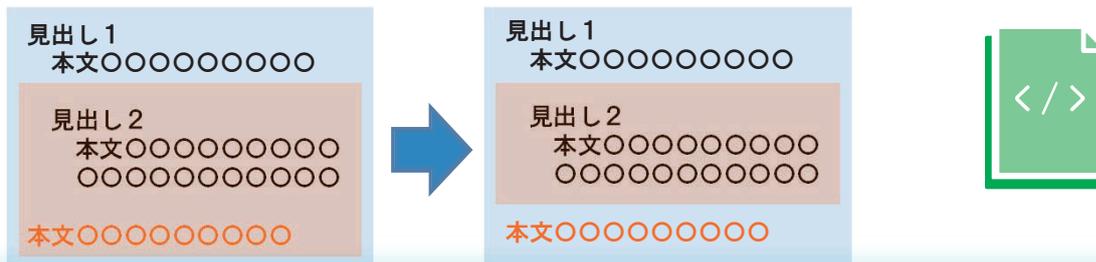
第2章 文章構造

2.2 セクショニングコンテンツ

2.2.2 文章の一区分 (sectionタグ)

- **section要素**は、1つの文書の固まり (セクション) を表す
- **section要素**は、h1~h6要素の任意の見出しを記述する必要がある
- 文書中の各セクションを記述していくとき、**article要素**、**aside要素**、**nav要素**の意味に一致しないセクションに**section要素**を使用する

例1) section要素の設定



61

第2章 文章構造

```
01 <!DOCTYPE html>
02 <html lang="ja">
03 <head>
04 <meta charset="UTF-8">
   .
   .
   .
   .
186 </body>
187 </html>
```

例1) 入力する要素一覧

```
<section> ~ </section>
```



62

第2章 文章構造

2.3 フォーム

2.3.1 フォーム (formタグ)

- フォームは、Webサイトの「お問合せ」や「検索ボックス」などのページのようにユーザがデータを入力して送信するページのこと
- **form要素**は、フォームに入力する入力欄のデータをどの形でどのように送るかを決めて送信する要素
- 1つのフォームに、1つのform要素を記述する
- form要素の送信のための属性、**action属性**と**method属性**を設定する 例1)
- **action属性**は、フォームに入力されたデータの送信先のURLを指定する
- **method属性**は、送信されるデータの送信方式を設定する
送信方式として、2種類、「**get**」と「**post**」がある

例1) form要素

```
<form action="send" method="post">  
<< 省略 >>  
</form>
```

例2) get送信例

```
https://www.google.co.jp/search?  
q=%E3%81%9F%E3%81%93%E
```

63

第2章 文章構造

2.3 フォーム

2.3.1 フォーム (formタグ)

- form要素は、**enctype属性**を設定し、送信するデータ形式を設定できる
- form要素は、**autocomplete属性**を設定し、入力候補の表示の有無を設定できる

例1) enctype属性値

エンコードタイプ	説明
application/x-www-form-urlencoded	データをURLエンコードする。(初期設定値)
multipart/form-data	ファイルデータをアップロードするための型 (マルチパート形式)
text/plain	プレーンなテキストデータ

例2) autocomplete属性値

設定値	説明
on	オートコンプリート機能をオンにする(記録させる)
off	オートコンプリート機能をオフにする(記録させない)

64

第2章 文章構造

```
01 <!DOCTYPE html>
02 <html lang="ja">
03 <head>
04 <meta charset="UTF-8">
   :
   :
   :
43 </body>
44 </html>
```



例1) 入力する要素一覧

```
<form action="contact.php" method="post" autocomplete="off">
</form>
```

65

第2章 文章構造

2.3 フォーム

2.3.2 フォームコントロール (inputタグ)

- **input要素**は、データを入力するフォームコントロール（入力欄）を表す
- input要素はtype属性、name属性、value属性を設定できる
- **type属性**を設定し、いろいろな形式のフォームコントロールに切り替えることができる
- **name属性**で送信するデータに対して名前を付ける
- **value属性**は、フォームコントロールの初期値を設定する
- value属性は、type属性の種類によって省略できる
- 送信する場合は、ユーザが入力または選択したinput要素の**name属性値**と**value属性値**を送信する
- form属性にform要素に設定しているid属性値を指定することでform要素の囲いの外に記述したフォームコントロールでもform要素と関連付けることができる



66

第2章 文章構造

2.3 フォーム

2.3.2 フォームコントロール (inputタグ)



(1) type属性、「text」の場合

- type属性に「text」を指定したinput要素は、1行のみのテキスト入力欄を表示する
- value属性は、初期値として文字列を設定する
- placeholder属性は、ダミーテキストを表示させる
入力欄に入力見本や、入力に対する注意書きなどのメッセージを入力する
- maxlength属性、minlength属性を設定することで「最大文字以内」「最小文字以上」の入力制限を設定できる

例1) input要素 テキスト

```
<input type="text" name="naamae" placeholder="名前を入力してください">
```

67

第2章 文章構造

```
01 <!DOCTYPE html>
02 <html lang="ja">
03 <head>
04 <meta charset="UTF-8">
   .
   .
   .
44 </body>
45 </html>
```

例1) 入力する要素一覧

```
<input type="text" name="name" placeholder="名前の入力">
```

• 実行結果



68

第2章 文章構造

2.3 フォーム

2.3.2 フォームコントロール (inputタグ)



(2) type属性、「email」の場合

- type属性に「email」を指定したinput要素は、1行のみのメール入力欄として表示する (例1)
- type属性がメールの場合、対応するWebブラウザの場合、メールアドレスの形式が適切ではない場合は、エラーが返される
- required属性を設定したフォームコントロールは、必須項目になり入力されていない場合データ送信がされない

使用できる他の属性…placeholder maxlength minlength form

例1) input要素 メール

```
<input type="email" name="email" placeholder="メールを入力してください" required>
```

69

第2章 文章構造



```
01 <!DOCTYPE html>
02 <html lang="ja">
03 <head>
04 <meta charset="UTF-8">
   .
   .
   .
   .
45 </body>
46 </html>
```

• 実行結果

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/HTML_site/contact/index.html'. The page content includes two input fields: '名前: [名前を入力してください]' and 'メールアドレス: [メールアドレスを入力して<'. The browser's title bar says 'Insert title here'.

70

第2章 文章構造

2.3 フォーム

2.3.2 フォームコントロール (inputタグ)



(3) type属性、「tel」の場合

- type属性に「tel」を指定したinput要素は、1行のみの電話番号入力欄として表示する
- 使用できる他の属性…placeholder maxlength minlength required

(4) type属性、「password」の場合

- type属性に「password」を指定したinput要素は、1行のみのパスワードの入力欄として表示する 例2)
- 使用できる他の属性…placeholder maxlength minlength required

例1) input要素 電話

```
<input type="tel" name="phone" placeholder="電話番号を入力してください">
```

例2) input要素 パスワード

```
<input type="password" name="password" placeholder="パスワード入力">
```

71

第2章 文章構造

```
01 <!DOCTYPE html>
02 <html lang="ja">
03 <head>
04 <meta charset="UTF-8">
   .
   .
   .
   .
47 </body>
48 </html>
```



• 実行結果

localhost:8080/HTML_site/contact/index.html

名前:

メールアドレス:

電話番号:

パスワード:

72

第2章 文章構造

2.3 フォーム

2.3.2 フォームコントロール (inputタグ)

(5) type属性、「checkbox」の場合

- type属性に「checkbox」を指定したinput要素は、複数選択が可能なチェックボックスを表示する
- checked属性を設定すると最初から選択されている状態にする

使用できる他の属性… required



例1) input要素 チェックボックス

```
<input type="checkbox" name="name" value="A" checked>Aランチ
```

73

第2章 文章構造

```
01 <!DOCTYPE html>
02 <html lang="ja">
03 <head>
04 <meta charset="UTF-8">
   .
   .
   .
   .
48 </body>
49 </html>
```



• 実行結果

名前:

メールアドレス:

電話番号:

パスワード:

好きな花の名前: あじさい ひまわり きんもくせい コスモス

74

第2章 文章構造

2.3 フォーム

2.3.2 フォームコントロール (inputタグ)



(6) type属性、「radio」の場合

- type属性に「radio」を指定したinput要素は、1つだけ選択可能なラジオボタンを表示する
- type属性を「radio」にしたinput要素のname属性の値は、すべて同じ値を記述する、同じではない場合、ラジオボタンとして機能しなくなる

使用できる他の属性… required checked

例1) input要素 ラジオボタン

```
<input type="radio" name="sei" value="man">男性  
<input type="radio" name="sei" value="woman">女性
```

75

第2章 文章構造

```
01 <!DOCTYPE html>  
02 <html lang="ja">  
03 <head>  
04 <meta charset="UTF-8">  
  .  
  .  
  .  
  .  
  .  
49 </body>  
50 </html>
```



• 実行結果

名前:

メールアドレス:

電話番号:

パスワード:

好きな花の名前: あじさい ひまわり きんもくせい コスモス

性別: 男性 女性

76

第2章 文章構造

2.3 フォーム

2.3.2 フォームコントロール (inputタグ)

(7) type属性、「date」の場合

- type属性に「date」を指定すると日付（年月日）の入力欄を表示する
- type属性が「date」に対応しているWebブラウザでは、カレンダーのインターフェースが表示される
- type属性が「date」の場合、送信される値は、yyyy-mm-dd(2023-10-01)の形式で送信される

使用できる他の属性… required

例1) input要素 サブミット

```
<input type="date" name="date">
```

77

第2章 文章構造

```
01 <!DOCTYPE html>
02 <html lang="ja">
03 <head>
04 <meta charset="UTF-8">
   .
   .
   .
50 </body>
51 </html>
```

• 実行結果

The screenshot shows a web browser window with the URL localhost:8080/HTML_site/contact/index.html. The form contains the following fields and options:

- 名前: 名前を入力してください
- メールアドレス: メールアドレスを入力してください
- 電話番号: 電話番号を入力してください
- パスワード: パスワードを入力してください
- 好きな花の名前: あじさい ひまわり きんもくせい コスモス
- 性別: 男性 女性
- 日付: 年 / 月 / 日

78

第2章 文章構造

2.3 フォーム

2.3.2 フォームコントロール (inputタグ)



(8) type属性、「submit」の場合

- type属性に「submit」を指定するとフォームに入力された情報を送信するボタンを表示する
- value属性値は表示されるボタンのラベルに使用される
- type属性が「submit」の場合、name属性値は、部品の名前を表す値になる

例1) input要素 サブミット

```
<input type="submit" name="submit" value="送信">
```

79

第2章 文章構造

```
01 <!DOCTYPE html>
02 <html lang="ja">
03 <head>
04 <meta charset="UTF-8">
   .
   .
   .
51 </body>
52 </html>
```



• 実行結果

Insert title here

localhost:8080/HTML_site/contact/index.html

名前:

メールアドレス:

電話番号:

パスワード:

好きな花の名前: あじさい ひまわり きんもくせい コスモス

性別: 男性 女性

日付: / /

80

第2章 文章構造

2.3 フォーム

2.3.2 フォームコントロール (inputタグ)



(9) type属性、「image」の場合

- type属性に「image」を指定したinput要素は、画像形式の送信ボタンを表示する
- type属性が「image」の場合は、input要素にsrc属性とalt属性が必須項目になり、画像の大きさは、width属性とheight属性でサイズ指定を行う

例1) input要素

```
<input type="image" src="bottn.gif" alt="送信ボタンの画像" width="100" height="60">
```

81

第2章 文章構造

```
01 <!DOCTYPE html>
02 <html lang="ja">
03 <head>
04 <meta charset="UTF-8">
   .
   .
   .
   .
52 </body>
53 </html>
```

• 実行結果

The screenshot shows a web browser window with the URL localhost:8080/HTML_site/contact/index.html. The form contains the following elements:

- 名前: [名前を入力してください]
- メールアドレス: [メールアドレスを入力して<
- 電話番号: [電話番号を入力してください]
- パスワード: [パスワードを入力してくださ
- 好きな花の名前: あじさい ひまわり きんちくせい コスモス
- 性別: 男性 女性
- 日付: [年 / 月 / 日]
- 送信

Below the form is a large blue button with the text 送信.

82

第2章 文章構造

2.3 フォーム

2.3.2 フォームコントロール (inputタグ)

(10) type属性、「reset」の場合

- type属性に「reset」を指定したinput要素は、フォームに入力したデータのリセットボタンを表示する



例1) input要素 リセット

```
<input type="reset" name="reset" value="リセット">
```

83

第2章 文章構造

```
01 <!DOCTYPE html>
02 <html lang="ja">
03 <head>
04 <meta charset="UTF-8">
   .
   .
   .
   .
52 </body>
53 </html>
```

• 実行結果

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/HTML_site/contact/index.html'. The form contains the following elements:

- 名前: [名前を入力してください]
- メールアドレス: [メールアドレスを入力して<
- 電話番号: [電話番号を入力してください]
- パスワード: [パスワードを入力してくださ
- 好きな花の名前: あじさい ひまわり きんもくせい コスモス
- 性別: 男性 女性
- 日付: [年 / 月 / 日]
- 送信 (Send) button
- リセット (Reset) button

84

第2章 文章構造

2.3 フォーム

2.3.3 ラベル (labelタグ)



- **label要素**は、フォームコントロールのラベルを表す
フォームコントロールとラベルを関連付けることができる
- フォームコントロールをラベルから選択することで、
フォームコントロールの編集ができる
- 2種類の設定方法
 - label要素でフォームコントロールとラベルを一緒に囲う方法
 - label要素のfor属性とフォームコントロールのid属性を
同じ設定値で関連付ける方法

例1) label要素

```
<input type="radio" name="sei" value="man" id="si_d"><label for="si_d">男性</label>
```

```
<label><input type="radio" name="sei" value="woman">女性</label>
```

85

第2章 文章構造

```
01 <!DOCTYPE html>
02 <html lang="ja">
03 <head>
04 <meta charset="UTF-8">
   .
   .
   .
   .
52 </body>
53 </html>
```

• 実行結果

名前:

メールアドレス:

電話番号:

パスワード:

好きな花の名前: あじさい ひまわり きんもくせい コスモス

性別: 男性 女性

日付:

送信

86

第2章 文章構造

2.3 フォーム

2.3.4 プルダウンメニュー（selectタグ、optionタグ）

- select要素は、選択式のメニュー（プルダウンメニュー）を表示する
- select要素は、プルダウンメニューの仕様を設定できる要素
- multiple 属性を指定すると複数選択ができる。
- プルダウンメニューの中に表示する項目をoption要素で記述する
- 使用できる他の属性… required checked form



例1) select要素 プルダウンメニュー

```
<label for="syohin">商品名 : </label>
<select id="syohin" name="syohinn" >
<option value="book">クッキングブック</option>
<option value="fork">フォーク</option>
<option value="spoon">スプーン</option>
</select>
```

87

第2章 文章構造

```
01 <!DOCTYPE html>
02 <html lang="ja">
03 <head>
04 <meta charset="UTF-8">
   .
   .
   .
   .
61 </body>
62 </html>
```

• 実行結果

名前:

メールアドレス:

電話番号:

パスワード:

好きな花の名前: あじさい ひまわり きんもくせい コスモス

性別: 男性 女性

日付:

好きな色:

送

リセット

88

第2章 文章構造

2.3 フォーム

2.3.5 テキストエリア (textareaタグ)

- **textarea要素**は、複数行のテキストを入力できる入力欄を表示する
- textarea要素は各Webブラウザでサイズの認識が違うのでrow属性、col属性を設定する
- **row属性**でテキストエリアの行数を設定する、値は整数で設定する
- **col属性**でテキストエリアの1行に含まれる文字数(幅)を設定する
値は整数で設定する
- 使用できる他の属性…placeholder maxlength minlength required form



例1) textarea要素

```
<label for="komento">コメント : </label>  
<textarea id="komento" name="komento" row="5" col="20">
```

89

第2章 文章構造

```
01 <!DOCTYPE html>  
02 <html lang="ja">  
03 <head>  
04 <meta charset="UTF-8">  
  .  
  .  
  .  
  .  
64 </body>  
65 </html>
```

• 実行結果

Insert title here

localhost:8080/HTML_site/contact/index.html

名前:

メールアドレス:

電話番号:

パスワード:

好きな花の名前: あじさい ひまわり きんもくせい コスモス

性別: 男性 女性

日付:

好きな色:

お問い合わせ内容

送信

90

第2章 文章構造

2.3 フォーム

2.3.6 ボタン (buttonタグ)



- **button要素**で囲ったデータをボタンとして表示する
- 空要素のinput要素と違い、button要素は、開始タグと終了タグがあるので、いろいろなデータを囲ってボタンとして使用できる
- **formaction属性**、**formenctype属性**、**formmethod属性**を設定すると関連つけられているform要素のそれぞれの属性値を書き換えることができる
- **disabled属性**を設定することでボタンを無効にすることができる
- type属性で「submit」と「reset」と「button」の3つの設定に切り替えることができる
- 使用できる他の属性…placeholder maxlength minlength required

例1) button要素

```
<button type="submit" name="botan">こちらからお入りください</button>
```

91

第2章 文章構造

```
01 <!DOCTYPE html>
02 <html lang="ja">
03 <head>
04 <meta charset="UTF-8">
   .
   .
   .
   .
64 </body>
65 </html>
```

• 実行結果

名前: [名前を入力してください]

メールアドレス: [メールアドレスを入力してください]

電話番号: [電話番号を入力してください]

パスワード: [パスワードを入力してください]

好きな花の名前: あじさい ひまわり きんもくせい コスモス

性別: 男性 女性

日付: [年/月/日]

好きな色: [blue]

お問合せ内容

送信

入力した内容をリセットします。

92

第2章 文章構造

2.4 ゲージ

2.4.1 進捗状況 (progressタグ)

- **progress属性**は、作業の進捗状況を視覚的にゲージで表示する終了地点での値に対する現在の進捗値を表すために使用する
- **max属性**をprogress要素に設定し、プログレスバーの最大値を設定するmax属性を省略する場合は、初期設定値は1である
- **value属性**は、完了したタスク量を表します



例1) progress要素

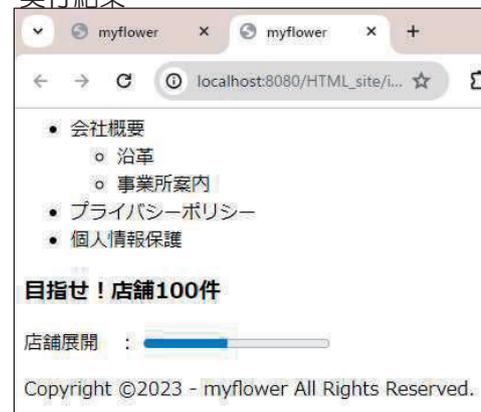
```
<progress value="70" max="100">70 %</progress>
```

93

第2章 文章構造

```
01 <!DOCTYPE html>
02 <html lang="ja">
03 <head>
04 <meta charset="UTF-8">
   .
   .
   .
   .
189 </body>
190 </html>
```

• 実行結果



94

第2章 文章構造

2.4 ゲージ

2.4.2 メーター（meterタグ）

- **meter**要素は、最初から最小値や最大値が決定している物の分量(割合)を表す
- **min**属性は範囲の最小値を設定し、**max**属性は範囲の最大値を設定する
- **low**属性は、ゲージの「低」として考えられる範囲を指定する
- **optimum**属性は、ゲージの「中」として最適値の範囲を指定する
- **high**属性は、ゲージの「高」として考えられる範囲を指定する
- **value**属性は、現在の値として指定する
- 指定する際は、
 - 「min <= value <= max」
 - 「min <= low <= max」
 - 「min <= optimum <= max」が成り立つように設定する

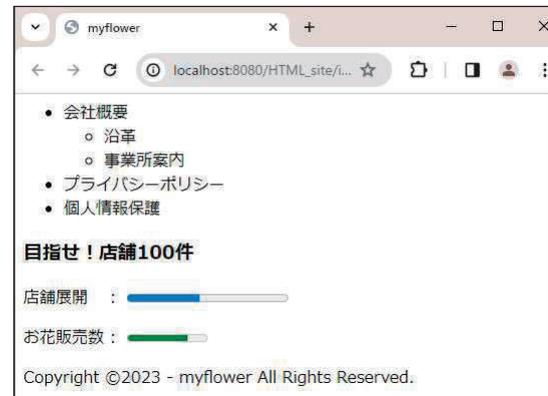


95

第2章 文章構造

```
01 <!DOCTYPE html>
02 <html lang="ja">
03 <head>
04 <meta charset="UTF-8">
   .
   .
   .
   .
190 </body>
191 </html>
```

• 実行結果



96

第2章 文章構造

2.5 外部参照

2.5.1 外部ファイルを参照する (linkタグ、scriptタグ、styleタグ)

- **link要素**は、HTMLファイルと外部ファイルの関係を設定する要素
- CSS3の外部ファイルをリンクに最もよく使用している
ファビコン (favicon) の設定にも使用する
※Favorite icon (お気に入りのアイコン) Webサイトのタイトル左上の
小さなアイコンのことを指す
- **rel属性**は、何のファイルが関係するか、ファイルの仕様を記述する 例1)
- **media属性**を設定するとリンクしている外部ファイルがどのメディアに
該当するかを指定する

例1) link要素

```
<link rel="stylesheet" href="css/css.css" type="text/css">
```

例1) media属性値

メディア	説明
all	すべてのデバイス
print	プリンターや印刷プレビュー
screen	printに一致しないすべてのデバイス

97

第2章 文章構造

2.5 外部参照

2.5.1 外部ファイルを参照する (linkタグ、scriptタグ、styleタグ)

- **style要素**は、HTMLファイルのhead要素の中にCSS(スタイルシート)を
記述する範囲を示す 例1)
- **script要素**は、javascriptなどのプリプロセッサ言語の埋め込み(例2)や
外部ファイルを参照する時に使用する 例3)

例1) style要素 css記述

```
<style>
.container {
    color:#fff;
    font-size:13px;
}
</style>
```

例2) script要素 埋め込み

```
<script>
    let answer=0;
    answer=1+2;
    console.log(answer);
</script>
```

例3) script要素 外部ファイル参照

```
<script src="js/script.js"></script>
```

98

HTML/CSS

第3章

CSS3

本章学習内容

本章では、CSS3の概要と記述ルール、CSS3の設定方法について学習します。

99

第3章 CSS3

3.1 CSS3

3.1.1 CSS3の記述場所

- **CSS3 (カスケードスタイルシート)**はHTMLプログラムで記述された文書に色やフォントサイズ、背景の色やレイアウトなどの**スタイル**を指定するための言語
- カスケードスタイルシートの「**カスケード**」は、段階的に連続しているものという意味があり、CSSでは、先に記述されたスタイルは次に引き継がれながらも、後に記述されたスタイルに上書きされる仕組み
- CSSのプログラムを記述場所は3つ
 - ①HTMLプログラムの各要素に直接記述する方法
 - ②HTMLのhead要素の中にstyle要素を使用して記述する方法
 - ③CSSファイル(外部ファイル)に記述する方法
- HTMLファイルのhead要素の中に**link要素**を使ってCSSファイル(外部ファイル)をリンクする

例1) CSS ソース

```
p{
  font-size : 15px;
  color : #003355;
  text-align : center;
}
```

100

第3章 CSS3

3.1 CSS3

3.1.1 CSS3の記述場所

(1) HTMLプログラムの各要素に直接記述する方法

例1) CSS ソース

```
:
<section>
<h4>地球温暖化</h4>
<p style="background:#0000ff;color:#ffffff;">地球温暖化の要因のひとつ、
CO<sub>2</sub><br>大気中のCO<sub>2</sub>の温室効果ガスなどが増
えると温室効果が強まり、地球の表面の気温が高くなります。</p>
</figure>
:
```

101

第3章 CSS3

```
:
48 <h3 style="color: #008000;">見出し3</h3>
49 <p>h1要素からh6要素の数字を意識しましょう。</p>
:
```

HTMLファイル (style属性)

```
index.html x
47 <hr>
48 <h3 style="color: #008000;">見出し3</h3>
49 <p>h1要素からh6要素の数字を意識しましょう。</p>
50 <hr>
51 <!-- メインコンテンツ -->
52 </body>
53 </html>
54
```

• 実行結果



102

第3章 CSS3

3.1 CSS3

3.1.1 CSS3の記述場所

(2) head要素の間にstyle要素を使用して記述する方法

```
:  
<head>  
<meta charset="UTF-8">  
<title>Insert title here</title>  
<style>  
p{  
  font-size : 15px;  
  color : #003355;  
  text-align : center;  
}  
</style>  
</head>  
<body>  
:
```

style要素の開始タグと終了タグの間にCSSを記述します。

103

第3章 CSS3

```
:  
06 <style>  
07 p{  
08   color : #ffa500;   
09 }  
10 </style>  
:
```

HTMLファイル (styleタグ)



• 実行結果



104

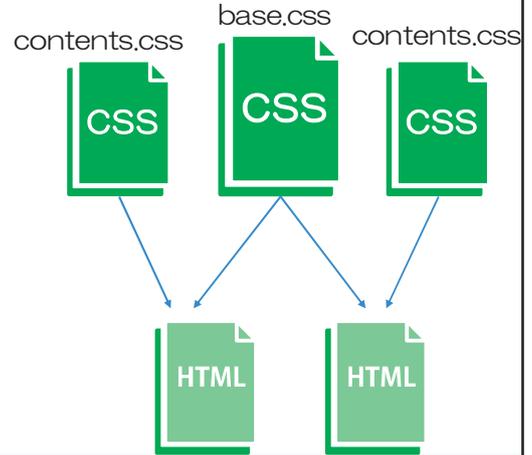
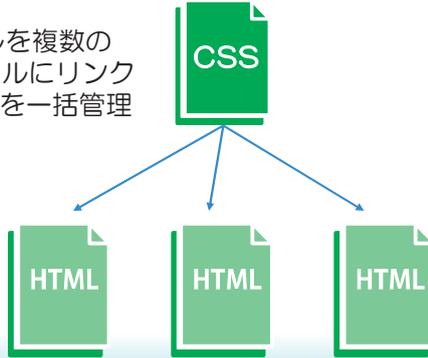
第3章 CSS3

3.1 CSS3

3.1.1 CSS3の記述場所

- (3) 外部ファイルをリンクしてCSS3を記述する方法
- よく使用されている記述方法

CSSファイルを複数のHTMLファイルにリンクさせてページを一括管理する



第3章 CSS3

3.1 CSS3

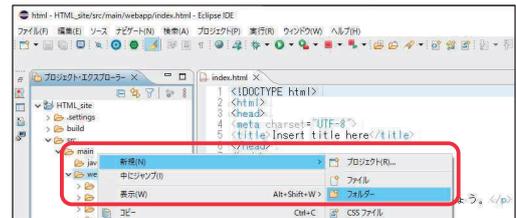
3.1.1 CSS3の記述場所

<外部ファイルの作成準備>

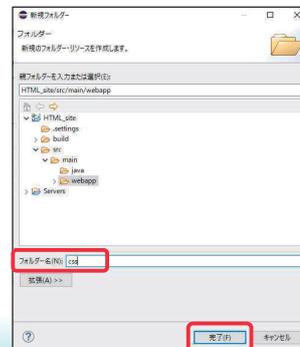
HTMLファイルにCSSの外部ファイルをリンクする

- ① CSSファイルを保存するフォルダを作成する
「webapp」フォルダの中に新規フォルダを「css」を作成する
- ② 「HTML_site」プロジェクト内の「src」フォルダの「main」フォルダの中の「webapp」フォルダにカーソルを合わせ右クリックし、「新規」の「フォルダ」をクリックする
- ③ 新規フォルダのウィンドウが表示するフォルダ名欄に「css」と入力する
「完了」をクリックする

例) ②



例) ③



第3章 CSS3

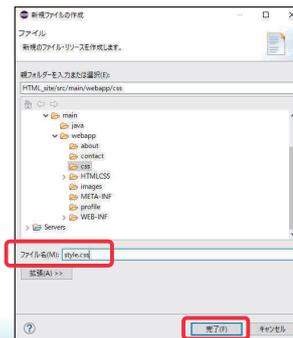
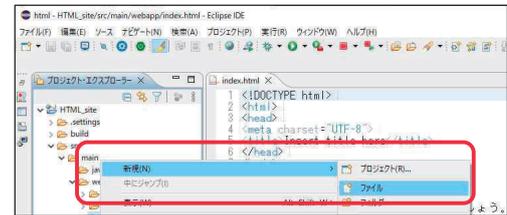
3.1 CSS3

3.1.1 CSS3の記述場所

<外部ファイルの作成>

- ① CSSファイルを作成する
先ほど作成した「css」フォルダの中に「style.css」という名前で新規ファイルを作成する
- ② 「webapp」フォルダの中の「css」フォルダにカーソルを合わせ右クリックする
ショートカットメニューから「新規」の「ファイル」をクリックする
- ③ 新規ファイルのウィンドウが表示される
ファイル名欄に「style.css」と入力する
「完了」をクリックする

例) ②



例) ③

107

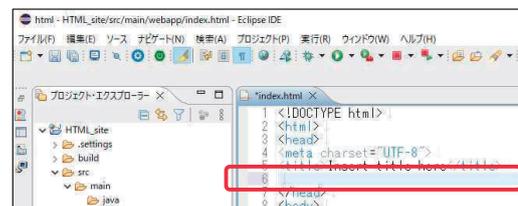
第3章 CSS3

3.1 CSS3

3.1.1 CSS3の記述場所

<外部ファイルのリンク>

- ① 「webapp」フォルダの中の「index.html」ファイルを開く
- ② HTMLファイルのhead要素の中にlink要素でCSSファイルをリンクする
- ③ HTMLファイルの5行目のtitle要素の後ろに改行を入れて6行目に例1)のlink要素を記述する
- ④ HTMLファイルを保存する



例1) link要素

```
<link rel="stylesheet" href="css/style.css" type="text/css">
```

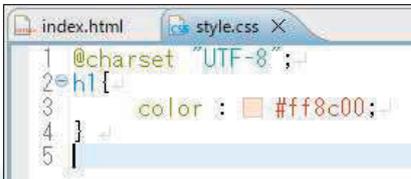
108

第3章 CSS3

-- CSSファイル --

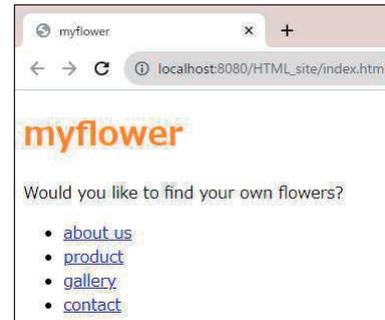
```
:  
02 h1{  
03     color : #ff8c00;   
04 }  
:
```

CSSファイル (style.css)



```
1 @charset "UTF-8";  
2 h1{  
3     color : #ff8c00;  
4 }  
5
```

• 実行結果



109

第3章 CSS3

3.1 CSS3

3.1.2 CSS3の記述ルール<CSS3の基本構文>

- CSS3の基本構文は、「セレクトア」「プロパティ」「値」の3つのパーツで構成されている (例1)
- セレクトアは、CSSを適用する対象を指定する
- プロパティは、どのようなCSSを適用するかを記述する
- 値は、対象に設定するスタイルの具体的な値
- 波括弧の間に適用したいスタイルの項目を記述する
- プロパティ名と値の間を「: (コロン)」で区切る
- プロパティは、複数設定できる
- プロパティと他のプロパティの間には、セミコロンで区切るようにする



例1) CSS3の基本構文



110

第3章 CSS3

3.1 CSS3

3.1.2 CSS3の記述ルール<CSS3の記述>

- 例1)の様に1行にまとめて記述できる
- CSSは、プログラムの記述する際は、好きなところに改行やタブ、半角スペースを入れられる
- 通常は、例2)の様に改行とタブなどを入れて各設定項目を見やすくする
- 全角スペースは、エラーになるので使用できない
- 「/*」と「*/」でテキストを囲むとプログラムとして認識されないコメントを記述できる、また複数行を囲うこともできる



例1) CSS3の記述例

```
p{color:#808080;font-size:14px;}
```

例2) CSS3の記述例

```
p {  
    color : #808080 ;  
    font-size : 14px ;  
}
```

111

第3章 CSS3

3.1 CSS3

3.1.3 CSS3 セレクタ

- **セレクタ**とは、スタイルを適用する際、対象となるHTML要素を指定するもの
- **セレクタ**の基本的な指定方法は、「要素」「class」「id」の3つある
- **要素セレクタ**は、例えばp要素をセレクタに指定した場合は、そのWebページ内に記述されているp要素すべてにスタイルが適用される



HTML

```
<p>〇〇〇</p>
```

p要素を使用したCSSの適用方法

CSS

```
p { color : #808080 ; }
```

セレクタにタグの名前をそのまま指定する

112

第3章 CSS3

-- CSSファイル --

```
01 :  
02 h2{  
03     color : #00bfff;   
04 }  
05 p{  
06     color : #696969;   
07 }
```

• 実行結果



113

第3章 CSS3

3.1 CSS3

3.1.3 CSS3 セレクタ<class>

- 「class(クラス)」セレクタは、任意の名前を付けられ、その名前で指定されたHTML要素だけにスタイルを適用する
- CSSにclassを記述する際、セレクタ名の先頭に「.(ドット)」付けるとclassとして指定ができる
- classは、1つのHTML要素に複数のclassを設定ができる、設定する際は、クラス名を半角スペースで区切る
- classは、CSSを設定する際、一番利用頻度が高いセレクタ



HTML

```
<p class="text">〇〇〇</p>
```

HTMLファイルの中での
classの適用の方法

```
<p class="text bold">〇〇</p>
```

classの複数適用の方法

CSS

```
.text{ color : #808080 ; }
```

ドットがclassを表す記号

114

第3章 CSS3

3.1 CSS3

3.1.3 CSS3 セレクタ<id>

- 「id(アイディ)」セレクタは、固有の要素を指定するのに使用する
固有名なのでページ内に同じ名前を複数使用できない
- CSSにidを記述する際、セレクタ名の先頭に「#(シャープ)」をつけると
idとして指定することができる



HTML

```
<p id="text">OOO</p>
```

HTMLファイルの中でのidの適用の方法

CSS

```
#text{ color : #808080 ; }
```

シャープがidを表す記号

115

第3章 CSS3

-- CSSファイル --

```
:
08 #contents{
09     background-color: #f5f5f5; 
10 }
11 .text{
12     color : #660099; 
13 }
:
```

-- HTMLファイル --

```
:
29 <div id="contents">
30     <h2> h2テキスト</h2>
31 :
34 <div>
35 <h3>h3-01テキスト</h3>
36 <p class="text">h3-01段落</p>
37 :
:
```

• 実行結果



116

第3章 CSS3

3.1 CSS3

3.1.3 CSS3 セレクタ<命名ルール>

- 「id」や「class」のセレクタ名は、半角英数字と「-(ハイフン)」「_(アンダースコア)」を使用して名前を記述する
- 「id」や「class」のセレクタ名は、大文字、小文字を区別して認識する
- 「id」や「class」のセレクタ名は、数字から始まる名前は使えない
- セレクタ名は、任意の名前をつけれるが、名前を見てスタイルの内容がイメージしやすい名前を考える

例1) セレクタ名に組み合わせでよく使用する単語例

単語	意味	単語	意味
container	全体を囲うもの	outer	外側
wrapper	包むもの	inner	内側
content	内容	item	内容

117



第3章 CSS3

3.1 CSS3

3.1.3 CSS3 セレクタ<再使用>

- セレクタを複数使用して指定することもできる
- 親要素のセレクタと子要素のセレクタとの間に半角スペースを入れて親要素と子要素の階層構造を指定する方法
- idやclassが適用されている要素を指定する方法

```
section div p { color : #fff ; }
```

例1) section要素の中のdiv要素の中のp要素に適用される

```
div#frame { color : #708090 ; }
```

例2) idが指定されているdiv要素に適用される

```
h3.text-item { color : #4a4a4a ; }
```

例3) classが指定されているh3要素に適用される

例) HTML要素

```
<div id="frame">  
<section>  
  <div>  
    <h3 class="text-item">h3テキスト</h3>  
    <p>h3段落</p>  
  </div>  
<p>段落</p>  
</section>  
</div>
```

118



第3章 CSS3

-- CSSファイル --

```
:
14 div#contents{
15     background-color: #ebf6f7;
16 }
17 p.text{
18     color: #b79b5b;
19 }
20 nav ul a{
21     color : #669900 ;
22 }
:
```



• 実行結果



119

第3章 CSS3

3.1 CSS3

3.1.3 CSS3 セレクタ<複数要素への適用>

- セレクタをカンマ区切りでまとめて記述できる 例1)
- 「>」で子要素を指定する方法 例2)
- 「+」で隣接する要素を指定する方法 例3)

```
div,h3,p{ color : #fff ;}
```

例1) カンマ区切りでまとめて適用

```
.frame> p{ color : #fff ;}
```

例2) オレンジ色のp要素のみに適用される

```
h3+p{ color : #708090 ;}
```

例3) h3要素に隣接するブルーのみ色に適用される

例)HTML要素

```
<div id="frame">
<h2>h2テキスト</h2>
  <p>段落</p>
<section>
  <div>
    <h3 class="text-item">h3テキスト</h3>
    <p>h3段落</p>
    <p>段落</p>
  </div>
</section>
</div>
```

120

第3章 CSS3

-- CSSファイル --

```

:
23 section>h3{
24     color: #f0908d; 
25 }
26 h2+p{
27     color: #e6b422; 
28 }
:
```

• 実行結果



121

第3章 CSS3

3.1 CSS3

3.1.3 CSS3 セレクタ<疑似要素>

- CSSのセレクタには、「疑似要素」「疑似クラス」の指定方法がある
- 疑似要素とは、要素の一部にスタイルを指定する
- 疑似要素の「::before」と「::after」を使用するときは、contentプロパティを使用する
- contentプロパティは、要素または、疑似要素内にコンテンツを追加する
- contentプロパティには、テキストや画像を指定できる



例1)よく使用する疑似要素

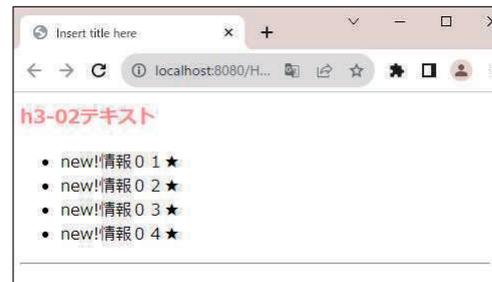
疑似要素	説明
::before	要素の内容の最初にコンテンツを追加
::after	要素の内容の最後にコンテンツを追加
::first-line	文章の1行目に対しスタイルを指定
::first-letter	文章の最初の1文字にスタイルを指定

122

第3章 CSS3

```
-- CSSファイル --
:
29 section h3+ul li::before{
30     content: "new!";
31 }
32 section h3+ul li::after{
33     content: "★";
34 }
:
```

• 実行結果



123

第3章 CSS3

3.1 CSS3

3.1.3 CSS3 セレクタ<疑似クラス>

- **疑似クラス**とは、要素の状態に対してスタイルを指定する
- よく使用する、疑似クラスは、a要素に関する設定
- 「ボタン」としての動作を設定する
- ボタンの動作設定の中で「:hover」は、必須



例1)よく使用する疑似クラス

疑似クラス	説明
:link	未訪問のリンクのスタイル
:visited	訪問済みのリンクのスタイル
:active	アクティブになった際のスタイル
:hover	マウスポインターを重ねられた要素のスタイル
:focus	フォーカスしている要素のスタイル

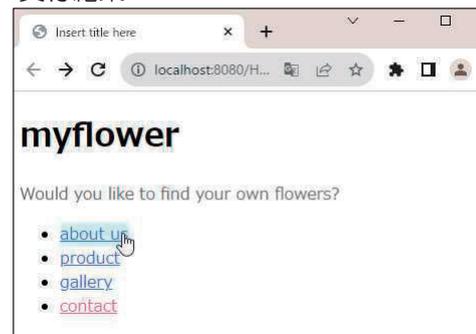
124

第3章 CSS3

-- CSSファイル --

```
:  
35 a:link{  
36     color:#4169e1;   
37 }  
38 a:visited{  
39     color: #db7093;   
40 }  
41 a:hover{  
42     background-color: #e0ffff;   
43 }  
:
```

• 実行結果



125

第3章 CSS3

3.1 CSS3

3.1.3 CSS3 セレクタ<子要素指定>

- 「:nth-child(n)」は、要素の子要素を最初から数えてn番目の要素にスタイルを指定する
- 「:nth-of-type(n)」は、要素の同一の子要素を最初から数えてn番目の要素にスタイルを指定する
- 「n」には、数値や数式、キーワード（odd(奇数)、even(偶数)）を指定する
- 「2n」と設定した場合は、nに自動的にゼロから順番に数値が入り計算する結果偶数に当たる要素だけにスタイルが適用する

例1)よく使用する疑似クラス

疑似クラス	説明
:first-child	最初の子要素にスタイルを適応
:last-child	最後の子要素にスタイルを適応
:nth-child(n)	子要素を数えたn番目にスタイルを適応
:nth-of-type(n)	同一の子要素を数えたn番目にスタイルを適応

126

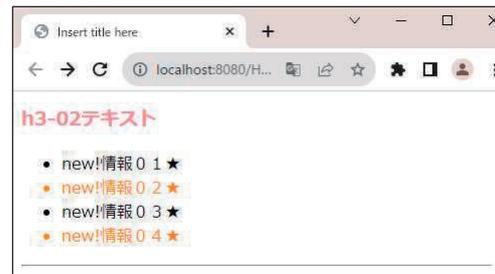
第3章 CSS3

-- CSSファイル --

```
:
44 section h3+ul li:nth-of-type(2n) {
45     color:#ff8c00;
46 }
:
```



• 実行結果



127

第3章 CSS3

3.1 CSS3

3.1.4 CSS3 継承

- 継承とは、親要素に適用したスタイルが引き継がれて子孫要素まで適用されること
- 継承されるプロパティ…color font-size font-family text-align など
- 継承されないプロパティ…border padding margin など
- 継承を制御する特別なプロパティ値があり、すべてのプロパティに設定可能



例1) 継承を制御するプロパティ値

プロパティ値	
inherit	継承する(継承しないプロパティも強制的に継承する)
initial	プロパティの初期値にリセットする
unset	自然に継承するものが無い場合はプロパティの初期値を指定
revert	ブラウザの持っているCSSの値にリセットする

128

第3章 CSS3

3.1 CSS3

3.1.5 CSS3 詳細度

- 詳細度とは、1つの要素に対して複数のセレクタでスタイルを指定した場合、どのセレクターが優先するかを決める数値
- 詳細度は、3種類の「ID」「CLASS」「TYPE」セレクタ階層に分かれており各セレクタの数がいくつかを計算し、大きい数値の方が優先する
- 数字が同一の数値の場合は、記述場所が後ろの方が優先する



例1) セレクタの詳細度例

セレクタ	ID	CLASS	TYPE	数値	
header.area h1	0	1	2	012	
#main header h1	1	0	2	102	優先
header h1	0	0	2	002	

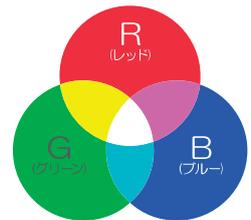
129

第3章 CSS3

3.1 CSS3

3.1.6 Webカラー<16進記法>

- Webでは、色を光の三原色 R (レッド)、G(グリーン)、B(ブルー)の三色を使用してフルカラーを表現する
- 色の指定は、「16進記法」を使用
「#」の後に6桁の数字とアルファベットで指定する
- 16進記法とは、0~9とA~Fまでの英数字を使用する16進数を使用、ゼロから16までを1桁で表記する方法
- それぞれの色が、0~255の数値を設定可能
16進数では、「255」→「FF」になる
- RGBの値がそれぞれ同じ値だった場合、省略して3桁で表示できる



例1) 16進数

0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F,10・・・

例2) 16進数カラーコード

R G B
(レッド) (グリーン) (ブルー)
00 00 00 ;
0~FF 0~FF 0~FF

例3) Webカラーの省略表記



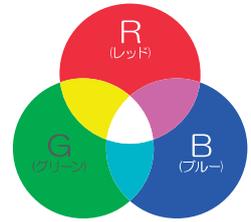
130

第3章 CSS3

3.1 CSS3

3.1.6 Webカラー<関数記法>

- 「関数記法」を使うと不透明度を設定できる
- 関数記法とは、「RGBA()」を使用して、RGBの各色を0~255の数値で色を設定する方法
- 「a」(アルファ)は、不透明度を0.0~1.0までの数値使用して、不透明度を設定する
- アルファの値に「0.5」と設定すると不透明度を50%に設定となる



例2) rgba()の色の指定

```
rgba( 000 , 000 , 000 , 00 );
```

R (レッド) G (グリーン) B (ブルー) a (アルファ)
0~255 0~255 0~255 0.0~1.0

HTML/CSS

第4章

CSS3の設定

本章学習内容

本章では、CSS3の各要素に対しての使用方法、デザインの作成について学習します。

132

第4章 CSS3の設定

4.1 テキストのデザイン

4.1.1 文字の色、文字の大きさ

- 「color」プロパティで文字の色を指定する
- 色は、16進記法または関数記法で指定する
- 「font-size」プロパティで文字の大きさを指定する
- 文字の大きさの単位は、「px」(ピクセル)や「%」(パーセント)がよく使用されている
- p要素に文字サイズの指定がない場合、Webブラウザが持つCSSの文字サイズ、「16px」が適用される

例1) 文字サイズの単位

単位	説明
px	ピクセル、画面を構成する色のついた点を1とした単位
%	親要素を幅や高さを基準する割合
em	フォントの専用、親要素フォントの大きさを基準した比率

133

第4章 CSS3の設定

-- CSSファイル --

```
:
02 /*-----タグの設定---*/
03 body{
04     color: #282828;
05 }
06 h1{
07     font-size: 27px;
08 }
09 p{
10     font-size: 13px;
11     color: #595757;
12 }
13 /*-----タグの設定---*/
:
```

・実行前



・実行結果



134

第4章 CSS3の設定

4.1 テキストのデザイン

4.1.2 文字の太さ、スタイル

- ・「**font-weight**」プロパティは、文字の太さを変更する
設定値は、**100, 200, 300, 400, 500, 600, 700, 800, 900**の数値と「normal」「bold」「bolder」「lighter」がある
- ・設定値の**100~900**の数値は、指定した数値と一致したフォント側のデータがあれば表示される、表示されない場合は、指定した数値に近い太さのフォントを表示する
- ・「**normal**」は、初期設定されている通常の太さで、設定値400と同じ太さ
- ・「**bold**」は、強調される太字の表示で、設定値700と同じ太さ
- ・「**lighter**」は、今のフォントの太さより1段階細く表示
- ・「**bolder**」は、今のフォントの太さより1段階太く表示



135

第4章 CSS3の設定

4.1 テキストのデザイン

4.1.2 文字の太さ、スタイル

- 「font-style」プロパティは、文字のスタイルを指定する
- 設定値は、「normal」「italic」「oblique」の3つ
- 「normal」は、標準のフォントで表示する
- 「italic」は、イタリック体のフォントで表示する
- 「oblique」は、斜体のフォントで表示する
- 「italic」「oblique」の両方のフォントがない場合は、文字が斜体のように傾いた状態で文字が表示される

例1) font-style イタリック体指定

```
p { font-style : italic ; }
```



136

第4章 CSS3の設定

-- HTMLファイル --

```
160 <h4>温暖化対策&lpar;SDGs&rpar;</h4>
161 <p>「<span class="bold">サンパチェンス</span>」は、1株で鉢植えの場合約60cm、
地球温暖化の原因とされる二酸化炭素&lpar;<span class="itari">CO<sub>2</sub></span>
&rpar;や、おもに自動車などの排気ガスにあるホルムアルデヒド&lpar;<span class="itari">
HCHO</span>&rpar;できわめて高い吸収能力を発揮することが実証されている。</p>
```

-- CSSファイル --

```
14 /* --サイドバーの設定--*/
15 .bold{
16     font-weight: bold;
17 }
18 .itai{
19     font-style: italic;
20 }
21 /* --サイドバーの設定--*/
```

• 実行前



• 実行結果



137

第4章 CSS3の設定

4.1 テキストのデザイン

4.1.3 文字位置、左揃え、中央揃え、右揃え、均等割り付け

- 「`text-align`」プロパティで文章の文字位置を設定する
- 設定値は、左揃えの「`left`」、中央揃えの「`center`」、右揃えの「`right`」、均等割り付けの「`justify`」がある
- 初期設定は「`left`」



例1) 左揃え

「color」プロパティで文字の色を指定する、「#」から始まる6桁の英数字、16進記法で指定する、「font-size」プロパティで文字の大きさを指定する

例2) 中央揃え

「color」プロパティで文字の色を指定する、「#」から始まる6桁の英数字、16進記法で指定する、「font-size」プロパティで文字の大きさを指定する

例3) 右揃え

「color」プロパティで文字の色を指定する、「#」から始まる6桁の英数字、16進記法で指定する、「font-size」プロパティで文字の大きさを指定する

例4) 均等割り付け

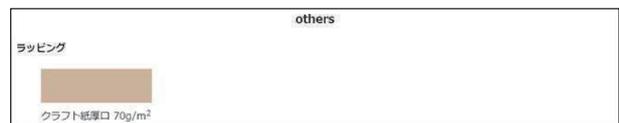
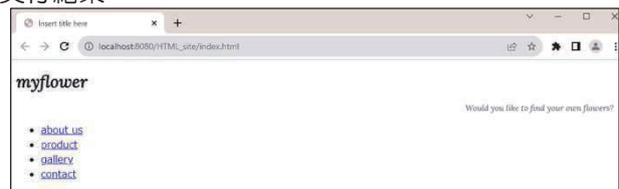
「color」プロパティで文字の色を指定する、「#」から始まる6桁の英数字、16進記法で指定する、「font-size」プロパティで文字の大きさを指定する

第4章 CSS3の設定

-- CSSファイル --

```
:
02 /*-----タグの設定----*/
03 body{
04     color: #282828;
05 }
:
29 footer>p {
30     text-align: center;
31 }
32 /* ---フッターの設定---*/
:
```

• 実行結果



第4章 CSS3の設定

4.1 テキストのデザイン

4.1.4 文字位置、上揃え、上下中央揃え、下揃え

- 「vertical-align」プロパティは、行内やセル内の垂直方向の位置を指定する
- 設定値は、「baseline」「top」「middle」「bottom」と数値での指定もできる
- 数値指定の場合は、「0」は、既存のベースラインの位置になり、そこから上方向が、正の数値、下方向は、負の数値を指定



140

第4章 CSS3の設定

-- CSSファイル --

```
:
36 .v-align01{
37     vertical-align: top;
38 }
39 .v-align02{
40     vertical-align: middle;
41 }
42 .v-align03{
43     vertical-align: bottom;
44 }
:
```

• 実行前



• 実行結果



141

第4章 CSS3の設定

4.1 テキストのデザイン

4.1.5 文字のフォント指定

- 「font-family」プロパティでフォントの指定ができる
- 指定されたフォントは、閲覧するパソコンに指定したフォントがない場合、Webブラウザの標準フォントで表示する
- 同じ系統のフォントを複数設定し、さらに「既定フォント」を設定する
- 「既定フォント」は、指定したフォントがない場合にWebブラウザのシステムフォントから既定フォントに設定したフォントを表示する

例1) font-family

```
p{ font-family: "MS PGothic", Osaka, Gothic, sans-serif; }
```

例2) 既定フォント

既定フォント	説明
serif	明朝系のフォントの指定
sans-serif	ゴシック系のフォント指定
monospace	等倍フォントの指定
system-ui	使用しているOSのUIと同じフォント

142

第4章 CSS3の設定

4.1 テキストのデザイン

4.1.6 Webフォント

- Webフォントは、閲覧環境に影響されず、指定したフォントで表示ができる
- Webフォントの使用方法は、「ダウンロードしてを使用する方法」と「フォントのサイトから直接読み込む方法」の2つ
- フォントの種類やウエイトの数が多いとフォントデータの読み込みに時間がかかる場合があるので注意する
- Googleフォントのサイトから読み込み場合は、アドレスを取得してCSSファイルの先頭に「@import」を使用して読み込む
- 読み込むためのアドレスは、Googleフォントのサイトから取得する



例1) Googleフォント読み込み

```
@import url('https://fonts.googleapis.com/css2?family=<省略>=Noto+Sans+JP&display=swap');
```

143

第4章 CSS3の設定

4.1 テキストのデザイン

4.1.6 Webフォント

- 「**フォントデータを使用する**」場合は、Googleフォントのサイトから使用したいフォントデータをダウンロードする
- 「**@font-face**」を使用してフォントデータを読み込む
- 「**font-family**」プロパティで指定するフォントファミリーの名前を任意で決める
- 「**src**」でフォントデータを指定するパスを指定する
- 「**format**」でフォントデータの形式を指定する

例1) フォントデータの読み込み

```
@font-face{
  font-family: "gosikku-medium";
  src: url("static/Gothic-Medium.ttf") format("truetype");
}
```

例2) フォントデータの形式

フォントの形式	説明
woff / woff2	WOFFフォント
truetype	TrueTypeフォント
opentype	OpenTypeフォント
svg	SVGフォント

144

第4章 CSS3の設定

```
-- CSSファイル --
:
02 @font-face{
03     font-family: "lora-mediumItalics";
04     src: url("../web-font/Lora-MediumItalic.ttf") format("truetype");
05 }
06 @font-face{
07     font-family: "Montserrat-regular";
08     src: url("../web-font/Montserrat-Regular.ttf") format("truetype");
09 }
10 /*-----タグの設定----*/
:
```

145

第4章 CSS3の設定

-- CSSファイル --

```
:  
14 h1{  
15     font-size: 27px;  
16     font-family: "lora-mediumItalics";  
17 }  
18 h1+p{  
19     text-align: right;  
20     font-family: "lora-mediumItalics";  
22 }  
23 h2{  
24     font-family: "Montserrat-regular";  
25 }  
:
```

• 実行前



• 実行結果



146

第4章 CSS3の設定

4.1 テキストのデザイン

4.1.7 テキストシャドウ

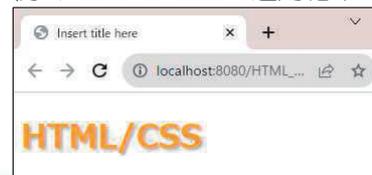
- 「`text-shadow`」プロパティで文字に影をつけることができる
- 設定値は、①「影の水平方向の位置」、
②「影の垂直方向の位置」、
③「ぼかし半径」、
④「色の設定」 の順番に設定する

例1) text-shadow

```
p { text-shadow : 2px 2px 3px #a0a0a0 ; }
```

① ② ③ ④

例2) text-shadow適用結果

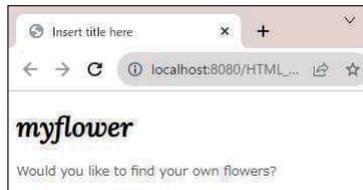


147

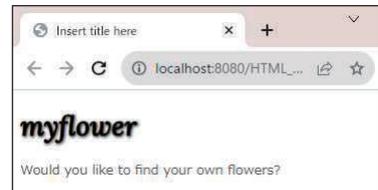
第4章 CSS3の設定

```
-- CSSファイル --  
:  
14 h1{  
15     font-size: 27px;  
16     font-family: "lora-mediumItalics";  
17     text-shadow: 1px 1px 3px #212121;  
18 }
```

• 実行前



• 実行結果



148

第4章 CSS3の設定

4.1 テキストのデザイン

4.1.8 背景の色、画像の設定

- 「background-color」を設定することで要素の背景に色が指定できる
- 色の指定は、16進記法、関数記法を使用

例1) 16進記法

```
p{ background-color : #66ccff ;
```

例2) 関数記法

```
p{ background-color : rgba( 102 , 204 , 255 , .5);
```

- 「background-image」を設定することで要素の背景の画像を指定できる
- 画像の設定は、プロパティ値をカンマで区切ることで複数指定できる
- 画像を複数指定した場合は、1つ目が最前面に表示され2枚目はその背面に表示される

例3) 背景画像の指定

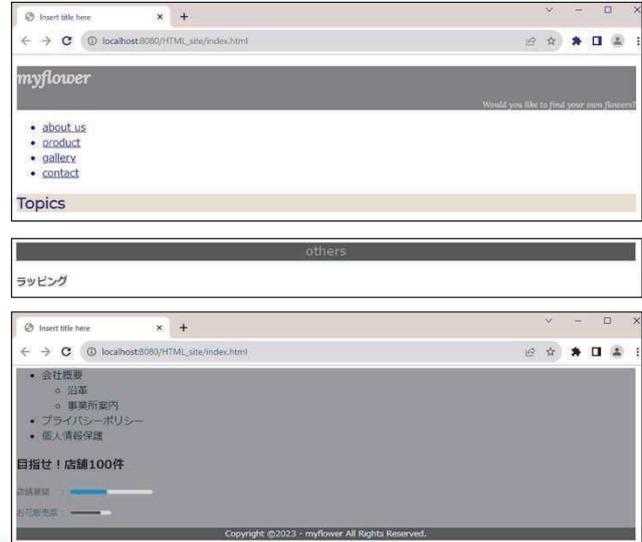
```
p{ background-image : url( images/bg-taiga.jpg); }
```

149

第4章 CSS3の設定

```
-- CSSファイル --
:
56 footer>p {
57     text-align: center;
58     color: #fff;
59     background-color: #595757;
60 }
61 /* ---フッターの設定---*/
:
```

・実行結果



150

第4章 CSS3の設定

4.1 テキストのデザイン

4.1.8 背景の色、画像の設定

- 背景に指定した画像は、**繰り返し(リピート)**で貼り付けられる
- 「**background-repeat**」を設定することで画像の繰り返しを制御ができる
カンマ区切りで複数の画像のリピートを指定できる
- 設定値を半角スペースで区切り、水平方向、垂直方向と個別に設定ができる

例1) スペース区切り

```
p { background-repeat : repeat no-repeat ;
```

例2) カンマ区切り(複数画像の指定)

```
p { background-repeat : repeat , no-repeat ;
```

- 「**repeat**」は、背景画像が繰り返し表示されるが、
指定したボックスからはみ出る部分は切り取られて表示する
- 「**space**」は、背景画像が繰り返し表示されるが、
指定した領域からはみ出ないように間隔を調整して表示する
- 「**round**」は、背景画像が繰り返し表示されるが、
指定した領域の中に納まるように自動で拡大縮小をして表示する



151

第4章 CSS3の設定

4.1 テキストのデザイン

4.1.8 背景の色、画像の設定

- 「repeat-x」を設定すると画像を水平方向に繰り返し表示する
「repeat no-repeat」と同値になる
- 「repeat-y」を設定すると画像を垂直方向に繰り返し表示する
「no-repeat repeat」と同値になる
- 「no-repeat」を設定すると背景画像を繰り返さない



152

第4章 CSS3の設定

-- CSSファイル --

```
:  
45 .haikei div {  
46     background-image: url(images/daiya.gif);  
47 }  
48 .haikei02 {  
49     background-repeat: space;  
50 }  
51 .haikei03 {  
52     background-repeat: round;  
53 }  
:
```

• 実行結果

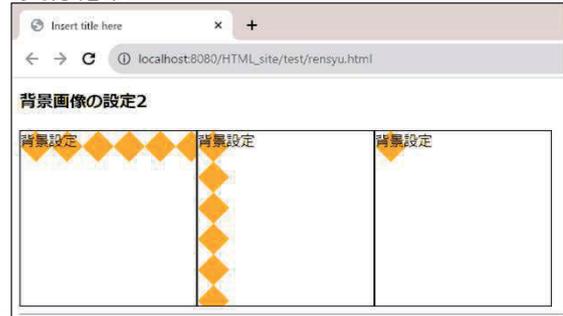


153

第4章 CSS3の設定

```
-- CSSファイル --
:
54 .haikei04 {
55     background-repeat: repeat-x;
56 }
57 .haikei05 {
58     background-repeat: repeat-y;
59 }
60 .haikei06 {
61     background-repeat: no-repeat;
62 }
:
```

• 実行結果



154

第4章 CSS3の設定

4.1 テキストのデザイン

4.1.8 背景の色、画像の設定

- 「`background-position`」を設定すると背景画像の表示位置を指定できる
- 設定値は、「`top`」、「`bottom`」、「`left`」、「`right`」、「`center`」と「`数値`」を設定でき、水平方向と垂直方向を対象とする
- 設定値は最大4つまで設定ができ、設定値を半角スペースで区切る
- 1つ目の設定値が、「`top`」、「`bottom`」、「`left`」、「`right`」のどれかの場合、2つ目の設定値に同じ値または、同じ方向を設定できない
- 1つの数値のみ設定した場合、水平方向左からの距離として設定され、垂直方向は、中央に配置される
- キーワード値が1つの場合、もう一方の設定が「`center`」になる
- 数値が2つの場合、前から水平方向、垂直方向として設定する



例1) `background`

```
p { background-position : top right ;
```

155

第4章 CSS3の設定

4.1 テキストのデザイン

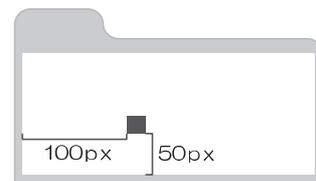
4.1.8 背景の色、画像の設定

- 「`background-position`」のプロパティは、設定値を最大4つまで設定できる
- 設定値が4つの場合は、1つ目と3つ目の値がキーワード値で、2つ目と4つ目の値が、オフセット値になる
- 設定値が3つの場合は、1つ目と2つ目の値のどちらかにキーワード値の設定が必要
- 2つ目にキーワード値を指定した際、その後の数値は、前の値のオフセット値になる



例1) 設定値が4つの場合

```
.imgs{ background-position : left 100px bottom 50px ;
```



例2) 設定値が4つの場合

156

第4章 CSS3の設定

-- CSSファイル --

```
:
45 .haikei div{
46     background-image: url(images/daiya.gif);
47     background-repeat: no-repeat;
48 }
49 .haikei02{
50     background-position: top;
51 }
52 .haikei03{
53     background-position: right;
54 }
:
```

• 実行結果



157

第4章 CSS3の設定

```
-- CSSファイル --
:
55 .haikei04{
56     background-position: left bottom;
57 }
58 .haikei05{
59     background-position: top 100px left;
60 }
61 .haikei06{
62     background-position: bottom 50px left 100px;
63 }
:
```

・実行結果



158

第4章 CSS3の設定

4.1 テキストのデザイン

4.1.8 背景の色、画像の設定

- 「background-size」を設定することで、背景画像のサイズを指定できる
- 設定値としては、「cover」、「contain」、「auto」、「数値」を設定することができる
- 「auto」を設定した場合、背景画像の表示サイズを自動調整して表示する
- 「cover」を設定すると背景画像の縦横の比率を保った状態で、指定された領域のすべてをカバーするように表示サイズを調整する
領域より背景画像が大きい場合は、背景画像が見切れる
- 「contain」を指定すると背景画像の縦横の比率を保持し、画像を切り取ることなく背景画像を指定領域の最大表示サイズで表示する
- 「数値」で指定する場合は、
2つの数値を半角スペースで区切り単位付き数値で設定する（幅、高さの順）
1つの数値で指定した場合は、2つ目の値は「auto」になる

例1) 数値での設定

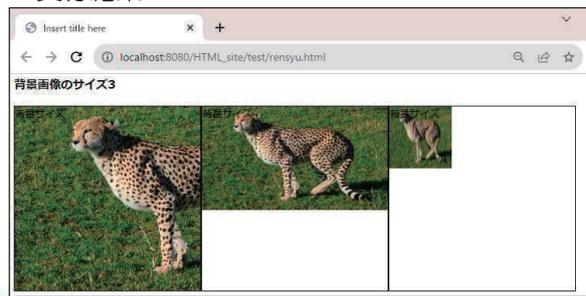
```
.imgs{ background-size : 100px 50px ;
```

159

第4章 CSS3の設定

```
-- CSSファイル --
:
64 .size-cover {
65 background-image:url(images/hyo.jpg);
66 background-size: cover;
67 }
68 .size-contain {
69 background-image: url(images/hyo.jpg);
70 background-size: contain;
71 }
72 .size-number {
73 background-image: url(images/hyo.jpg);
74 background-size: 100px 100px;
75 }
:
```

• 実行結果



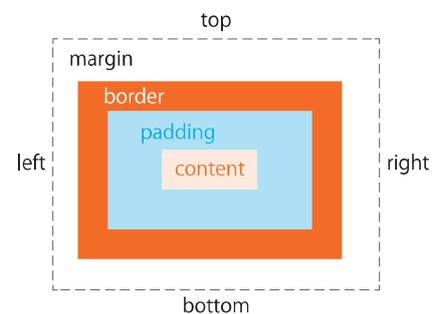
160

第4章 CSS3の設定

4.1 テキストのデザイン

4.1.9 ボックスモデル

- CSSには、各要素を取り囲む、四角形の領域「ボックス」という概念がありレイアウトや要素のサイズ指定などに使用する
- ボックスには、外側から順に「margin」(マージン)、「border」(ボーダー)、「padding」(パディング)、「content」(コンテンツ)で構成されている
- 「content」は、テキストデータの部分
- 「padding」は、テキストとボーダーとの間の余白背景の領域に含まれる
- 「border」は、テキストの四方に指定できる罫線
- 「margin」は、borderの外側に指定できる余白



例1) ボックスモデル

引用書籍：できるポケット Web制作必携 HTML & CSS全事典改定3版
著者：加藤善規&できるシリーズ編集部
出版社名：株式会社インプレス

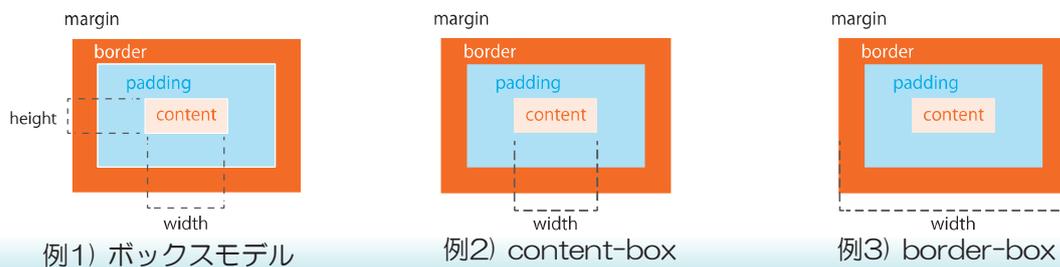
161

第4章 CSS3の設定

4.1 テキストのデザイン

4.1.1 0 幅 (width)、高さ (height) (ボックスモデル)

- 「ボックスモデル」の「width」(幅)、「height」(高さ)にborderとpaddingは含まれない
- 「`box-sizing`」プロパティの「`content-box`」を設定すると
「width(幅)」、「height(高さ)」にborderとpaddingを含まない計算
- 「`box-sizing`」プロパティの「`border-box`」を設定すると
「width(幅)」、「height(高さ)」にborderとpaddingを含めた計算



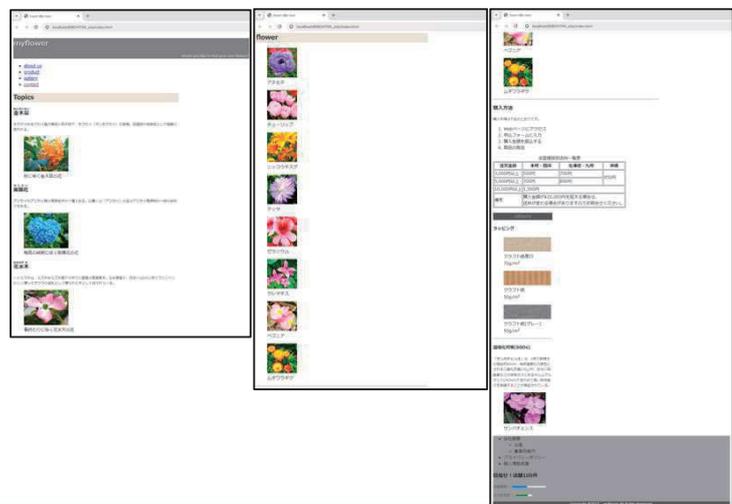
162

第4章 CSS3の設定

-- HTMLファイル --

```
:
08 <body>
09 <!-- ----メインコンテンツ---- -->
10 <!-- -----wrapper-start -->
11 <div id="wrapper">
12   <header>
13     <h1>myflower</h1>
14   :
208 </div>
209 <!-- -----wrapper-end -->
210 </body>
211 </html>
```

実行結果



163

第4章 CSS3の設定

4.1 テキストのデザイン

4.1.1.1 パディング（ボックスモデル）

- 「padding」プロパティは、ボックス内の余白を設定する
- パディングは、padding-top、padding-right、padding-bottom、padding-leftと各方向ごとに設定ができる 例1)
- 各方向をまとめて指定する方法「ショートハンド」がある

例1) パディングの設定

```
p {  
  padding-top : 20px ;  
  padding-left : 10px ;  
  padding-right : 10px ;  
  padding-bottom : 20px ;  
}
```

ショートハンド	説明
padding : 20px ;	四方向すべて同じ設定
padding : 20px 10px ;	前から上下、左右の設定
padding : 20px 10px 20px ;	前から上、左右、下の設定
padding : 20px 10px 20px 10px ;	前から上、右、下、左の順番に指定

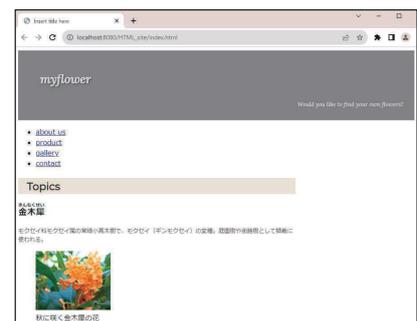
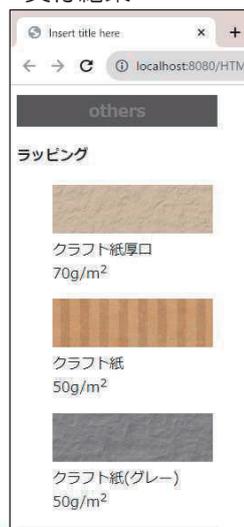
164

第4章 CSS3の設定

-- CSSファイル --

```
:  
01 @charset "UTF-8";  
02 @font-face {  
03     font-family: "lora-mediumItalics";  
:  
87 .footer-container {  
88     width: 500px;  
89 }  
90 /* ---フッターの設定---*/
```

• 実行結果



165

第4章 CSS3の設定

4.1 テキストのデザイン

4.1.1 2 ボーダ（ボックスモデル）

- 「border」プロパティは、ボックスのパディングの外側に線を指定する
- ボーダは、border-top、border-right、border-bottom、border-left の4方向に設定できる
- 設定として、「border-color」：線の色、「border-weight」：線の太さ、「border-style」：線種 がある
- 接種は、「solid」：実線、「double」：二重線、「dotted」：点線、「dashed」：破線 などがある
- 線種と線幅、線の色をまとめて記述するショートハンドが使える
- ショートハンドを使用するときは、必ず「線種」を設定しなければならない

例1) borderのショートハンド

```
p{border : solid 1px #d96913 ;}
```

```
p{
border-style : solid ;
border-weight : 5px ;
border-color : #000 ;
}
```

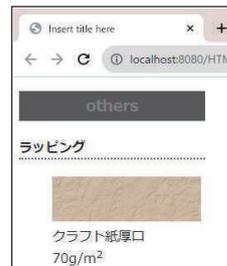
166

第4章 CSS3の設定

-- CSSファイル --

```
:
01 @charset "UTF-8";
02 @font-face{
03     font-family: "lora-mediumItalics";
:
94 .footer-container{
95     width: 500px;
96 }
97 /* ---フッターの設定---*/
```

• 実行結果



167

第4章 CSS3の設定

4.1 テキストのデザイン

4.1.1.3 マージン（ボックスモデル）

- 「margin」プロパティは、ボックスの外側の余白を設定する
- マージンは、margin-top、margin-right、margin-bottom、margin-leftと各方向ごとに設定ができる 例1)
- 各方向をまとめて指定する方法「ショートハンド」がある
- ボックスの左右マージンを「auto」に設定するとボックスがセンタに配置される、ただし、ボックスの「width」の設定が必須

例1) marginの設定

```
p {  
  margin-top : 15px ;  
  margin-left : 20px ;  
  margin-right : 20px ;  
  margin-bottom : 15px ;  
}
```

ショートハンド	説明
margin : 15px ;	四方向すべて同じ設定
margin : 15px 20px ;	前から上下、左右の設定
margin : 15px 20px 15px ;	前から上、左右、下の設定
margin : 15px 20px 15px 20px ;	前から上、右、下、左の順番に指定

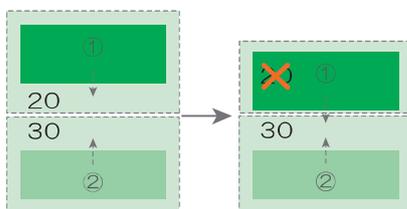
168

第4章 CSS3の設定

4.1 テキストのデザイン

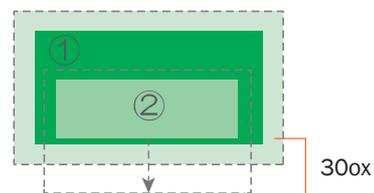
4.1.1.3 マージン（ボックスモデル）

- 「margin」プロパティは、隣り合う要素の上下のマージンは相殺され、設定されている大きい数値が適用される
- 要素の入れ子状態の場合、親要素と子要素の上下間の同じ方向にマージンが設定されると、相殺され大きい数値の方が適用される



マージン

- ①がマージン20pxと
- ②のマージンが、30pxが隣り合うとき相殺が起こり大きい値の30pxが適用される



親要素①がマージン20pxと
子要素②のマージンが、30pxが同じ上下
方向に合うとき相殺が起こり大きい値の
30pxが適用される

169

第4章 CSS3の設定

```
-- CSSファイル --
:
01 @charset "UTF-8";
02 @font-face{
03     font-family: "lora-mediumItalics";
:
106 .footer-container{
107     width: 500px;
108 }
109 /* ---フッターの設定---*/
```



• 実行結果



170

第4章 CSS3の設定

4.1 テキストのデザイン

4.1.1.4 ボックスモデル

- 「ボックス」には、「**ブロックボックス**」と「**インラインボックス**」の2種類がある
- **ブロックボックス**は、他のブロックボックス、インラインボックスを内包しウィンドウ幅、左右いっぱいには四角形の領域を形成する
- **ブロックボックス**は、垂直方向に配置され
インラインボックスは、水平方向に配置される
- ブロックボックスは、幅、高さやmargin、padding、borderを指定できる
インラインボックスは、一部指定できない部分がある
- 要素によっては、初期の設定で、ブロックボックス扱いをできるもの
インラインボックス扱いをできるもの
と決まっている場合があるが、「**display**」プロパティで変更することができる
- 「**display**」プロパティの「**none**」は、指定された要素を非表示にする

引用書籍：できるポケット Web制作必携 HTML & CSS全事典改定3版
著者：加藤善規&できるシリーズ編集部
出版社名：株式会社インプレス

171

第4章 CSS3の設定

```
-- CSSファイル --
:
87 header nav ul li {
88     width: 255px;
89 }
90 header nav ul li a {
91     display: block;
92 }
93 /* -----ナビゲーションの設定 */
:
```



172

第4章 CSS3の設定

4.1 テキストのデザイン

4.1.15 スクロールバー

- 「**overflow**」は、ボックスに収まらないテキストの表示方法を指定する
- 「**overflow : scroll;**」を設定するとボックスからはみ出たテキストにスクロールバーを使用して表示する
- 「**overflow : hidden;**」を設定するとボックスからはみ出たテキストを非表示にする
- 「**overflow : auto;**」を設定するとボックスからテキストがはみ出した場合は、自動ではみ出した方向にスクロールバーが表示される

- 「**overflow-x**」(水平方向)、「**overflow-y**」(垂直方向)と各方向で設定もできる
- 「**overflow**」の設定値を2つ設定することができ、最初の設定値は、「**overflow-x**」の値になり、2個目は、「**overflow-y**」の値になる



173

第4章 CSS3の設定

-- CSSファイル --

```
:
01 @charset "UTF-8";
02 @font-face{
03     font-family: "lora-mediumItalics";
:
104 .footer-container{
105     width: 500px;
106 }
107 /* ---フッターの設定---*/
```

• 実行結果



174

第4章 CSS3の設定

4.1 テキストのデザイン

4.1.16 ボックスシャドウ

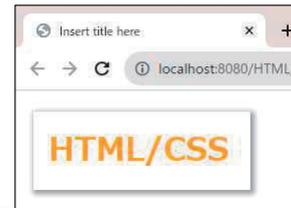
- 「`box-shadow`」を設定することでボックスに影をつける
- 設定値は、①「影の水平方向の位置」、②「影の垂直方向の位置」、③「ぼかし半径」、④「広がり」、⑤「色の設定」の順番に設定する
- 「`box-shadow : none ;`」を設定すると影がない状態を表します。
- 「`box-shadow : inset ;`」を設定するとボックス内に影をつける

例1) `box-shadow`

```
p{ box-shadow : 2px 2px 5px 2px #a0a0a0 ; }
```

① ② ③ ④ ⑤

例2) `box-shadow`適用結果



175

第4章 CSS3の設定

-- CSSファイル --

```
107 footer>p {  
108     text-align: center;  
109     color: #fff;  
110     background-color: #595757;  
111     padding: 8px 0 5px;  
112     margin-top: 20px;  
113     box-shadow: 0px -2px 2px #333;  
114 }
```

• 実行結果



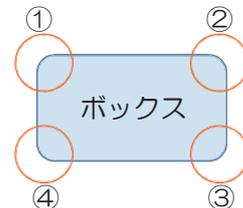
176

第4章 CSS3の設定

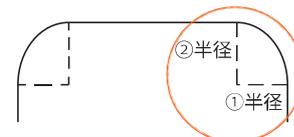
4.1 テキストのデザイン

4.1.17 角丸

- 「**border-radius**」を設定することでボックス角を丸めることができる
- 数値が1つの場合は、正円の半径として解釈され
- 数値が2つの場合は、1個目が水平方向の半径、2個目は垂直方向の半径の楕円
- 「**border-top-left-radius**」は、左上角①
- 「**border-top-right-radius**」は、右上角②
- 「**border-bottom-right-radius**」は、右下角③
- 「**border-bottom-left-radius**」は、左下角④



1つの設定
正円の半径と
して設定



2つの設定
水平の半径と
垂直の半径になる

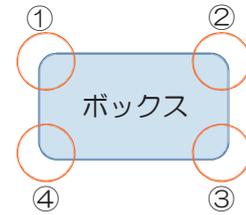
177

第4章 CSS3の設定

4.1 テキストのデザイン

4.1.1.7 角丸

- 「`border-radius`」のそれぞれの角の設定にショートハンドが使える
- 値が1つの場合：すべて同じ角丸になる
- 値が2つの場合：1つ目は、左上角①と右下角③、2つ目は、右上角②と左下角④に適用
- 値が3つの場合：1つ目は、左上角①、2つ目は右上角②と左下角④、3つ目は右下角③に適用
- 値が4つの場合：1つ目から順に、左上角①、右上角②、右下角③、左下角④に適用
- 水平方向と垂直方向に各値を設定したい場合は、「水平方向/垂直方向」と「/」で区切って記述



例1) `border-radius`

```
p { border-radius : 8px 4px 8px 4px; }
```

例2) `border-radius` 水平方向、垂直方向別指定

```
p { border-radius : 3px 3px 6px 6px /6px 6px 3px 3px ; }
```

178

第4章 CSS3の設定

-- CSSファイル --

```
:  
47 aside>h3 {  
48     text-align: center;  
49     color: #808080;  
50     background-color: #595757;  
51     padding: 5px 10px 3px;  
52     margin-top: 25px;  
53     border-radius: 8px 8px 0px 0px;  
54 }
```

• 実行結果



179

第4章 CSS3の設定

4.2 レイアウト

4.2.1 フロート

- 「float」プロパティは、要素を横に回り込ませて横並びに配置
- 設定値を「left」を設定すると要素を左揃えに配置
- 設定値を「right」を設定すると要素を右揃えに配置
- 設定値を「none」を設定すると回り込みをしない
- 「float」を指定した要素は、周りにある要素に機能を自動的に継承する
回り込みをさせたくない要素には、「clear」を設定して解除する
- 「clear」の設定値には、「right」「left」「both」「none」の4つがあり
floatで指定された方向と同じ値を設定することで解除できる
- 「both」は、両方の方向を解除する値でよく解除に使用する
「none」は回り込みを解除しない



180

第4章 CSS3の設定

4.2 レイアウト

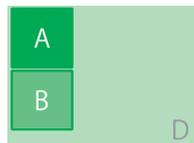
4.2.1 フロート

- 「float」の意味としては、「浮く」なので、要素を浮かして回り込みをする
「float」を指定していない要素は、浮いた要素の下に入り込み表視されない場合がある
- 解消方法としては、隠れている要素に同じ方向のfloatを指定する

例1) border-radius

```
p { float : left; }
```

例2) floatの解消方法



①A,Bは通常の表示位置



②Aにfloatをleftに指定するとBが下に入り込む



③Bにもfloatのleftを指定すると横並びになる

181

第4章 CSS3の設定

4.2 レイアウト

4.2.1 フロート

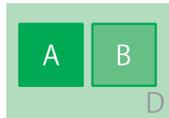
- 「float」を指定した子要素を囲む親要素が、子要素の下に入り込む場合もある
その場合の解消方法は、「clearfix」を使う
- 「clearfix」は、「float」の不具合を解消するためのテクニック



例1) clearfix

```
.clearfix::after{
  content: "";
  display: block;
  clear: both;
}
```

例2) floatの解消方法



①A,Bに
floatを指定
すると



②A,Bを囲んで
る親要素Dが下
に入り込む



③親要素Dに
clearfixを指定
して解消

182

第4章 CSS3の設定

-- CSSファイル --

```
:
01 @charset "UTF-8";
02 @font-face{
03 font-family: "lora-mediumItalics";
:
127 .footer-container ul li {
128 float: left;
129 }
130 /* ---フッターの設定---*/
:
```

• 実行結果



183

第4章 CSS3の設定

4.2 レイアウト

4.2.2 フレックスボックス

- 「Flexbox」(フレックスボックス)は、要素を横や縦に並べて配置
- フレックスボックスを指定するには、親要素に「`display : flex ;`」を適用
- 横並びにする要素をまとめて囲う親要素を 「フレックスコンテナ」といい、
その中にある横並びになる子要素を 「フレックスアイテム」という
- 設定値「flex」は、ブロックボックスの要素に設定
- 設定値「inline-flex」は、インラインブロックの要素に設定



184

第4章 CSS3の設定

4.2 レイアウト

4.2.2 フレックスボックス

- `flex-direction` プロパティを指定して、子要素の並び向きを指定
- 「row」は初期値で、子要素を左から右に配置 例1)
「row-reverse」はその逆で、子要素を右から左に配置 例2)
- 「column」は、子要素を上から下に配置 例3)
「column-reverse」は、その逆で、子要素を下から上に配置 例4)

例) flexの設定

```
.container {  
  display: flex;  
  flex-direction: row ;  
}
```

例1) rowの設定



例2) row-reverseの設定



例3) columnの設定



例4) column-reverseの設定

185

第4章 CSS3の設定

4.2 レイアウト

4.2.2 フレックスボックス

- `flex-wrap`プロパティを指定して、子要素の折り返し方を設定
- 「`nowrap`」は、初期値で、子要素を折り返しせずに一行で配置 例1)
- 「`wrap`」は、子要素を複数行に上から下へ折り返して配置 例2)
- 「`wrap-reverse`」は、「`wrap`」の逆で、子要素を複数行に下から上へ折り返して配置 例3)

例1) nowrapの設定



例2) wrapの設定



例3) wrap-reverseの設定



186

第4章 CSS3の設定

4.2 レイアウト

4.2.2 フレックスボックス

- 「`flex-flow`」プロパティは、`flex-direction`と`flex-wrap`をまとめて指定
- 「`justify-content`」は、子要素の水平方向の揃え方を指定
- 「`flex-start`」は、初期値で、行の開始位置から左揃えに配置 例1)
- 「`flex-end`」は、行末から右揃えにする 例2)、「`center`」は、中央揃えに配置 例3)
- 「`space-between`」は、最初と最後の子要素を両端に配置し、残りの要素は均等に間隔をあけて配置 例4)
- 「`space-around`」は、両端の子要素も含め、均等に間隔をあけて配置 例5)

例) flex-flowの設定

```
.container{  
  display: flex;  
  flex-flow: row wrap  
}
```

例1) flex-startの設定



例3) centerの設定



例4) space-betweenの設定



例2) flex-endの設定



例5) space-aroundの設定



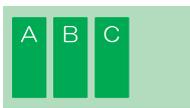
187

第4章 CSS3の設定

4.2 レイアウト

4.2.2 フレックスボックス

- 「`align-items`」プロパティは、子要素の垂直方向の揃え方を指定
- 「`stretch`」は、親要素の高さ、又はコンテンツの一番多い子要素の高さに合わせて配置 例1)
- 「`flex-start`」は、親要素領域の開始位置から上揃えに配置 例2)
- 「`flex-end`」は、親要素領域の終点から下揃えに配置 例3)
- 「`center`」は、子要素を中央揃えに配置 例4)
- 「`baseline`」は、子要素をベースラインで揃えて配置 例5)



例1) stretch
の設定



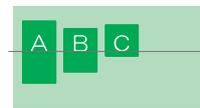
例2) flex-start
の設定



例3) flex-end
の設定



例4) centerの
設定



例5) baseline
の設定

188

第4章 CSS3の設定

4.2 レイアウト

4.2.2 フレックスボックス

- 「`align-content`」プロパティは、子要素が複数行の時の揃え方を指定
- 「`stretch`」は、初期値で、親要素の高さに合わせて子要素を広げて配置 例1)
- 「`flex-start`」は、親要素の開始位置から子要素を上揃えに配置 例2)
- 「`flex-end`」は、親要素の終点から子要素を下揃えに配置 例3)
- 「`center`」は、子要素を中央揃えに配置 例4)



例1) stretchの設定



例2) flex-startの設定



例3) flex-endの設定



例4) centerの設定

189

第4章 CSS3の設定

4.2 レイアウト

4.2.2 フレックスボックス

- 「space-between」は、最初と最後の子要素を上下の端に配置し、残りの要素は均等に間隔をあけて配置 例1)
- 「space-around」は、上下端にある子要素も含め、均等に間隔をあけて配置 例2)



例1) space-betweenの設定



例2) space-aroundの設定

第4章 CSS3の設定

-- CSSファイル --

```
:
01 @charset "UTF-8";
02 @font-face{
03 font-family: "lora-mediumltalics";
:
151 .footer-container ul li {
152 float: left;
153 }
154 /* ---フッターの設定---*/
:
```

• 実行結果



第4章 CSS3の設定

-- CSSファイル --

```
:
01 @charset "UTF-8";
02 @font-face{
03 font-family: "lora-mediumItalics";
   :
164 .footer-container ul li {
165     float: left;
166 }
167 /* ---フッターの設定---*/
   :
```

• 実行結果



The screenshot shows a web browser window with the URL localhost8080/HTML_site/index.html. The page content includes a section titled "購入方法" (Purchase Method) with a list of steps: 1. Webページにアクセス, 2. 申込フォームに入力, 3. 購入金額を振込する, 4. 商品の発送. To the right is a table titled "全国価格別送料一覧表" (National Price and Shipping Fee Table).

注文金額	本州・四国	北海道・九州	沖縄
3,000円以上	500円	700円	950円
5,000円以上	700円	850円	
10,000円以上	1,500円		
備考	購入金額が¥20,000円を超える場合は、送料が変わる場合がありますのでお問合せください。		

192

第4章 CSS3の設定

-- CSSファイル --

```
:
01 @charset "UTF-8";
02 @font-face{
03 font-family: "lora-mediumItalics";
   :
177 .footer-container ul li {
178     float: left;
179 }
180 /* ---フッターの設定---*/
   :
```

• 実行結果



The screenshot shows a web browser window with the URL localhost8080/HTML... The page displays a website layout with a header, a "Topics" section with three images, a "flower" section with a large image, and a "購入方法" (Purchase Method) section with a table. The table is identical to the one in the previous screenshot.

193

第4章 CSS3の設定

-- CSSファイル --

```
:
01 @charset "UTF-8";
02 @font-face{
03 font-family: "lora-mediumItalics";
:
186 .goal h3{
187 margin: 0;
188 }
189 /* ---フッターの設定---*/
:
```

• 実行結果



194

第4章 CSS3の設定

4.2 レイアウト

4.2.3 グリッドレイアウト

- 「**grid**」プロパティは、グリッド（格子状）レイアウトを指定
- 並べる子要素を囲う親要素を「**グリッドコンテナ**」といい、その中の子要素を「**グリッドアイテム**」という
- グリッドコンテナになる親要素に「**display : grid ;**」を指定することで、グリッドレイアウトを指定
- 「**grid-template-columns**」プロパティは、数値を半角スペースで区切り複数列の幅を指定
- 「**grid-template-rows**」プロパティは、数値を半角スペースで区切り複数行の高さを指定

例1) clearfix

```
.container {
display: grid;
grid-template-columns: 200px 200px 200px ;
}
```

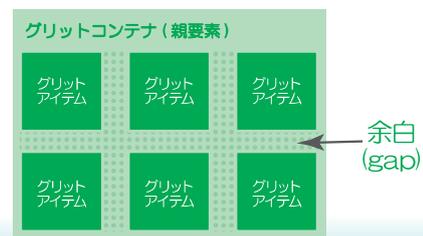
195

第4章 CSS3の設定

4.2 レイアウト

4.2.3 グリッドレイアウト

- 「gap」プロパティは、グリッドアイテム間の余白を設定
数値1つで数値を設定した場合は、すべて同じ大きさで、
数値2つの数値を設定した場合は、1つ目が水平方向、2つ目が垂直方向の設定
- グリッドで使用できる単位「fr」は、fractionの略で意味は「比率」
親要素から見た子要素の大きさを割合で指定
- 「px」(ピクセル)で幅を指定した場合は、固定のまま
「fr」で指定すると、親要素のサイズが変更すると子要素もリサイズに合わせて変更



196

第4章 CSS3の設定

-- CSSファイル --

```
:
01  @charset "UTF-8";
02  @font-face{
03  font-family: "lora-mediumitalics";
:
189 .goal h3{
190     margin: 0;
191 }
192 /* ---フッターの設定---*/
:
```

• 実行結果



197

第4章 CSS3の設定

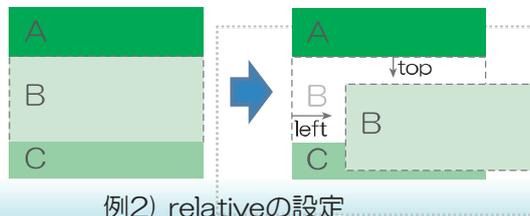
4.2 レイアウト

4.2.4 ポジション

- 「**position**」プロパティは、ボックスの位置を指定
位置の指定は、「**top**」「**bottom**」「**right**」「**left**」プロパティを組み合わせで使用
各プロパティは、単位付き数値で指定
- 設定値は、「**static**」「**relative**」「**absolute**」「**fixed**」の4つ
- 「**static**」は、初期値で位置指定をしない設定
- 「**relative**」は相対配置で、指定した要素が通常配置される位置を起点とし
「**top**」「**bottom**」「**right**」「**left**」プロパティで指定された場所へ配置 例1、2)

例1) positionの設定

```
div{  
  position : relative;  
  top : 50px ;  
  left : 100px ;  
}
```



例2) relativeの設定

198

第4章 CSS3の設定

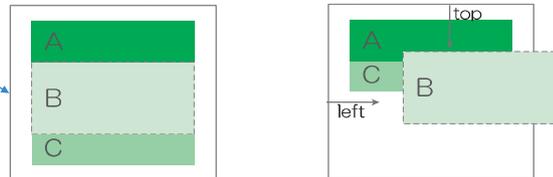
4.2 レイアウト

4.2.4 ポジション

- 「**absolute**」は絶対配置で、ブラウザウィンドウ枠を起点とし
「**top**」「**bottom**」「**right**」「**left**」プロパティで指定された場所へ配置
- 「**absolute**」を指定された要素は、元の配置される場所から取り除かれ存在しないものとして処理される 例3)

例3) absoluteの設定

ブラウザ
ウィンドウ



- 「**absolute**」を指定した要素の親要素に「**static**」以外のpositionプロパティが指定されている場合は、その親要素を起点として配置
他の「**absolute**」「**fixed**」などを設定すると、親要素自体の位置が変わり不具合
親要素を起点とする場合は、「**relative**」を設定

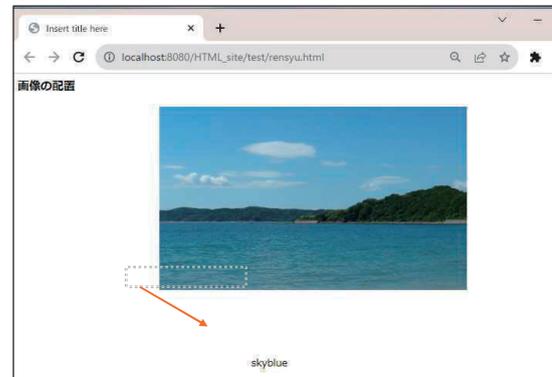
199

第4章 CSS3の設定

-- CSSファイル --

```
:  
76 .blue figcaption{  
77     position:relative;  
78     top: 100px;  
79     left: 150px;  
80 }
```

• 実行結果



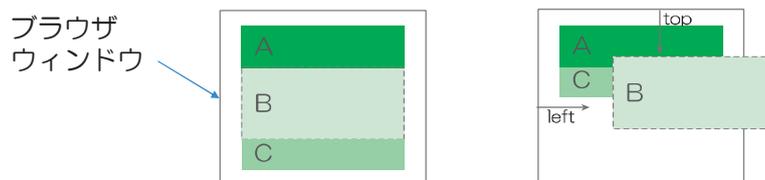
200

第4章 CSS3の設定

4.2 レイアウト

4.2.4 ポジション

- 「fixed」は、ブラウザウィンドウを基準に、「top」「bottom」「right」「left」プロパティで指定された場所へ配置
- 「absolute」と「fixed」の違いは、スクロールされた時に
「absolute」は、スクロールに応じて動く
「fixed」は、固定されて動かない



例1) fixedの設定

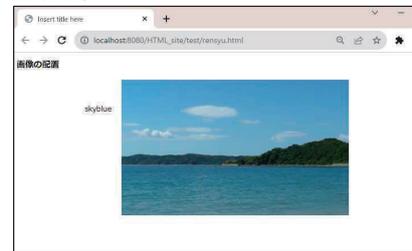
201

第4章 CSS3の設定

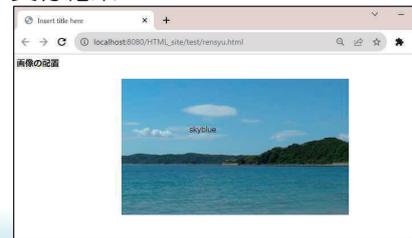
-- CSSファイル --

```
:  
81 .blue figure{  
82     position: relative;  
83 }  
:
```

• 実行前



• 実行結果



202

第4章 CSS3の設定

4.3 ナビゲーションのデザイン

4.3.1 リスト（ナビゲーション）

- リスト要素は、Webページのナビゲーションを指定
- リスト要素の初期設定で設定されている、余白（margin、padding）をリセット
- リスト要素の頭につく図形や番号は、「list-style : none ;」で非表示
- リストを水平方向に並べるため、ul要素に対して、フレックスボックスを設定
- li要素に「box-sizing : border-box ;」を設定
ボタンの区切り線を設定した際のレイアウト崩れを防ぐ

203

第4章 CSS3の設定

4.3 ナビゲーションのデザイン

4.3.2 リンク (ナビゲーション)

- リンク(a要素)の初期値設定で、テキストが青くなり下に傍線が表示
その傍線を「`text-decoration: none ;`」指定で非表示にする
- 疑似クラスの「`: link`」を使用しリンクのデザインを設定
- パディングでテキスト周囲に余白を設定し、ボタンの領域を大きく設定
- 疑似クラスの「`:hover`」をa要素に設定して、
カーソルがリンクテキストに重なったときに色が変わるように設定
- カーソルが指差しマークのポインタに変わるa要素の領域を広げるため、
リンク (a要素)に「`display : block ;`」を設定

例1) ナビゲーション



204

第4章 CSS3の設定

-- CSSファイル --

```
:
01 @charset "UTF-8";
02 @font-face {
03   font-family: "lora-mediumItalics";
   :
216 .goal h3 {
217     margin: 0;
218 }
219 /* ---フッターの設定--- */
:
```

• 実行結果



205

第4章 CSS3の設定

4.3 ナビゲーションのデザイン

4.3.3 グラデーション

- 「`linear-gradient()`」を指定することでグラデーション表示を設定
通常は、「`background-image`」のプロパティ値として設定する
- 設定値は、①方向、グラデーションの方向を数値で指定
②色と③位置は、グラデーションに使用する各色と色の開始位置を指定
- 色は、複数設定可能、位置については、0%から100%の間で指定
- 各色の位置を指定しなかった場合は、各色が均一に設定

例1) グラデーションの設定

```
.container {  
    background-image: linear-gradient( 180deg, #fff 0%, #000 100% );  
}
```

①方向 ②色 ③位置 ④色 ⑤位置
 ↓
 ③位置

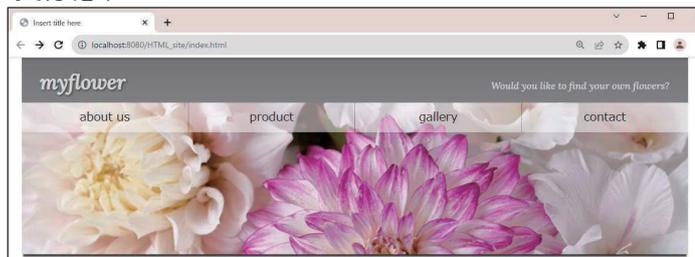
206

第4章 CSS3の設定

-- CSSファイル --

```
:  
01 @charset "UTF-8";  
02 @font-face {  
03 font-family: "lora-mediumItalics";  
:  
217 .goal h3 {  
218 margin: 0;  
219 }  
220 /* ---フッターの設定---*/  
:
```

• 実行結果



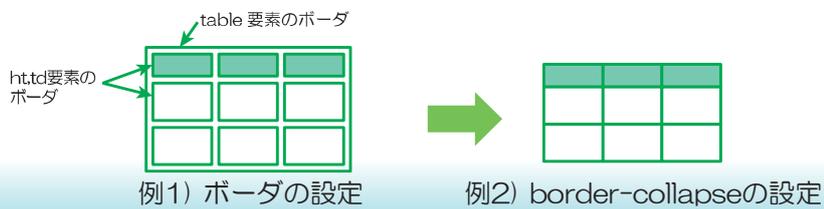
207

第4章 CSS3の設定

4.4 表組のデザイン

4.4.1 境界線の設定

- 表組で線の設定は、table要素のborder属性で指定 ⇒HTMLから廃止予定→CSSで設定
- ボーダの設定は、「table」「td」「th」要素に設定が必要
- 「table」「td」「th」要素にボーダを設定した場合、例1)のように table要素のボーダと td や th 要素のボーダとの間に余白がある状態が表示
- 例2)のように ボーダとの間の余白がない状態の表を作るには、table要素にCSSで「border-collapse : collapse ;」を設定
- 「border-collapse : collapse ;」を設定すると、table要素と td や th s 要素のボーダが重なり合って一本の線に見える



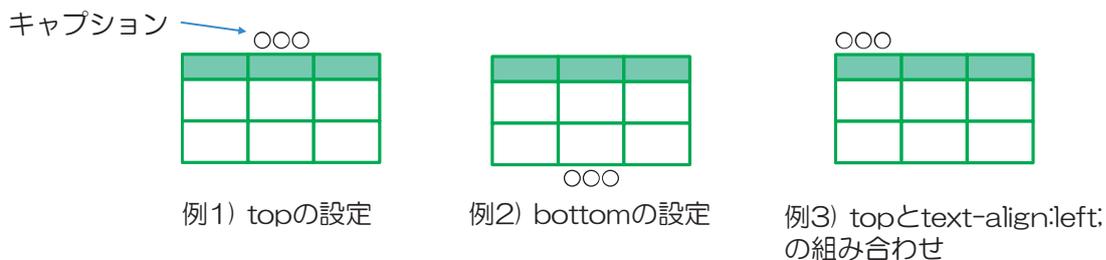
208

第4章 CSS3の設定

4.4 表組のデザイン

4.4.2 キャプションの設定

- 表のキャプションの設定は、「caption-side」で指定
設定値は、「top」「bottom」の値で設定
- 表のcaptionの位置を変更する際は、「caption-side」と「text-align」プロパティを組み合わせ使用



209

第4章 CSS3の設定

-- CSSファイル --

```
:
01 @charset "UTF-8";
02 @font-face{
03 font-family: "lora-mediumItalics";
:
239 .goal h3{
240 margin: 0;
241 }
242 /* ---フッターの設定---*/
:
```

・実行結果



210

第4章 CSS3の設定

4.5 フォームのデザイン

4.5.1 フォーカス

- ・疑似クラスの「:focus」で、フォーム部品がフォーカスを持っているときの色の設定
- ・フォーカスとは、マウスでクリックするなどで入力対象を選択した時、フォーム部品の縁に色が付き**ハイライト表示になること**
フォーカスを設定することで、**ユーザービリティ**(使いやすさ)の向上につながる
- ・フォーカス時の縁の色を変更する際は、「outline」プロパティを使用して変更
- ・設定値は、borderのショートハンドの様に「**線種 線幅 色**」

A form with three input fields: '名前' (Name), 'メールアドレス' (Email address), and '電話番号' (Phone number). The 'メールアドレス' field is highlighted with a blue border, indicating it is the active element.

例1) 通常の場合

A form with three input fields: '名前' (Name), 'メールアドレス' (Email address), and '電話番号' (Phone number). The 'メールアドレス' field is highlighted with a blue border, indicating it is the active element.

例2) フォーカスした場合

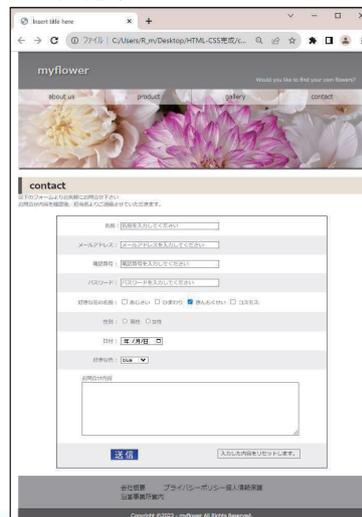
211

第4章 CSS3の設定

-- CSSファイル --

```
:
01 @charset "UTF-8";
02 @font-face{
03 font-family: "lora-mediumItalics";
:
297 ul.inline li{
298     float: left;
299 }/*-----フォームの設定-----*/
:
```

• 実行結果



212


```

01 <!DOCTYPE html>
02 <html [ ]>
03 <[ ]>
04 <meta charset="[ ]">
05 <title>桃太郎</title>
06 </[ ]>
07 <body>
08 <[ ]>桃太郎（ももたろう）</[ ]>
09 <[ ]>日本のおとぎ話の一つ。桃の実から生まれた男子「桃太郎」が、お爺さんお婆さんから<[ ]>
10 黍団子（きびだんご）を もらって、イヌ、サル、キジを家来にし、鬼ヶ島まで鬼を退治しに行く物
    語。</[ ]>
11 <h2>その1 ストーリー</h2>
12 <[ ]>以下のような粗筋のものが「標準型」となっている。<[ ]>
13 桃から生まれた桃太郎は、老婆老爺に養われ、鬼ヶ島へ鬼退治に出征、道中遭遇する<[ ]>
14 イヌ、サル、キジにきび団子を褒美に家来とし、鬼の財宝を持ち帰り、郷里に凱旋する</[ ]>
15 <[ ]>異本</[ ]>
16 <p>より古い系統の桃太郎説話は、この「標準型」とは異なるものである。<br>
17 19世紀初頭にはじめてみられるもので、それまでの草双紙では桃を食べたお爺さんお婆さんが<br>
18 若返り出産する型（「回春型」）が主流だった</p>
19 <[ ]>口承文学の異伝</[ ]>
20 <p>口承話では「果生型」が多いとされているが、その他、桃太郎の誕生の仕方については<br>
21 <赤い箱と白い箱が流れて来て、赤い箱を拾ったら赤ん坊が入っていた>など差異のあるものが多数伝
    わっている。</p>
22 <[ ]>その2 由来</[ ]>
23 <p>特定の伝説に拠る物語の由来については諸説存在し、それぞれ論争のあるところである。<br>
24 桃太郎の起源を岡山とする説に関して、戦前の頃までその支持は、愛知県や香川県をゆかりとする説に
    大きく後れを取っていたが、<br>
25 1960年以降の岡山地域の促進運動によってその知名度が上がっている</p>
26 <h3>成立過程</h3>
27 <p>物語としての成立年代は正確には分かっていないが、原型（口承文学）の発祥は室町時代末期から
    江戸時代初期頃とされる</p>
28 </body>
29 </html>

```

問題2 (難易度 ☆)

- ・ 二酸化炭素の概要について紹介するページを作成する
- ・ HTMLプログラムは、「webapp」フォルダの「exercise02.html」ファイルに保存する
- ・ プログラム作成後に、「exercise02.html」ファイルを実行し、Webブラウザに表示される実行結果を確認する

[要件]

- ・ 「exercise02.html」ファイル内にHTMLの基本構文を記述し、その中に見出し要素と段落要素、ルビと上付き文字、下付き文字を使用する
- ・ Webページの言語は「日本語」を指定、エンコードは「UTF-8」とする

[実行結果]

にさんかたんそ
二酸化炭素

炭素の酸化物の一つで、化学式が CO_2 と表される無機化合物^{※1}である。化学式から「シーオーツー」とも呼ばれる。地球温暖化対策の文脈などで「カーボンフリー」「カーボンニュートラル^{※2}」など「カーボン」が使われることがあるが、これは二酸化炭素由来の炭素を意味する。

性質

二酸化炭素は温室効果を持ち、地球の気温を保つのに必要な温室効果ガスの一つである。常温常圧では無色無臭の気体^{※3}。常圧では液体^{※4}にならず、 -79°C で凝華^{※5}して固体（ドライアイス）となる。水に比較的よく溶け、水溶液（炭酸水）は弱酸性を示す。このためアルカリ金属およびアルカリ土類金属の水酸化物の水溶液および固体は二酸化炭素を吸収して、炭酸塩^{※6}または炭酸水素塩を生ずる。高压で二酸化炭素の飽和水溶液を冷却すると八水和物^{※7} $\text{CO}_2 \cdot 8 \text{H}_2\text{O}$ を生ずる。

脚注

※1 有機化合物以外の化合物であり、具体的には単純な一部の炭素化合物と、炭素以外の元素で構成される化合物である。無機には「生命力を有さない」という意味がある。

※2 二酸化炭素など温室効果ガスの排出量と吸収量を均衡させ、その排出量を「実質ゼロ」に抑える、という概念。日本語で直訳すると炭素中立となる。

※3 物質の状態のひとつであり、一定の形と体積を持たず、自由に流動し圧力の増減で体積が容易に変化する状態のこと。「ガス体」とも言う。

※4 物質の状態（固体・液体・気体）の一つである。気体と同様に流動的で、容器に合わせて形を変える。液体は気体に比して圧縮性が小さい。気体とは異なり、容器全体に広がることはなく、ほぼ一定の密度を保つ。液体特有の性質として表面張力があり、それによって「濡れ」という現象が起きる。

※5 元素や化合物が液体を経ずに気体から固体へと相転移する現象。温度と圧力の交点が三重点より下へ来た場合に起こる。

※6 炭酸イオンを含む化合物の総称である。英語の carbonate は炭酸塩と炭酸イオンの他、炭酸エステル、炭酸塩化、炭化、飲料などに炭酸を加える操作のことも指す

※7 無機化学および有機化学において、水分子を含む物質のことを表す用語である。含まれる水のことは、水和水と呼ぶ。

参考サイト ウィキペディア : <https://ja.wikipedia.org/wiki/>

exercise02.html (続き)

34 </html>

問題3 (難易度 ☆☆)

- ・あるパン屋さんの本日のおすすめパンを紹介するページを作成する
- ・HTMLプログラムは、「webapp」フォルダの「webapp」フォルダに保存する
- ・プログラム作成後に、「exercise03.html」ファイルを実行し、Webブラウザに表示される実行結果を確認する

[要件]

- ・「exercise03.html」ファイル内にHTMLの基本構文を記述し、その中に見出し要素と段落要素、リストと挿絵として3枚の画像(be-konmayo.jpg、hanamaru.jpg、kurowassan.jpg)を使用し、代替テキストとキャプションにそれぞれのパンの名前を入れる
- ・画像の名前は上から、「ベーコンマヨ」「はな丸パン」「クロワッサン」とする
- ・画像は「exercise_img」フォルダの画像を使用し、サイズは、幅300px、高さ200pxとする
- ・リストは、「home」「bread」「Deli bread」「access」「contact」とする
- ・Webページの言語は「日本語」を指定、エンコードは「UTF-8」とする

[実行結果]



exercise03.html

```
01 <!DOCTYPE html>
02 <html lang="ja">
03 <head>
04 <meta charset="UTF-8">
05 <title>おいしいパン屋さん</title>
06 </head>
07 <body>
08 <h1>おいしいパン屋さん</h1>
09 <ul>
10
11
12
13
14
15 </ul>
16 <h2>本日のおすすめパン</h2>
17 <figure>
18
19
20 </figure>
21 <p>ふんわりパンに黒コショウの利いたベーコンとマヨネーズとで焼き上げたお惣菜パン</p>
22 <figure>
23
24
25 </figure>
26 <p>お花をモチーフにした甘くておいしいミルクパン</p>
27 <figure>
28
29
30 </figure>
31 <p>ふんわりサクサク、バター風味がたっぷりクロワッサン</p>
32 </body>
33 </html>
```

問題4 (難易度 ☆☆)

- ・赤ずきんの概要について紹介するページを作成する
- ・HTMLプログラムは、「webapp」フォルダの「exercise04.html」ファイルに保存する
- ・プログラム作成後に、「exercise04.html」ファイルを実行し、Webブラウザに表示される実行結果を確認する

【要件】

- ・「exercise04.html」ファイル内にHTMLの基本構文を記述し、その中に見出し要素と段落要素、順番を付加したリストとリストを使用する
- ・Webページの言語は「日本語」を指定、エンコードは「UTF-8」とする
- ・順番を付加したリストとリストは、下記のテキストを使用する

●順番を付加したリストテキスト

1. 赤ずきんと呼ばれる女の子がいた。彼女はお使いを頼まれて森の向こうのおばあさんの家へと向かうが、その途中で一匹の狼に遭い、唆されて道草をする。
2. 狼は先回りをしておばあさんの家へ行き、家にいたおばあさんを食べてしまう。そしておばあさんの姿に変装して、赤ずきんが来るのを待つ。
3. 赤ずきんがおばあさんの家に到着すると、おばあさんに化けていた狼に赤ずきんは食べられてしまう。
4. 満腹になった狼が寝入っていたところを通りかかった猟師が気づき、狼の腹の中から二人を助け出す。
5. 赤ずきんは「言いつけを守らなかったから酷い目に遭った」と反省し、二度と道草をしたり知らない人の誘いに乗らないことを誓う。

●リストテキスト

「赤ずきんからの派生作品（英語版）」「赤ずきん（曖昧さ回避）」「アカンずきん」
「マイメロディの赤ずきん」「リトル・レッド レシピ泥棒は誰だ!?!」「おかしな赤頭巾」

【実行結果】

The screenshot shows a web browser window with the URL `localhost:8080/HTML_exercise/exercise04.html`. The page content is as follows:

赤ずきん

「赤ずきん」のあらすじ

1. 赤ずきんと呼ばれる女の子がいた。彼女はお使いを頼まれて森の向こうのおばあさんの家へと向かうが、その途中で一匹の狼に遭い、唆されて道草をする。
2. 狼は先回りをしておばあさんの家へ行き、家にいたおばあさんを食べてしまう。そしておばあさんの姿に変装して、赤ずきんが来るのを待つ。
3. 赤ずきんがおばあさんの家に到着すると、おばあさんに化けていた狼に赤ずきんは食べられてしまう。
4. 満腹になった狼が寝入っていたところを通りかかった猟師が気づき、狼の腹の中から二人を助け出す。
5. 赤ずきんは「言いつけを守らなかったから酷い目に遭った」と反省し、2度と道草をしたり知らない人の誘いに乗らないことを誓う。

グリム童話の赤ずきん

グリム童話の「赤ずきん」は長い間、ドイツのとある農家の非識字者である老婆が語る話を聞き取り、手を加えずに原稿に起こし出版したものであると信じられていた。しかし、実は話の提供者にそんな人物は一人もいないということがハインツ・レケの研究により判明した。赤ずきんの話の提供者は、ヘッセン選帝侯国に属する高級官僚ハッセンプフルーク家の娘たちである。

赤ずきんモチーフ

- ・ 赤ずきんからの派生作品（英語版）
- ・ 赤ずきん（曖昧さ回避）
- ・ アカンずきん
- ・ マイメロディの赤ずきん
- ・ リトル・レッド レシピ泥棒は誰だ!?!
- ・ おかしな赤頭巾

参考サイト ウィキペディア : <https://ja.wikipedia.org/wiki/>

exercise04.html

```
01 <!DOCTYPE html>
02 <html lang="ja">
03 <head>
04 <meta charset="UTF-8">
05 <title>赤ずきんちゃん</title>
06 </head>
07 <body>
08 <h1>赤ずきん</h1>
09 <p>「赤ずきん」のあらすじ</p>
10
11
12
13
14
15
16
17
18
19
20 <h2>グリム童話の赤ずきん</h2>
21 <p>グリム童話の「赤ずきん」は長い間、ドイツのとある農家の非識字者である老婆が語る話を聞き
22 取り、手を加えずに原稿に起こし出版したものであると信じられていた。<br>
23 しかし、実は話の提供者にそんな人物は一人もいないということがハインツ・レレケの研究により判
24 明した。<br>
25 赤ずきんの話の提供者は、ヘッセン選帝侯国に属する高級官僚ハッセンプフルーク家の娘たちである。
26 </p>
27 <h3>赤ずきんモチーフ</h3>
28
29
30
31
32
33 </body>
34 </html>
```

問題5（難易度 ☆☆☆）

- ・こぶとりじいさんの概要について紹介するページを作成する
- ・HTMLプログラムは、「webapp」フォルダの「exercise05.html」ファイルに保存する
- ・プログラム作成後に、「exercise05.html」ファイルを実行し、Webブラウザに表示される実行結果を確認する

[要件]

- ・「exercise05.html」ファイル内にHTMLの基本構文を記述し、その中に見出し要素と段落要素、セクショニングコンテンツのセクション要素、アーティクル要素を使用し文書構造を作成する
- ・プログラム作成後に、「exercise05.html」ファイルを実行し、Webブラウザに表示される実行結果を確認する

[実行結果]

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/HTML_exercise/exercise05.html'. The page content is as follows:

こぶとりじいさん

こぶとりじいさんは、日本の民話。老人が、鬼に餌草として頬の瘡を取られる説話群で、「隣の爺型民話」の大分類のうちに数えられる。二人の翁（老人）が連夜で鬼の宴に参加し、隣の翁は逆に瘡を増やされる型が典型であるが、民話の常として様々な類型があり、ストーリーも様々である。本話は開歌吾『日本昔話集成』や『日本昔話大成』で「隣の爺型」の大分類のうち194番「瘡取爺」に分類され、日本全国各地の例が挙げられている。

異本例

隣の翁に瘡がないパターン

越後などで採取。瘡の無い隣の翁は鬼の宝物が目当てで参加。「日本昔話百選」などに収録。鬼から踊りの褒美に金銀をやると言われた踊りがとても上手なじいさんは「金銀は要らない。じゃまな瘡を取ってくれ」と言って瘡を取ってもらう。隣のじいさん（瘡は無い。鬼も昨日の翁と別人であることは承知している）は、鬼から「お前の欲しいものは何だ。」と訊かれ、「金銀が欲しい」とそのまま言うのは鬼に軽蔑されると思い、「わしが欲しいのは、昨日のじいさんが「要らない」と言ったもの（金銀のこと）だ」と答えたので、鬼は誤解し「昨日のじいさんにとって要らないもの（瘡のこと）」を褒美に与え、にわたりの声にあわてて去ってしまう。

隣の翁が瘡を取ってもらうことに成功する（出羽で採取）

隣のじさが「待て待て、鬼っこ」と呼ぶと「昨日の面白え爺がまた来てんでねべか」と鬼が戻ってきた。鬼は「じさ、昨日瘡取ったのに、また瘡つけてきて駄目でねか」言うてバンと叩くと瘡とれてねくなってまったど。隣のじさは喜んで鬼と一緒にバカスグと朝まで夢中になって踊ったど…。瘡が増えてしまい落ち込む意地悪な翁を見かねた陽気な翁が踊りで元気付けようとする。悩んでいるのが馬鹿らしくなり、歌と踊りを教えてもらううちに自然と瘡が取れてしまう。明るい性格になった隣の翁は村人からも好かれるようになった。

話の源

同じ型の話は古く13世紀成立の『宇治拾遺物語』第三話に見られる冒頭で翁の瘡の大きさが「大柑子（おおこうじ）」（大ぶりのみかん）ほどであると描写されるが、これは現在の夏みかんの事だとも推察されている。また、この瘡のせいで人と交わることができず、薪を集めて生業にしていたと、職業が木樵であったことが明確になっている。

参考サイト ウィキペディア：<https://ja.wikipedia.org/wiki/>

exercise03.html

```

01 <!DOCTYPE html>
02 <html lang="ja">
03 <head>
04   <meta charset="UTF-8">
05   <title>こぶとりじいさん</title>
06 </head>
07 <body>
08
09
10   <p>こぶとりじいさんは、日本の民話。老人が、鬼に質草として頬の瘤を取られる説話群で、
11   「隣の爺型民話」の大分類のうちに数えられる。<br>
12   二人の翁（老人）が連夜で鬼の宴に参加し、隣の翁は逆に瘤を増やされる型が典型であるが、
13   民話の常として様々な類型があり、ストーリーも様々である。<br>
14   本話は関敬吾『日本昔話集成』や『日本昔話大成』で「隣の爺型」の大分類のうち194番「瘤
15   取爺」に分類され、日本全国各地の例が挙げられている。</p>
16
17   <p>越後などで採取。瘤の無い隣の翁は鬼の宝物が目当てで参加。「日本昔話百選」
18   などに収録。<br>
19   鬼から踊りの褒美に金銀をやると言われた踊りがとても上手なじいさんは「金銀は
20   要らない。じゃまな瘤を取ってくれ」と<br>
21   言って瘤を取ってもらう。隣のじいさん（瘤は無い。鬼も昨日の翁と別人であるこ
22   とは承知している）は、<br>
23   鬼から「お前の欲しいものは何だ。」と訊かれ、「金銀が欲しい」とそのまま言うの
24   は鬼に軽蔑されると思い、「わしが欲しいのは、<br>
25   昨日のじいさんが「要らない」と言ったもの（金銀のこと）だ」と答えたので、鬼
26   は誤解し「昨日のじいさんにとって要らないもの（瘤のこと）」を<br>
27   褒美に与え、にわたりの声にあわてて去ってしまう。</p>
28
29   <p>隣のじいさんが「待て待て、鬼っこ」と呼ぶと「昨日の面白え爺がまた来てんでねべ
30   か」と鬼が戻ってきた。鬼は「じい、昨日瘤取ったのに、<br>
31   また瘤つけてきて駄目でねか」言うてバンと叩くと瘤とれてねぐなまってまったど。
32   隣のじいさんは喜んで鬼と一緒にバガスグと朝まで夢中になって踊ったど…。<br>
33   瘤が増えてしまい落ち込む意地悪な翁を見かねた陽気な翁が踊りで元気付けようと
34   する。悩んでいるのが馬鹿らしくなり、歌と踊りを教えてもらううちに<br>
35   自然と瘤が取れてしまう。明るい性格になった隣の翁は村人からも好かれるよう
36   になった。</p>
37
38   <section>
39     <h2>話の源</h2>
40     <p>同じ型の話は古く13世紀成立の『宇治拾遺物語』第三話に見られる冒頭で翁の瘤の
41     大きさが「大柑子（おおこうじ）」（大ぶりのみかん）ほどであると<br>
42     描写されるが、これは現在の夏みかんの事だとも推察されている。また、この瘤のせ
43     いで人と交わることができず、薪を集めて生業にしていたと、<br>

```

HTML/CSS

exercise05.html (続き)

```
36     職業が木樵であったことが明確になっている</p>
37     </section>
38
39 </body>
40 </html>
```

問題6（難易度 ☆☆☆）

- ・あるケーキ屋さんのチーズケーキの種類を紹介するページを作成する
- ・HTMLプログラムは、「webapp」フォルダの「exercise06.html」ファイルに保存する
- ・プログラム作成後に、「exercise06.html」ファイルを実行し、Webブラウザに表示される実行結果を確認する

[要件]

- ・「exercise06.html」ファイル内にHTMLの基本構文を記述し、その中に見出し要素と段落要素、ヘッダー要素とメイン要素、フッター要素とセクショニングコンテンツのアーティクル要素、セクション要素を使用して作成すること

[実行結果]

参考サイト ウィキペディア：<https://ja.wikipedia.org/wiki/>

exercise06.html

01. <!DOCTYPE html>
02. <html lang="ja">
03. <head>
04. <meta charset="UTF-8">
05. <title>おいしいケーキ屋</title>
06. </head>
07. <body>
08.
09. <h1>おいしいケーキ屋</h1>

exercise06.html (続き)

```

10.    <p>チーズケーキ &lpar;cheesecake&rpar;は、温製のベイクトと湯煎焼きのスフレ、冷製のレアがある。</p>
11.
12.
13.
14.
15.
16.
17.
18.
19.    <section>
20.        <h3>ベイクト・チーズケーキ</h3>
21.        <p>オープンできつね色になるまで焼いたものはベイクト・チーズケーキと呼ばれる。ニューヨークチーズケーキと呼ぶこともある。<br>
22.        型にクリームチーズやカッテージチーズ、マスカルポーネ、リコッタチーズなどのやわらかく塩分の少ないチーズと砂糖、卵黄、<br>
23.        コーンスターチや小麦粉などをすり混ぜた生地を流しいれ、オープンで焼いたもの。</p>
24.    </section>
25.    <section>
26.        <h3>レア・チーズケーキ</h3>
27.        <p>火を通さず、クリームチーズなどに生クリームを混ぜ合わせたものを冷やし固めたものであり、クッキーを砕いて作ったクラスト生地を<br>
28.        下に敷くこともある。あるいはフィリングだけをババロアやムースのようにグラスに盛る「グラス・チーズケーキ」もある。<br>
29.        近年はガーゼで包んで販売するものが登場している。</p>
30.    </section>
31.    <h2>チーズケーキの歴史</h2>
32.    <p>チーズケーキの起源は古代ギリシャまでさかのぼり、紀元前776年の第一回古代オリンピックの期間中、アスリートたちに振る舞われていた。<br>
33.    また、紀元前160年ごろに書かれた農書『農業について (英語版)』には、libum、サビルム (savillum)、placenta という神に奉納された<br>
34.    チーズケーキのレシピが掲載されている</p>
35.
36.
37.
38.    <p>Copyright © 2015 cheesecake Corporation. All Rights Reserved.</p>
39.
40. </body>
41. </html>

```

問題7 (難易度 ☆☆☆☆)

- ・あるパン屋さんのWebサイトを作成する
- ・HTMLプログラムは、「webapp」フォルダの「exercise07.html」ファイルに保存する
- ・プログラム作成後に、「exercise07.html」ファイルを実行し、Webブラウザに表示される実行結果を確認する

[要件]

- ・「exercise07.html」ファイル内にHTMLの基本構文を記述し、その中に見出し要素と段落要素とヘッダー要素、メイン要素、フッター要素、セクショニングコンテンツのアーティクル要素、ナビ要素、リスト要素、パイパーリンクを使用して作成する
- ・挿絵として1枚画像(haigasyokupan.jpg)を使用し、代替テキストとキャプションにそれぞれのパンの名前を入れ、パンの名前は「胚芽ブレッド」とする
画像は「exercise_img」フォルダの画像を使用し、サイズは、幅300px、高さ200pxとする
- ・リスト要素の3項目にそれぞれのパイパーリンクを設定する
リンク先は、「contact」フォルダのindex.htmlと「item」フォルダのindex.htmlに設定する
また、「contact」フォルダと「item」フォルダのindex.htmlのリストにもパイパーリンクを設定し、ページ間を移動できるようにする
- ・文字化けの恐れがある記号については、特殊記号を使用する
- ・リストは、下記のテキストをする

●使用するテキスト

Plain bread

国産小麦粉を使ったオーガニックなおいしいパンを食べられる

- ・ home
- ・ item
- ・ contact

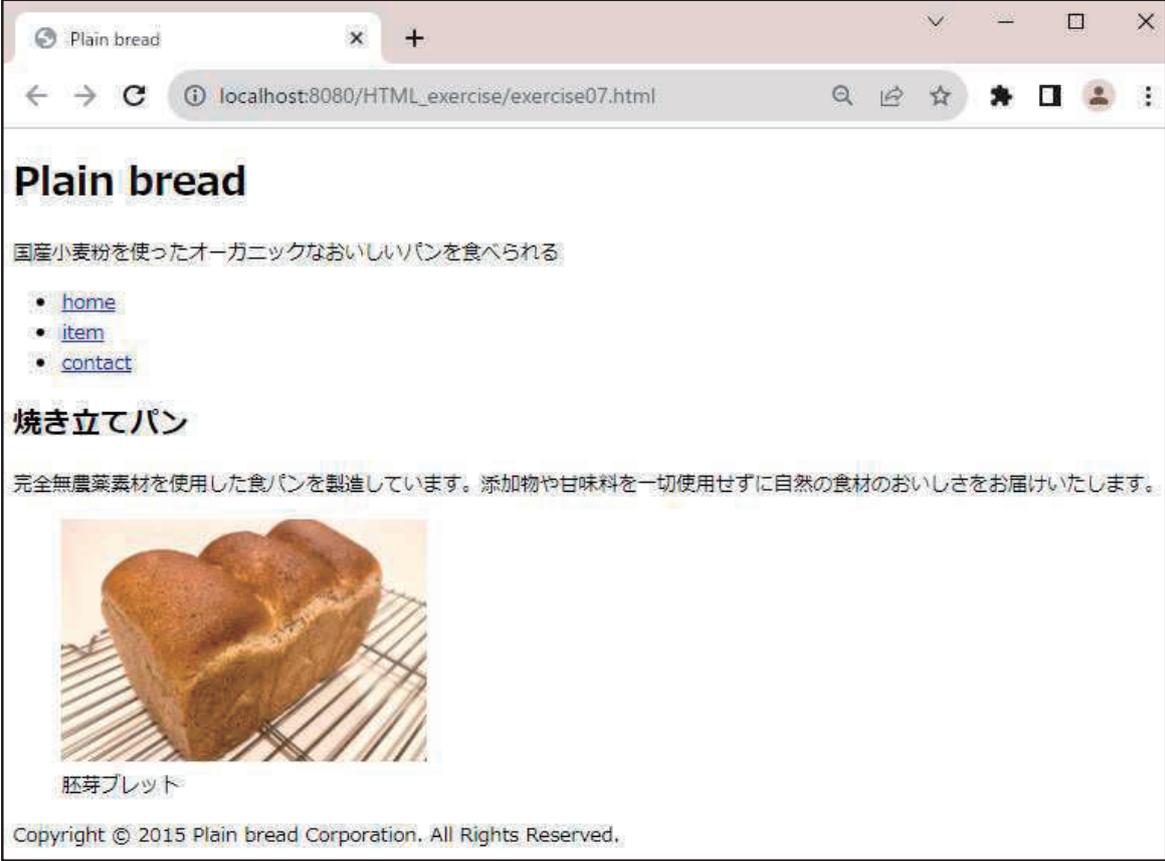
焼き立てパン

完全無農薬素材を使用した食パンを製造しています。他の添加物や甘味料を一切使用せずに自然の食材のおいしさをお届けいたします。

Copyright © 2015 Plain bread Corporation. All Rights Reserved.

問題7 続き

[実行結果]



The screenshot shows a web browser window with the following content:

- Page title: Plain bread
- Address bar: localhost:8080/HTML_exercise/exercise07.html
- Section header: **Plain bread**
- Text: 国産小麦粉を使ったオーガニックなおいしいパンを食べられる
- Navigation links: [home](#), [item](#), [contact](#)
- Section header: **焼き立てパン**
- Text: 完全無農薬素材を使用した食パンを製造しています。添加物や甘味料を一切使用せずに自然の食材のおいしさをお届けいたします。
- Image: A loaf of bread on a wire rack.
- Caption: 胚芽プレット
- Copyright: Copyright © 2015 Plain bread Corporation. All Rights Reserved.

問題8 (難易度 ☆☆☆☆)

- ・あるケーキ屋さんのメニュー紹介のページを作成する
- ・HTMLプログラムは、「webapp」フォルダの「exercise08.html」ファイルに保存する
- ・プログラム作成後に、「exercise08.html」ファイルを実行し、Webブラウザに表示される実行結果を確認する

[要件]

- ・「exercise08.html」ファイル内にHTMLの基本構文を記述し、その中に見出し要素と段落要素とヘッダー要素とメイン要素、フッター要素とセクショニングコンテンツのアーティクル要素、セクション要素、ナビ要素とリスト要素を使用して作成すること
- ・ページのタイトルに当たる見出しに画像を使用し、代替テキストを設定する
- ・また、挿絵として3枚画像(syouto.jpg、appurutaruto.jpg、itigonomu-su.jpg)を使用し、代替テキストとキャプションにそれぞれのケーキの名前を入れる
- ・ケーキの名前は、「ショートケーキ」「アップルタルト」「イチゴのムース」とする
- ・画像は「exercise_img」フォルダの画像を使用し、サイズは、幅200px、高さ200pxとする
- ・Webページの言語は「日本語」を指定、エンコードは「UTF-8」とする
- ・文字化けの恐れがある記号については、特殊記号を使用する
- ・リストは、下記のテキストを使用する

●使用するテキスト

デリシャス(タイトル:画像(exercise-07.gif) delicious(デリシャス)使用する)
自然素材を使った、おいしいケーキ、国産小麦粉・純生クリーム、オーガニックチョコレートなど安心して食べられる素材を使用

- ・ home
- ・ concept
- ・ menu
- ・ news

Menu

ショートケーキ

国産の卵を使用したこだわりのスポンジとなめらかなクリームに極上のイチゴを乗せました。

パティシエ自慢のショートケーキです。

アップルタルト

国産小麦を使用したこだわりタルトに完熟のリンゴを使用し甘酸っぱく仕上げました。サクサクの生地をお楽しみください。

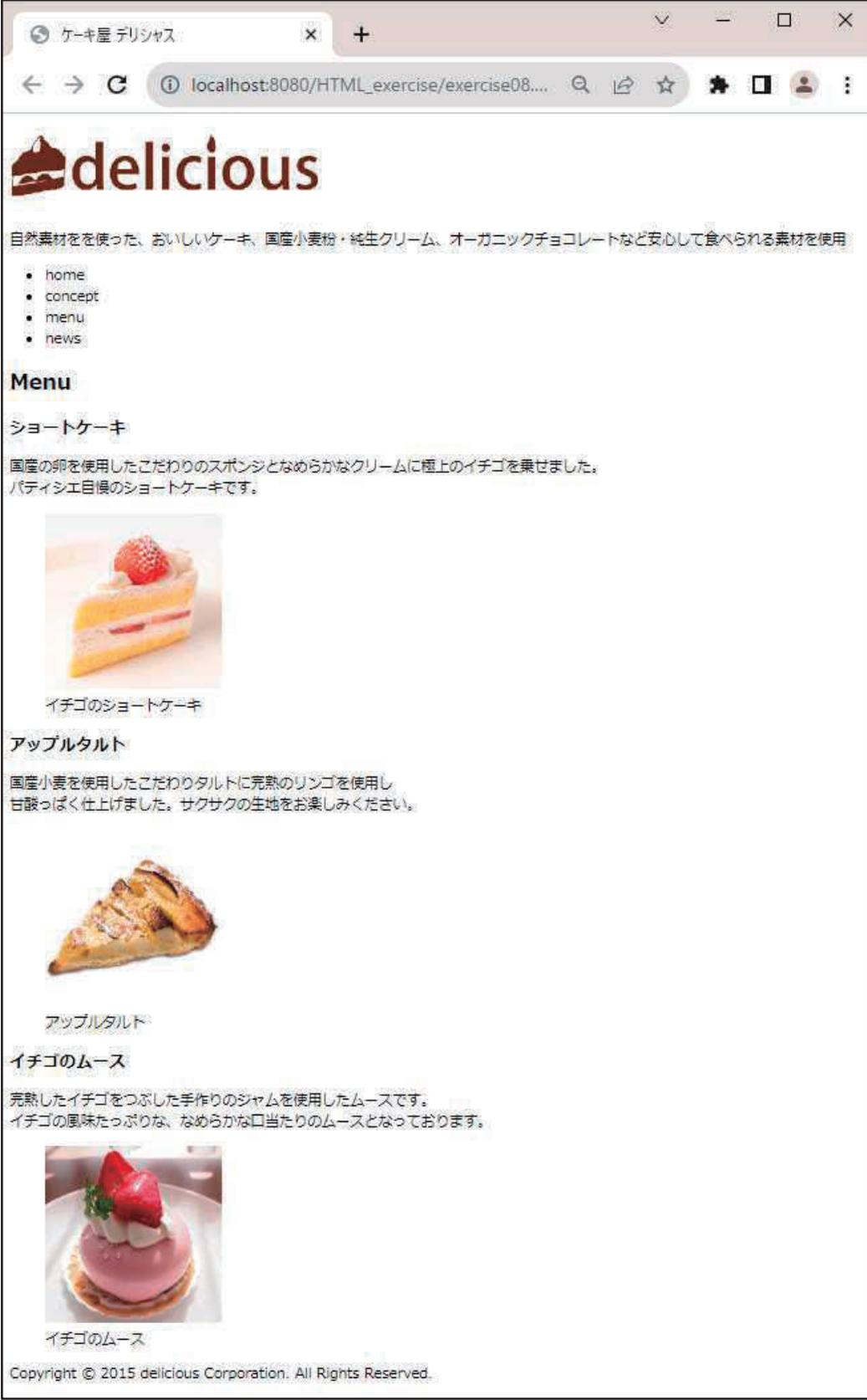
イチゴのムース

完熟したイチゴをつぶした手作りのジャムを使用したムースです。イチゴの風味たっぷりの、なめらかな口当たりのムースとなっております。

Copyright © 2015 delicious Corporation. All Rights Reserved.

問題8 続き

[実行結果]



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/HTML_exercise/exercise08...'. The website content includes a logo for 'delicious' with a cake icon, a paragraph about natural ingredients, a navigation menu with links for 'home', 'concept', 'menu', and 'news', and three featured cake items: Strawberry Shortcake, Apple Tart, and Strawberry Mousse. Each item has a description and a corresponding image.

自然素材を使った、おいしいケーキ、国産小麦粉・純生クリーム、オーガニックチョコレートなど安心して食べられる素材を使用

- home
- concept
- menu
- news

Menu

ショートケーキ

国産の卵を使用したこだわりのスポンジとなめらかなクリームに極上のイチゴを乗せました。パティシエ自慢のショートケーキです。



イチゴのショートケーキ

アップルタルト

国産小麦を使用したこだわりタルトに完熟のリンゴを使用し甘酸っぱく仕上げました。サクサクの生地をお楽しみください。



アップルタルト

イチゴのムース

完熟したイチゴをつぶした手作りのジャムを使用したムースです。イチゴの風味たっぷりの、なめらかな口当たりのムースとなっております。



イチゴのムース

Copyright © 2015 delicious Corporation. All Rights Reserved.

問題9（難易度 ☆☆☆☆☆）

- ・ある美術館の入場料一覧表のページを作成する
- ・HTMLプログラムは、「webapp」フォルダの「exercise09.html」ファイルに保存する
- ・プログラム作成後に、「exercise09.html」ファイルを実行し、Webブラウザに表示される実行結果を確認する

[要件]

- ・「exercise09.html」ファイル内にHTMLの基本構文を記述し、その中に見出し要素と段落要素とヘッダー要素とメイン要素、フッター要素とセクショニングコンテンツのアーティクル要素、ナビ要素とリスト要素、テーブル要素を使用して作成する
- ・入場料の価格表を作成する、また、線の設定は、テーブル要素に「border」属性を使用し、「border="1"」を設定する
- ・Webページの言語は「日本語」を指定、エンコードは「UTF-8」とする
- ・文字化けの恐れがある記号については、特殊記号を使用する
- ・リストは、下記のテキストを使用する

●使用するテキスト

ごうか美術館

この世の中で一番美しい物を収集した展示会「GOUKA」

- ・ home
- ・ exhibition
- ・ access

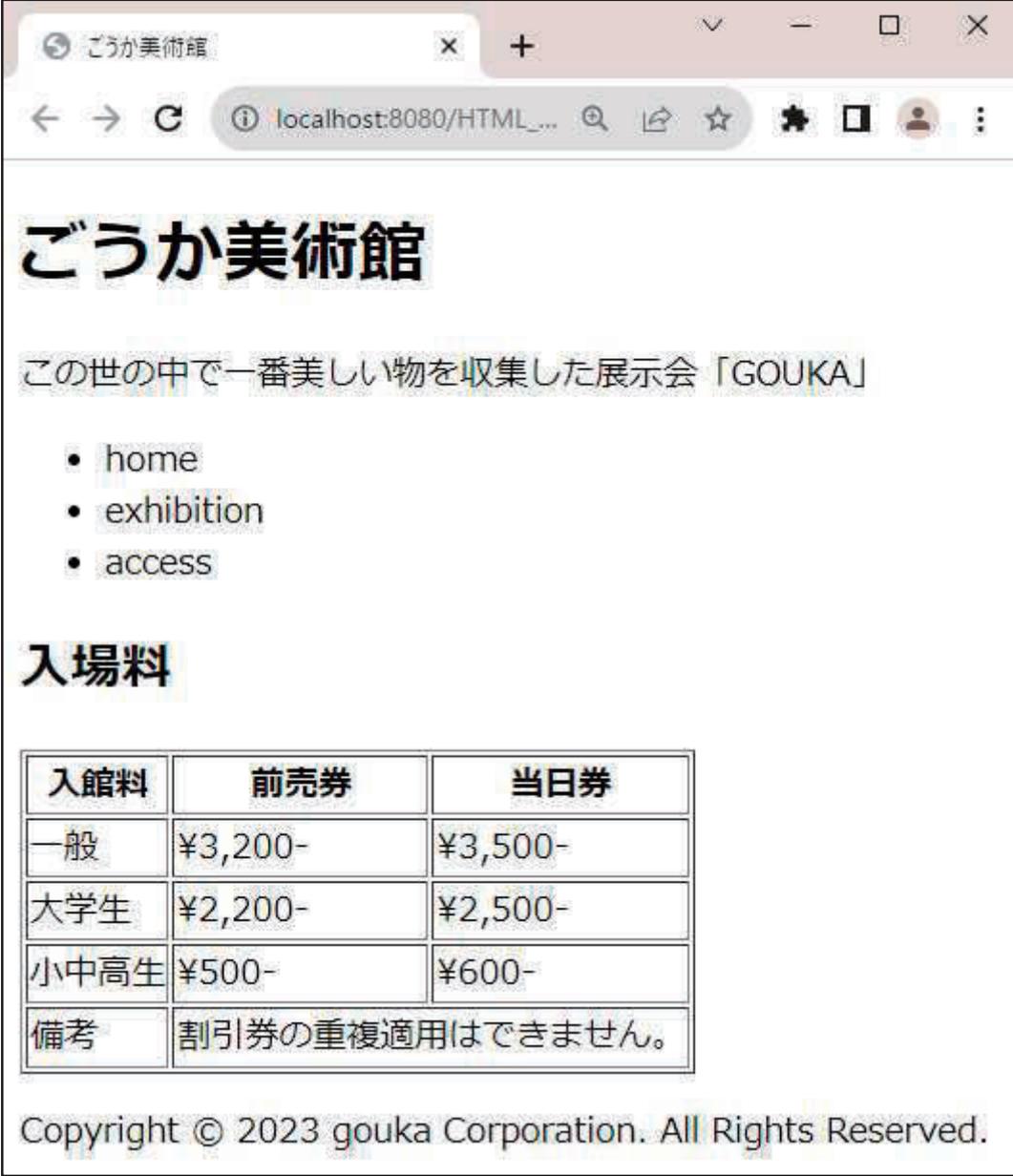
入場料

入館料	前売券	当日券
一般	¥3,200-	¥3,500-
大学生	¥2,200-	¥2,500-
小中高生	¥500-	¥600-
備考	割引券の重複適用はできません。	

Copyright © 2023 gouka Corporation. All Rights Reserved.

問題9 続き

[実行結果]



ごうか美術館

この世の中で一番美しい物を収集した展示会「GOUKA」

- [home](#)
- [exhibition](#)
- [access](#)

入場料

入館料	前売券	当日券
一般	¥3,200-	¥3,500-
大学生	¥2,200-	¥2,500-
小中高生	¥500-	¥600-
備考	割引券の重複適用はできません。	

Copyright © 2023 gouka Corporation. All Rights Reserved.

問題10 (難易度 ☆☆☆☆☆)

- ・ある文具店で行われる、懸賞の応募ページを作成する
- ・HTMLプログラムは、「webapp」フォルダの「exercise10.html」ファイルに保存する
- ・プログラム作成後に、「exercise10.html」ファイルを実行し、Webブラウザに表示される実行結果を確認する

[要件]

- ・「exercise10.html」ファイル内にHTMLの基本構文を記述し、その中に見出し要素と段落要素とヘッダー要素とメイン要素、フッター要素とセクショニングコンテンツのアーティクル要素、ナビ要素とリスト要素、テーブル要素とフォーム要素を使用して作成する
- ・応募フォームを表形式でフォームを作成する、また、線の設定は、テーブル要素に「border」属性を使用し、「border="1"」を設定する
- ・フォーム要素の「action」属性は、ダミーとして、「Apply.php」とし、「method」属性は、「post」を設定する、オートコンプリート機能を「OFF」にする
- ・「名前」フォームは、入力最大文字数を「10」とし、ダミーテキストには、「名前を入力してください」とし、部品名は、「name」とする
- ・「メールアドレス」フォームは、必須項目とし入力せず送信した場合は、エラーを表示するダミーテキストには、「メールアドレスを入力してください」とし、部品名は、「mail」とする
- ・「性別」フォームの部品名は、「seibetu」、「年代」フォームの部品名は、「age」とする
- ・「懸賞コース」フォームは、プルダウンメニューとし選択項目として、「Aコース」「Bコース」「Cコース」を設定、部品名は、「course」とする
- ・それぞれのフォーム部品の値は、部品に設定されているテキストを設定する
- ・送信ボタンとリセットボタンは、「button」要素を使用する
- ・フォーム部品に「label」要素を設定し、テキストをクリックしてもフォーム部品を選択できるようにする
- ・「label」要素の設定は、要素で囲む方法とid属性で関連付ける方法を使用する
- ・Webページの言語は「日本語」を指定、エンコードは「UTF-8」とする
- ・文字化けの恐れがある記号については、特殊記号を使用する
- ・リストは、下記のテキストをする

問題 10 続き

- 使用するテキストとフォームのレイアウト

ごおか文具店

- home
- item
- access

応募フォーム

名前	:	<input type="text" value="名前を入力してください"/>
メールアドレス	:	<input type="text" value="メールアドレスを入力してください"/>
性別	:	<input type="radio"/> 男性 <input type="radio"/> 女性
年代	:	<input type="radio"/> 20代 <input type="radio"/> 30代 <input type="radio"/> 40代 <input type="radio"/> 50代以上
懸賞コース	:	<input type="text" value="Aコース v"/>
お問合せ内容	<input type="text"/>	
<input type="button" value="送信"/>		<input type="button" value="リセット"/>

Copyright © 2023 gouka Corporation. All Rights Reserved.

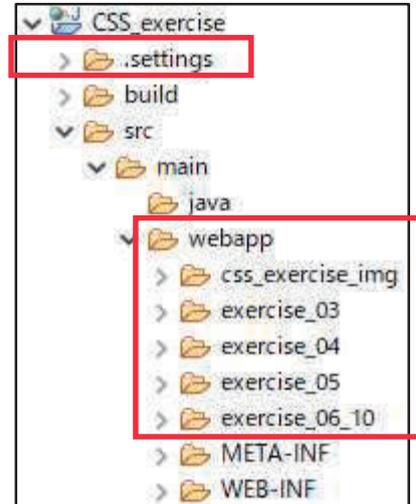
[実行結果]

The screenshot shows a web browser window with the URL localhost:8080/HTML_exercise/exercise10... The page content is identical to the design above, showing the navigation menu, the '応募フォーム' (Application Form) with input fields for name, email, gender, age, prize course, and a message box, and '送信' (Send) and 'リセット' (Reset) buttons. The copyright notice 'Copyright © 2023 gouka Corporation. All Rights Reserved.' is visible at the bottom.

第4章 プログラミング演習

事前準備

- ①Eclipse上で動的Webプロジェクトを作成する
プロジェクト名 : CSS_exercise
- ②配布資料の「演習」フォルダの中のデータをコピーする
画像フォルダ名 : css_exercise_imgフォルダ
フォルダ名 : exercise_03
フォルダ名 : exercise_04
フォルダ名 : exercise_05
フォルダ名 : exercise_06_10
- ③配付データを貼り付ける
保存先 : webapp



次の問題1～問題10を読んで、提示された[要件]と[実行結果]を満たすプログラムを作成しなさい。

問題1 (難易度 ☆)

- ・色の話のページを作成する
- ・HTMLプログラム・CSSのプログラムは、「webapp」フォルダの「exercise_01.html」ファイルに保存する
- ・プログラム作成後に、「exercise_01.html」ファイルを実行し、Webブラウザに表示される実行結果を確認する

[要件]

- ・「exercise_01.html」ファイル内のhead要素内にスタイル要素を使用する方法とHTML要素に直接CSSを記述する方法を使用し、CSSを入力する
- ・各要素の文字の大きさ、文字の色の設定と見出し1要素にテキストシャドウの設定と各段落テキストに背景色を設定する
- ・見出し2の文字の太さを「ノーマル」に設定し、細くする
- ・それぞれの文章の揃え位置を「青」の文章は左揃え、「赤」の文章は中央揃え、「緑」の文章は、右揃えに設定する
- ・ページ全体に文字の色(#808080)を設定し、文字の色を継承できるようにする
- ・ページ全体に背景色を設定する

[実行結果]



参考サイト ウィキペディア：<https://ja.wikipedia.org/wiki/>

```

01 <!DOCTYPE html>
02 <html>
03 <head>
04 <meta charset="UTF-8">
05 <title>色の話</title>
06 <[ ]>
07   body{
08     [ ]: #808080;
09     [ ]: #ebf6f7;
10   }
11   [ ]{
12     [ ]: 28px;
13     color: rgba(230, 180, 34, 1);
14     [ ]: 1px 2px 2px #808080;
15   }
16   h2{
17     [ ]: 18px;
18     font-[ ]: normal;
19   }
20   [ ]{
21     [ ]: 14px;
22     [ ]-color: #fdeff2;
23   }
24 }
25 </[ ]>
26 </head>
27 <body>
28   <h1>色の話</h1>
29   <h2 [ ]=" [ ]: #165e83;">色名としての青</h2>
30   <p [ ]=" [ ]: left;">青という基本色名は、その他多くの固有色名を総称として含んでいる。
31   たとえば、<br>
32   水色（みずいろ）・空色（そらいろ）と呼ばれるような明度が高く彩度の低い、淡い色合いのもの、
33   <br>
34   紺色（こんいろ）や藍色（あいいろ）、群青色（ぐんじょういろ）などの明度が低い、濃い色合いのもの
35   などが<br>
36   青に含まれる。</p>
37   <h2 [ ]=" [ ]: #ba2636 ;">基本色名としての赤</h2>
38   <p [ ]=" [ ]: center;">丹（タン）が色を名指すときは赭土（シャド）、赤土の色の意味である。
39   <br>
40   朱（シュ）は、硫化水銀による赤色顔料辰砂の意味を持つ。オレンジがかった赤。緋（ヒ）は、<br>
41   濃く明るい赤色を指す。紅（コウ）は、わずかに紫がかった赤を指す。</p>
42   <h2 [ ]=" [ ]: #38b48b ;">緑という色名</h2>
43   <p [ ]="text-align: [ ];">緑に相当する色はかなり広範に及ぶ色の総称であるが、<br>
44   それぞれの色には多く「柳色」や「モスグリーン」などの固有の色名が付いている。<br>
45   緑はまた漢字で碧や翠とも表記されるが、この場合やや青みの強い色を表すことが多く、<br>
46   比較的藍緑色に近い色合いである。</p>
47 </body>
48 </html>

```

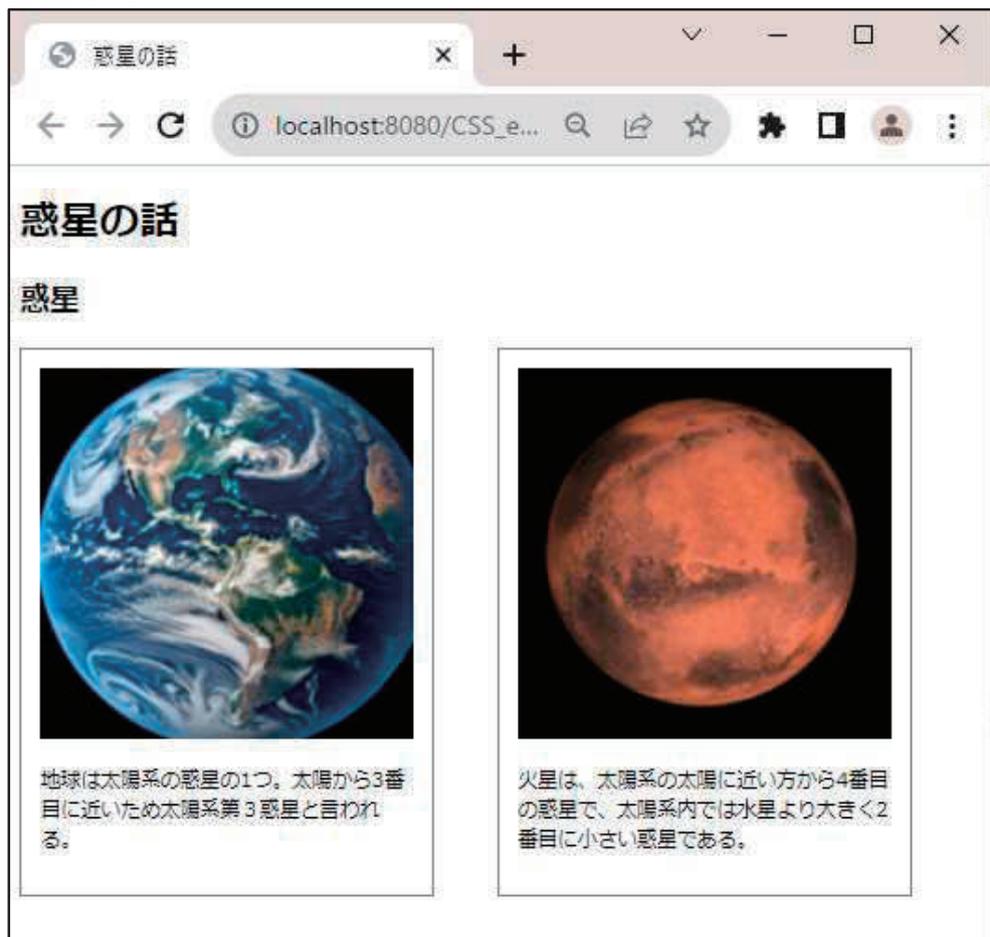
問題2 (難易度 ☆)

- ・惑星の話のページを作成する
- ・HTMLプログラム・CSSのプログラムは、「webapp」フォルダの「exercise_02.html」ファイルに保存する
- ・プログラム作成後に、「exercise_02.html」ファイルを実行し、Webブラウザに表示される実行結果を確認する

[要件]

- ・「exercise_02.html」ファイル内のhead要素内にスタイル要素を使用する方法とHTML要素に直接CSSを記述する方法を使用し、CSSを入力する
- ・CSSを使用して、画像要素に幅「100%」と高さ「auto」を設定し、画像の大きさが親要素の幅に追従するように設定する
※幅だけの設定にすると画像の縦横の比率が崩れてしまうので高さ「auto」を設定する
- ・CSSを使用して、図表要素に元からついている余白をゼロに設定する
- ・CSSを使用して、画像と文章を1つのブロックボックスとして囲っている要素に幅「300px」、線「1px」、線の色「黒 (#808080)」、線の内側に余白「15px」、隣のブロックボックスとの間に外側の余白「50px」を、クラス「sizes」を使用して設定する。
ただし、線の設定には、ショートハンドを使用する
- ・CSSを使用して、ブロックボックスを囲っている親要素にフレックスレイアウトを設定し横並びにする

[実行結果]



参考サイト ウィキペディア : <https://ja.wikipedia.org/wiki/>

```

01 <!DOCTYPE html>
02 <html lang="ja">
03 <head>
04   <meta charset="UTF-8">
05   <title>惑星の話</title>
06   <style>
07     img {
08       [ ]: 100%;
09       height: auto;
10     }
11     figure {
12       margin: 0;
13     }
14     .sizes {
15       [ ]: 300px;
16       padding: 15px;
17       [ ]: 50px;
18       border: [ ];
19     }
20   </style>
21 </head>
22 <body>
23   <header>
24     <h1>惑星の話</h1>
25   </header>
26   <main>
27     <article>
28       <h2>惑星</h2>
29       <[ ] style="display: [ ]; [ ]: row;">
30         <div [ ]>
31           <figure>
32             
33           </figure>
34           <p>地球は太陽系の惑星の1つ。太陽から3番目に近いので太陽系第3惑星と言われる。</p>
35         </div>
36         <div [ ]>
37           <figure>
38             
39           </figure>
40           <p>火星は、太陽系の太陽に近い方から4番目の惑星で、太陽系内では水星より大きく2番目に小さい惑星である。</p>
41         </div>
42       </[ ]>
43     </article>
44   </main>
45 </body>
46 </html>

```

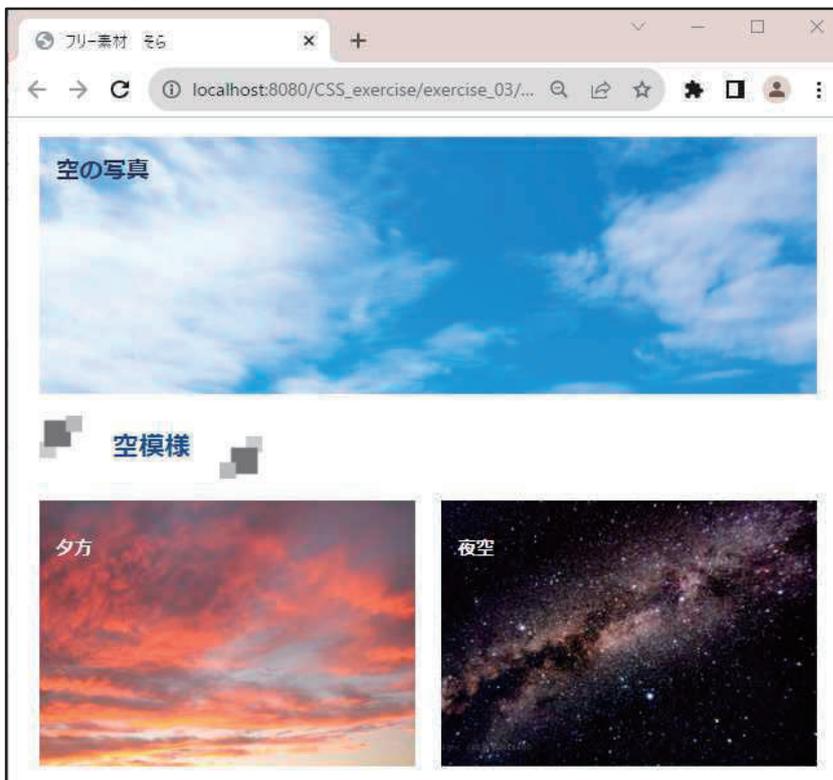
問題3 (難易度 ☆☆)

- ・フリー素材の空の写真を紹介するページを作成する
- ・HTMLプログラムは、「webapp」フォルダの「index.html」ファイルに保存する
- ・CSSプログラムは、「webapp」フォルダの「exercise_03.css」ファイルに保存する
- ・プログラム作成後に、「index.html」ファイルを実行し、Webブラウザに表示される実行結果を確認する

[要件]

- ・「exercise_03」フォルダの中の「index.html」ファイルの6行目に外部ファイルの「exercise_03.css」を読み込む設定をする
- ・「exercise_03.css」に「index.html」に記述されているHTMLプログラムに対してCSSを設定する
- ・「exercise_03.css」にクラス「wrapper」を使用して、Webページ全体のサイズと中央配置の設定をする。幅は「900px」、文字の大きさは「18px」、文字の色は「白」を設定する
- ・画像は、「images」フォルダに保存されている画像を使用する。
- ・CSSを使用して、ヘッダーの領域に背景画像(aozora.jpg)を設定し、領域からはみ出しても良いので縦横の比率を保ったまま、領域いっぱいに表示する
- ・CSSを使用して、見出し2の背景に左上と右下に画像を配置し、見出しのデザインを完成させる。左上に使用する画像は「css_ensyu-3-1.jpg」、右下に使用する画像は「css_ensyu-3-2.jpg」とする
- ・CSSを使用して、見出し3が記述されているブロックボックス2つにクラス「content」を適用し、同じ形式にする、ブロックボックスの背景に疑似クラスを使用して、画像を設定する。左の背景画像は「yuugata.jpg」、右の背景画像は「yoruzora.jpg」とする。背景画像を縦横の比率を保ったまま、背景画像を見切れないように表示する
- ・CSSを使用して、2つのブロックボックスの親要素にグリッドレイアウトを設定し、横並びにし、グリッドの間の余白を「30px」とする

[実行結果]



HTML/CSS

index.html

```
:  
05 <title>フリー素材 そら</title>  
06 <[ ] rel="stylesheet" [ ] type="text/css">  
07 </head>
```

exercise_03.css

```
01 @charset "UTF-8";  
02 .wrapper{  
03 [ ]  
04 [ ]  
05     font-size: 18px;  
06     color: #fff;  
07 }  
08 /*-----ヘッダーの設定*/  
09 header{  
10     width: 100%;  
11     height: 300px;  
12 [ ]  
13 [ ]  
14 }  
15 header h1{  
16     padding: 20px;  
17     color: #192f60;  
18 }  
19 h2{  
20     font-size: 30px;  
21     text-align: center;  
22     color: #19448e ;  
23     width: 200px;  
24     padding: 15px 30px;  
25 [ ]  
26 [ ]  
27 [ ]: left top, right bottom;  
28 }  
29 .container{  
30     display:grid;  
31 [ ]:1fr 1fr;  
32     gap: 30px;  
33 }  
34 .content{  
35     width: 435px;  
36     height: 400px;  
37     padding: 20px;  
38     box-sizing: border-box;  
39 [ ]  
40 [ ]  
41 }  
42 . [ ]
```

exercise_03.css (続き)

```
43     background-image: url(../images/yuugata.jpg);
44 }
45 .
46     background-image: url(../images/yoruzora.jpg);
47 }
48 /*-----ヘッダーの設定*/
```

問題4（難易度 ☆☆）

- ・おいしいパン屋さんの割引カレンダーのページを作成する
- ・CSSプログラムは、「exercise_04」フォルダの、「CSS」フォルダの「exercise_04.css」ファイル保存する
- ・プログラム作成後に、exercise_04フォルダの「index.html」ファイルを実行し、Webブラウザに表示される実行結果を確認する

[要件]

- ・「exercise_04.css」ファイルにexercise_04フォルダの「index.html」に記述されているHTMLプログラムに対してCSSを設定する
- ・CSSを使用して、表の外枠に線を設定と線を重ねて表示する設定を行い、線を1本で表示する
- ・CSSを使用して、表のキャプション「4月」の揃え位置を「左」、文字の大きさを「30px」、文字の太さを「bold」に設定する
- ・CSSを使用して、表の列の幅は「150px」、高さ「150px」に設定する
- ・CSSを使用して、表見出しの行に背景色「#808080」、内側の余白を上下「10px」、左右「20px」、列の右側の線を幅「1px」、線種は「実線」、線の色は「白」を設定するただし、線の設定はショートハンドを使用する
- ・CSSを使用して、表見出しの列幅に線と余白を含む設定にし、幅が「150px」より大きくならないようにする
- ・CSSを使用して、表見出しの一番右側の最後の列に、右側の線が表の外形線とかぶるので線の設定をなしにする
- ・CSSを使用して、各列に対して、文字の大きさ「35px」、列の垂直方向の位置を「上」、文字の太さを「bold」、線を幅「1px」、線種は「実線」、線の色は「#808080」を設定するただし、線の設定はショートハンドを使用する
- ・疑似クラスを使用して、「日曜日」の列の文字の色を「赤色（#e83929）」、「土曜日」の列の文字の色を「青色（#007bbb）」に設定する
- ・CSSを使用して、割引を行う日（22日）に疑似クラスを使用して、赤丸「5」を設定し、割引の日程をカレンダーに表示する
- ・CSSを使用して、赤丸「5」は、疑似要素で追加し、ブロックボックスに変更して加工する幅高さを「100px」背景の色は「#e83929」角丸は「100px」文字の色は「白」赤丸の中にテキスト「5」を入力、赤丸の左外側の余白を「25px」に設定する
- ・CSSを使用して、カレンダーの前月の日付、次の月の日付は、文字の色を「#e3e3e3」を設定し薄い色で表示する

[実行結果]



Insert title here x +

localhost:8080/CSS_exercise/exercise_04/index.html

おいしいケーキ屋

毎月ランダムに割引を実施。割引日・定休日を下記のカレンダーからご確認ください。

割引カレンダー

4月

日	月	火	水	木	金	土
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	1	2	3	4	5	6

※赤丸5%が5%割引の日になります。

```

01 @charset "UTF-8";
02 /*-----tableの設定*/
03 table{
04     
05 }
06 }
07 table caption{
08     text-align: left;
09     font-size: 30px;
10     font-weight: bold;
11 }
12 table thead tr th{
13     
14 }
15 }
16 }
17 }
18 }
19 }
20 table thead tr th: 
21     border-right:none;
22 }
23 table tbody tr:first-of-type td:nth-of-type(n):nth-of-type(-n+6){
24     color: #e3e3e3 ; }
25 table tbody tr:last-of-type td:nth-of-type(n+2):nth-of-type(-n+7){
26     color: #e3e3e3 ; }
27 table tbody tr td{
28     height: 150px;
29     
30 }
31 }
32 }
33 }
34 table tbody tr td:  {
35     color: #e83929 ;
36 }
37 table tbody tr td:  {
38     color: #007bbb ;
39 }
40 table tbody tr:nth-of-type(4) td:nth-of-type(7)::after{
41      : " 5";
42      : block;
43     width: 100px;
44     height: 100px;
45     background-color: #e83929;
46     box-sizing: border-box;
47     padding: 23px 20px 17px 5px;
48     
49     color: #fff;

```

HTML/CSS

```
50     margin-left: 25px;  
51 }/*-----tableの設定*/
```

問題5 (難易度 ☆☆☆)

- ・ 駄菓子屋本舗のアンケートフォームを作成する
- ・ HTMLプログラムは、「exercise_05」フォルダの、「index.html」ファイルに保存する
- ・ CSSプログラムは、「exercise_05」フォルダの、「CSS」フォルダの「exercise_05.css」ファイル保存する
- ・ プログラム作成後に、「index.html」ファイルを実行し、Webブラウザに表示される実行結果を確認する

[要件]

- ・ exercise_05フォルダの「index.html」ファイルに記述されているHTMLプログラムの中に、コンテンツ全体を囲むブロックボックスの要素を追加する
- ・ 「exercise_05.css」ファイルにexercise_05フォルダの「index.html」に記述されているHTMLプログラムに対してCSSを設定する
- ・ 「index.html」ファイルに追加したブロックボックスの要素に対して、「exercise_05.css」のファイルにクラス「wrapper」を使用して、幅「850px」と中央配置の設定をする
- ・ CSSを使用して、表の外枠に線の設定と、線を重ねて表示する設定をする
- ・ CSSを使用して、表の偶数行に疑似クラスを使用して、背景色「#e3e3e3」を設定する
- ・ CSSを使用して、表の列に内側の余白を上下「15px」、左右「20px」、列の垂直方向の位置を「中央」に設定する
- ・ CSSを使用して、フォーム部品の「送信」ボタンと「リセット」ボタンを列の中央揃えに設定、ボタンの内側の余白を上下「5px」、左右「ゼロ」を設定して、ボタンを大きく表示する
- ・ CSSを使用して、フォーム備品の「メールアドレス」の入力欄の幅を「250px」と内側の余白上下「5px」、左右「ゼロ」に設定する
- ・ CSSを使用して、各フォーム部品にフォーカスされた時の部品の線色を「青色 (#1e50a2)」に設定する
- ・ CSSを使用して、「送信」と「リセット」のボタンがフォーカスされた時のボタンの線色を「赤色 (#b7282e)」に設定する
- ・ CSSを使用して、コピーライトの表示位置を中央揃えにする

[実行結果]



駄菓子屋本舗

- home
- item
- access

アンケート

メールアドレス :

性別 : 男性 女性

年代 : 20代 30代 40代

好きな駄菓子 : いちご飴 棒きなこ 駄菓子 ポップライス きびだんご

お問合せ内容

Copyright © 2023 dagashiyahonpo Corporation. All Rights Reserved.

HTML/CSS

index.html

```
:
08 <body>
09   [redacted]
10     <header>
11       <h1>駄菓子屋本舗</h1>
12     [redacted]
13   </footer>
67 </body>
68 </html>
69
70
```

exercise_05.css

```
01 @charset "UTF-8";
02 .wrapper{
03   [redacted]
04 }
05
06 table{
07   [redacted]
08 }
09
10 [redacted], [redacted]{
11   background-color:#e3e3e3 ;
12 }
13 table tr td{
14   [redacted]
15 }
16
17 table tfoot tr{
18   text-align: center;
19 }
20 [redacted] input{
21   width: 250px;
22   padding: 5px 0;
23 }
24 table tfoot tr td button{
25   width: 150px;
26   padding: 5px 0;
27 }
28 [redacted]{
29   outline:solid 3px #1e50a2;
30 }
31 [redacted] #b7282e;
32 }
33
34 footer p{
35   text-align: center;
36 }
```

問題6 (難易度 ☆☆☆)

- ・「GOUKA COFFEE」カフェ屋さんのWebページ(ヘッダー部分)を作成する
- ・exercise_06_10フォルダの「index.html」ファイル内の5行目から18行目までのHTMLプログラムを対象に、HTMLプログラムとCSSプログラムを記述してカフェ屋さんのヘッダー部分を作成する
- ・CSSフォルダの「style.css」ファイル内の32行目からCSSプログラムを記述する
- ・HTMLプログラムは、「exercise_06_10」フォルダの、「index.html」ファイルに保存する
- ・CSSプログラムは、「exercise_06_10」フォルダの、「css」フォルダの、「style.css」ファイルに保存する
- ・プログラム作成後に、「index.html」ファイルを実行し、Webブラウザに表示される実行果を確認する

[要件]

- ・「index.html」ファイル内のhead要素の間に外部ファイルの「style.css」をリンクする設定、Webページ全体を囲うブロックボックスの要素を追加する
追加した、ブロックボックスの要素にCSSでクラス属性を使用して、クラス名「wrapper」を記述する
見出し1とリストを囲うブロックボックスの要素を追加し、クラス属性を使用して、クラス名「header_area」を記述する
- ・「style.css」ファイルに記述する際は、コメントタグで「exercise06」と記述されている32行目からCSSプログラムを記述する
- ・クラス名「wrapper」に幅「900px」とコンテンツを中央配置に設定する
- ・36行目にコメントタグを使用し、「/*-----ヘッダーの設定*/」を記述する
- ・クラス名「header_area」に見出し1とリストを横並びにするフレックスレイアウトを設定する
その際に見出し1とリストを下揃えと両端に揃える設定をする
上と下にそれぞれ線を表示する上の線は太さ「7px」、色は「#320c00」、下の線は太さ「3px」、色は「#320c00」と同じ色を設定する
- ・画像は、「images」フォルダに保存されている物を使用する。
- ・クラス「header_area」の見出し1の左側にカフェのロゴ(cafе_rogо.gif)を背景画像として表示させる
見出し1とロゴが重ならないようにと水平方向で揃うように内側の余白を上「20px」右「ゼロ」下「20px」左「100px」に設定する
上下の線との間の余白を「10px」設定し、背景画像として設定したロゴを左端から離すために背景画像の位置を左から「10px」に設定する
- ・クラス「header_area」の見出し1は、Webフォントを使用する。フォントのデータは「Web-font」フォルダの中にある「Oswald-Medium」を適用する。「style.css」の9行目と10行目にフォントデータを読み込むプログラムを記述する
- ・CSSを使用して、リスト要素を垂直方向から水平方向に並べるためフレックスレイアウトを使用する
その際、下側の線からリストを離すために下から「10px」と、右側からも離すため「10px」の外側の余白を設定する
- ・クラス「header_area」のリスト項目に対して、Webフォント「FiraSans-Medium」を設定する
- ・クラス「header_area」のリストのリンクの設定に対して、文字の色を「#3d3d3d」下線を表示しない設定をする
リンク要素をインラインボックスからブロックボックスに変更し、内側の余白を上下「5px」左右「10px」を設定する
リンクに疑似クラスを使用し、カーソルが重なった時に背景の色を「#3d3d3d」に変わるように設定する

問題 06 続き

[実行結果]



index.html

```

:
05 <title>GOUKA COFFEE</title>

```

06

07 </head>

08 <body>

09

10 <header>

11

12 <h1>GOUKA COFFEE</h1>

:

20

21 </nav>

22

:

116 </footer>

117

118 </body>

119 </html>

:

```
:
32 .wrapper{
33     
34 }
35 }
36 /*-----ヘッダーの設定*/
37 .header_area{
38     
39
40
41     align-items: flex-end;
42     border-top: solid 7px #320c00;
43     border-bottom: solid 3px #320c00;
44 }
45 .header_area h1{
46     
47
48
49
50     padding: 20px 0px 20px 100px;
51     margin: 10px 0px;
52 }
53 .header_area ul{
54     list-style: none;
55     display: flex;
56     flex-direction: row;
57     margin-bottom: 10px;
58     margin-right: 10px;
59 }
60 .header_area ul li{
61     
62 }
63 .header_area ul li a{
64     color: #3d3d3d;
65     
66
67     padding: 5px 10px;
68 }
69 .header_area ul li a:  {
70     background-color: #eeeeee ;
71 }
```

問題7 (難易度 ☆☆☆)

- ・「GOUKA COFFEE」カフェ屋さんのWebページ(ビジュアル部分)を作成する
- ・exercise_06_10フォルダ「index.html」ファイル内の23行目から27行目のHTMLプログラムを対象にHTMLプログラムとCSSプログラムを記述してカフェ屋さんのビジュアル部分を作成しなさい
- ・CSSフォルダの中の「style.css」ファイルの74行目からCSSプログラムを記述する
- ・HTMLプログラムは、「exercise_06_10」フォルダの、「index.html」ファイルに保存する
- ・CSSプログラムは、「exercise_06_10」フォルダの、「css」フォルダの、「style.css」ファイルに保存する
- ・プログラム作成後に、「index.html」ファイルを実行し、Webブラウザに表示される実行果を確認する

[要件]

- ・「index.html」ファイル内の23行目のメインビジュアルの画像とすぐ下の見出し2と段落を囲むブロックボックスを追加する
追加したブロックボックスにクラス属性を使用し、クラス「main_visual」を記述する
- ・クラス「main_visual」の中の見出し2と段落を囲むブロックボックスを追加する
追加したブロックボックスにクラス属性を使用して、クラス「visual_title」を記述する
- ・「style.css」ファイルに記述する際は、コメントタグで「exercise07」が記述されている74行目からCSSプログラムを記述する
- ・クラス「main_visual」に画像の下側に影をつけるCSSを設定する
影は、左右の移動距離は「ゼロ」、垂直方向の移動は「5px」、ぼかし「3px」、広がりを「-3p」を設定する
- ・この時、画像の下に影と画像との間に余白が表示されるので、余白を消すための設定として、CSSファイルの26行目「img」要素に垂直方向の位置を「top」に指定する
その際に幅「100%」と高さ「auto」を一緒に設定する
- ・クラス「visual_title」にブロックボックスの幅「350px」文字の色「#fff」の設定と背景色を半透明で使用したいので、関数記法の「rgba」を使用する
設定値は、R:50 G:12 B:0 A:0.7とする。
- ・CSSを使用して、テキストと画像を重ねて表示させたいので、CSSの「position」を使用して位置を指定する
その際、位置指定する際の基準をクラス「main_visual」に設定したブロックボックスを基準にするので、クラス「main_visual」に「position」の「絶対配置」を追加で設定する
- ・クラス「visual_title」に、「position」を「絶対配置」、「left」を「50px」、「bottom」を「30px」に設定する
- ・クラス「visual_title」の見出し2に文字の大きさ「27px」、外側の余白を上下「20px」、左右「50px」、文字のフォントをWebフォントの「NotoSansJP-ExtraLight」に設定する
- ・クラス「visual_title」の見出し2の2行目の文章をずらして表示するため、見出し2のspan要素で囲んだ部分にのみ適用する設定をする
2行目の文字にインラインボックスの要素で囲み、その要素をブロックボックスに変更し、左外側の余白を「50px」を設定する
- ・クラス「visual_title」の段落に文字の大きさ「15px」、揃え位置を「中央」、文字のフォントをWebフォントの「NotoSansJP-ExtraLight」に設定する
段落内の日本語と英語の文章をドット線で区切る
英語の文章をインラインボックスの要素で囲み、その要素の上側にドット線を太さ「2px」線の色「#fff」で設定する
- ・クラス「visual_title」の段落の中のインラインボックスの要素とドット線の間の内側の余白上だけ「10px」を設定する
また、ドット線の外側の余白を上「8px」、左右「ゼロ」、下「30px」を設定する
- ・111行目にコメントタグを使用して、「/*-----ヘッダーの設定*/」を記述する

問題 07 続き

[実行結果]



問題8 (難易度 ☆☆☆☆)

- ・「GOUKA COFFEE」カフェ屋さんのWebページ(コンセプト部分)を作成する
- ・exercise_06_10フォルダ「index.html」ファイル内の35行目から60行目のHTMLプログラムを対象にHTMLプログラムとCSSプログラムを記述してカフェ屋さんのコンセプト部分を作成する
- ・CSSフォルダの中の「style.css」ファイルの114行目からCSSプログラムを記述する
- ・HTMLプログラムは、「exercise_06_10」フォルダの、「index.html」ファイルに保存する
- ・CSSプログラムは、「exercise_06_10」フォルダの、「css」フォルダの、「style.css」ファイルに保存する
- ・プログラム作成後に、「index.html」ファイルを実行し、Webブラウザに表示される実行果を確認する

【要件】

- ・「index.html」ファイル内の見出し2「concept」の次36行目にコンセプトの3つの項目を囲むブロックボックスの要素を追加する
追加したブロックボックスにクラス属性を使用してクラス「contents_upward」を記述する
- ・「concept」の各項目の見出し3と段落を囲むブロックボックスの各項目に追加する
追加したブロックボックスにクラス属性を使用して、クラス「text_outer」を各項目に記述する
- ・114行目にコメントタグを使用して、「/*-----コンテンツの設定*/」記述する
- ・CSSを使用して、見出し2「concept」とビジュアルの画像の間を広げたいので、ヘッダー要素の下方向に余白「50px」を設定する
- ・CSSを使用して、見出し2「concept」にWebフォントの「FiraSans-Medium」を設定、見出しのデザインとして、見出し下側に線を太さ「2px」、背景の色を「#320c00」に設定する
ただし、線の設定には、ショートハンドを使用する
見出しと線の間の内側の余白として「5px」、線の外側「concept」の1つ目の項目との間の外側の余白「75px」を設定する
- ・クラス「contents_upward」に幅「100%」を設定する
- ・CSSを使用してクラス「contents_upward」の「section」要素に、クラス「text_outer」を適用したブロックボックスと画像を横並びにするため、フレックスレイアウトを設定する
また、「section」要素の幅「700px」、中央配置の設定と、次との項目の間の外側の下の余白を「75px」に設定する
- ・クラス「text_outer」に幅「75%」、右側にある画像との間、右外側の余白「25px」を設定する
- ・CSSを使用してクラス「text_outer」の見出し3に文字の大きさ「16px」と段落との間に線を幅「1px」、線の色「#320c00」を設定する。また、見出し3の初期値の外側の余白をリセットするための外側の余白「ゼロ」を設定する
- ・クラス「text_outer」の段落に文字の大きさ「13px」を設定する
- ・CSSを使用して、2つ目の項目に背景色の設定とブロックボックスと画像の背景に色を設定するために、「section」要素を囲む、ブロックボックスの要素を追加し、クラス「band」を記述する
- ・クラス「band」に内側の余白「20px」と背景色に関数記法を使用して、「R:52 G:12 B:0 A:0.1」を設定し、次の項目との間、外側の余白「75px」を設定する
- ・クラス「band」の「section」要素に内側の余白「20px」を設定する
また、2つ目の項目のクラス「section」に設定している下側の余白「75px」を打ち消すための設定をクラス「band」の「section」要素に外側の余白「ゼロ」を設定する
- ・CSSを使用して、2つ目の項目だけ、画像とブロックボックスの並びをフレックスレイアウトの並び順の設定を使用して、逆並びにする
また、逆並びにするとクラス「text_outer」の設定で右外側の余白「25px」が反対側に必要になるので、クラス「band」の「section」要素のクラス「text_outer」に対して、再度、上「ゼロ」、右「ゼロ」、下「ゼロ」、左「25px」を設定する

問題 08 続き

[実行結果]

GOUKA COFFEE

Your own space to relax
Enjoy the finest coffee

concept

極上のコーヒー

ブルーマウンテンをベースに、トータル3種類の珈琲豆をブレンドしています。芳醇な香りや上品でスッキリした苦味や甘味が本来の特徴ですが、その特徴を最大限に活かし、他の2種類の珈琲豆も、そのものの特性を活かした焙煎を行い、アフターミックスでブレンドを行っています。ブルーマウンテンを用いた珈琲豆としては、あまりお目にかかれない、しっかりした飲みごたえを感じて頂ける商品になっています。



ゆったり過ごすランチ

安心、健康なお食事をご提供したいため、食材はできる限りオーガニックのものを使用しております。奥味の味をそのままお楽しみいただければと思います。どんな時であっても、どなたと食事しても、ほっと心に響かなければとるような、目で楽しみ、口で味わい、体で感じる料理を味わっていただきたいという思いを込めております。楽しいランチタイムをゆったりとおくつろぎください。



リラックスできる自分だけの空間

ブルーマウンテンをベースに、トータル3種類の珈琲豆をブレンドしています。芳醇な香りや上品でスッキリした苦味や甘味が本来の特徴ですが、その特徴を最大限に活かし、他の2種類の珈琲豆も、そのものの特性を活かした焙煎を行い、アフターミックスでブレンドを行っています。ブルーマウンテンを用いた珈琲豆としては、あまりお目にかかれない、しっかりした飲みごたえを感じて頂ける商品になっています。



問題9（難易度 ☆☆☆☆）

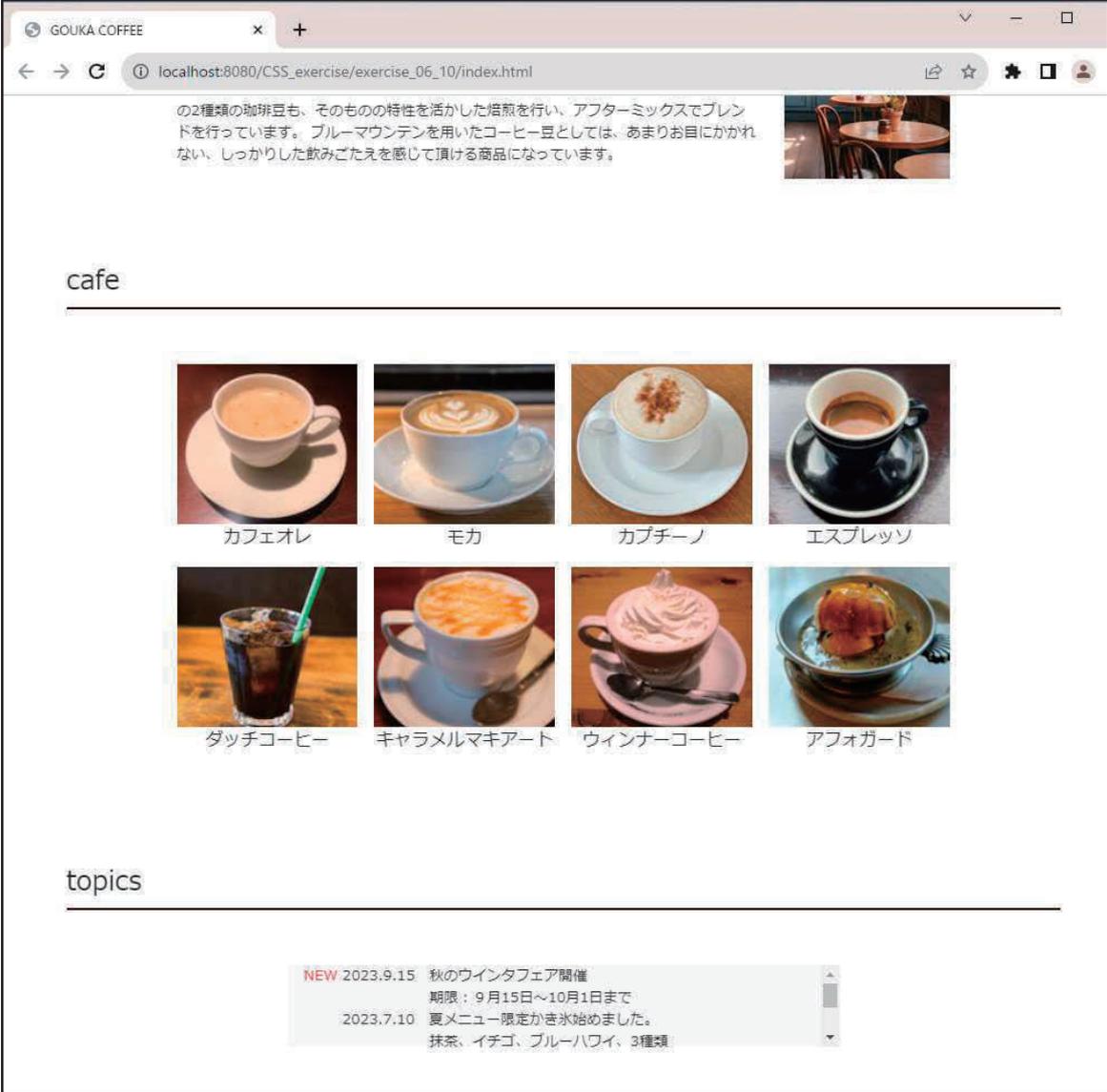
- ・「GOUKA COFFEE」カフェ屋さんのWebページ(カフェ・トピックス部分)を作成する
- ・exercise_06_10フォルダ「index.html」ファイル内の71行目から112行目のHTMLプログラムを対象にHTMLプログラムとCSSプログラムを記述してカフェ屋さんのコンセプト部分を作成しなさい
- ・CSSフォルダの中の「style.css」ファイルの161行目からCSSプログラムを記述する
- ・HTMLプログラムは、「exercise_06_10」フォルダの、「index.html」ファイルに保存する
- ・CSSプログラムは、「exercise_06_10」フォルダの、「css」フォルダの、「style.css」ファイルに保存する
- ・プログラム作成後に、「index.html」ファイルを実行し、Webブラウザに表示される実行果を確認する

[要件]

- ・「index.html」ファイル内の見出し2「cafe」の次72行目にコーヒーの画像8枚を囲むブロックボックスの要素を追加する
追加したブロックボックスにクラス属性を使用してクラス「under_contents」を記述する
- ・見出し2「topics」のリスト要素にクラス属性を使用して、クラス「topics」を記述する
- ・クラス「under_contents」に幅「700px」、外側の余白を上「ゼロ」、左右「auto」、下「100px」を設定し、8枚の画像をグリッドレイアウトで2行4列のレイアウトに設定する
列ごとの幅の指定は、親要素から見た子要素の大きさを割合で指定する
また、画像と画像の間の余白を「15px」に設定する
- ・画像のキャプションのテキスト位置を「中央揃え」にする
- ・CSSを使用してアーティクル要素の中の見出し2要素の「cafe」と「topics」に、線と画像の間の余白に下側「50px」を設定する
- ・クラス「topics」に幅「450px」、文字の大きさ「13px」、高さ「75px」、背景色「#f8f8f8」を設定し、枠からはみ出したテキストをスクロールして見られるようにする
- ・クラス「topics」にブロックボックスを中央配置に設定する
- ・疑似要素を使用してトピックスの一番新しい内容の頭に「NEW」を表示する
文字の色は「#e83929」、文章の頭にはみ出して表示させるので外側の余白を「-35px」に設定する
リスト項目の要素に外側の余白に「50px」を設定し、追加した「NEW」の文字とリスト項目がすべて表示できるようにする
- ・CSSを使用して各リスト項目を2行で表示させたいので、2行目のテキストにインラインボックスの要素を設定し、CSSを使用して、ブロックボックスに変換し、左の外側の余白を「79px」に設定し位置を調整する
- ・197行目にコメントタグを使用して、「/*-----トピックスの設定*/」を記述する

問題 09 続き

[実行結果]



の2種類の珈琲豆も、そのものの特性を活かした焙煎を行い、アフターミックスでブレンドを行っています。フルマウンテンを用いたコーヒー豆としては、あまりお目にかかれない、しっかりした飲みごたえを感じて頂ける商品になっています。



cafe



カフェオレ モカ カプチーノ エスプレッソ

ダッチコーヒー キャラメルマキアート ウィンナーコーヒー アフォガード

topics

- NEW** 2023.9.15 秋のウインタフェア開催
期限：9月15日～10月1日まで
- 2023.7.10 夏メニュー限定かき氷始めました。
抹茶、イチゴ、ブルーハワイ、3種類

問題10 (難易度 ☆☆☆)

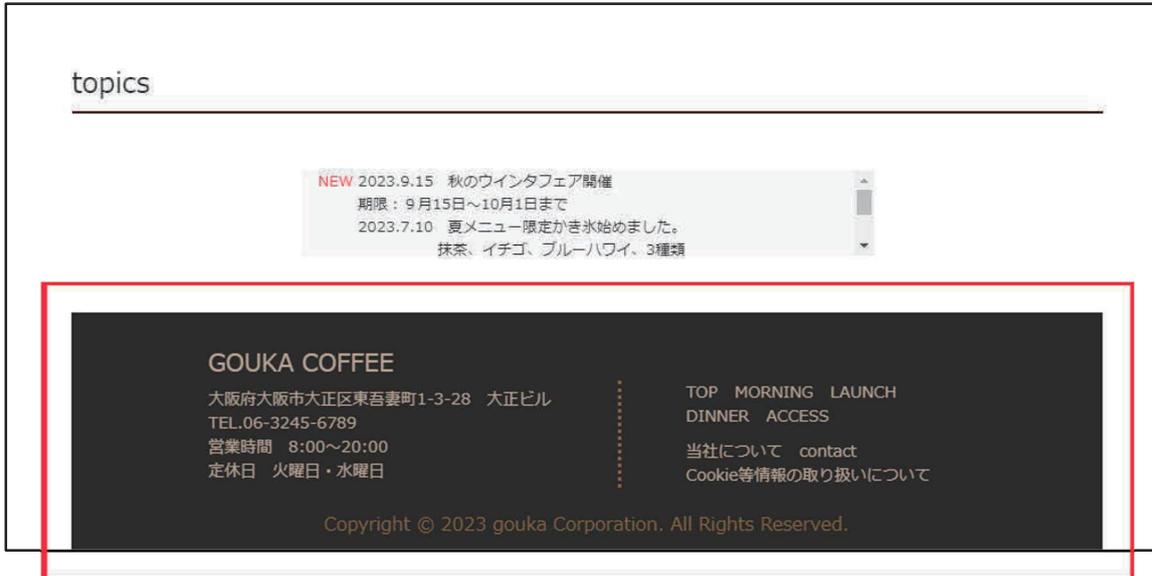
- ・「GOUKA COFFEE」カフェ屋さんのWebページ(フッター部分)を作成する
- ・exercise_06_10フォルダ「index.html」ファイル内の117行目から132行目のHTMLプログラムを対象にHTMLプログラムとCSSプログラムを記述してカフェ屋さんのコンセプト部分を作成する
- ・CSSフォルダの中の「style.css」ファイルの200行目からCSSプログラムを記述する
- ・HTMLプログラムは、「exercise_06_10」フォルダの、「index.html」ファイルに保存する
- ・CSSプログラムは、「exercise_06_10」フォルダの、「css」フォルダの、「style.css」ファイルに保存する
- ・プログラム作成後に、「index.html」ファイルを実行し、Webブラウザに表示される実行果を確認する

[要件]

- ・「index.html」ファイル内のフッター要素の開始タグの次 118 行目にフッターの見出し 2 と段落とリストを囲むブロックボックスの要素を追加する
追加したブロックボックスにクラス属性を使用してクラス「footer_content」を記述する
- ・さらに見出し 2 と段落を囲むブロックボックスの要素を追加する
追加したブロックボックスにクラス属性を使用してクラス「footer_left」を記述する
- ・フッターにある 2 つのリスト要素を囲むブロックボックスの要素を追加する
追加したブロックボックスにクラス属性を使用してクラス「footer_right」を記述する
- ・CSS を使用して、フッター要素に背景色「#303030」と文字の色「#c0ad98」を設定する
- ・クラス「footer_content」にクラス「footer_left」を記述したブロックボックスとクラス「footer_left」を記述したブロックボックスが横並びになるようにフレックスレイアウトを設定し、その際、垂直方向の位置を上揃えにする
また、幅「600px」、文字の大きさ「14px」、内の余白を上「25px」、外側の余白を上「50px」、左右「auto」、下「10px」に設定する。外側の余白の設定は、ショートハンドを使用する
- ・クラス「footer_left」に幅「300px」、右側余白「57px」を設定する
- ・クラス「footer_left」の見出し 2 に外側の余白にショートハンドを使用して、上下「5px」、左右「ゼロ」を設定する
- ・クラス「footer_left」の段落に外側の余白にショートハンドを使用して、上下「5px」、左右「ゼロ」を設定する
- ・クラス「footer_right」に幅「300px」、外側の余白を上「35px」、線の設定でドット線を指定し、線の太さ「3px」、線の色「#836543」を設定する
- ・クラス「footer_right」のリスト要素に、外側の余白「57px」を設定する
- ・2 つのリスト要素の間に項目を区切る余白を設定する、疑似クラスを使用して 2 つ目のリスト要素だけに、外側の余白を上「10px」に設定する
- ・クラス「footer_right」のリスト項目にフロートの「left」を設定し、リスト項目を横並びに表示する。ただし、クラス「footer_right」に設定した幅よりはみ出す場合は、折り返して表示する。また、内側の余白の右「15px」を設定する
- ・リスト項目にフロートを指定するとリスト要素の高さがなくなるので、「クリアフィックス」を各リスト要素にクラス属性を使って設定しリスト要素の高さを維持する
- ・クラス「footer_content」の段落「コピーライト」に CSS を使用して内側の余白「10px」と、テキストの位置は「中央揃え」、文字の色は「#836543」を設定する
段落要素の初期値の外側の余白をなしにするため、外側の余白に「ゼロ」を設定する

問題 10 続き

[実行結果]



令和5年度「専修学校による地域産業中核的人材養成事業」
IT分野DX人材養成のモデルプログラム開発と実証事業
フロントエンドエンジニア教材資料 1

令和6年2月

一般社団法人全国専門学校情報教育協会
〒164-0003 東京都中野区東中野 1-57-8 辻沢ビル 3F
電話：03-5332-5081 FAX 03-5332-5083

●本書の内容を無断で転記、掲載することは禁じます。