



オープンソースカンファレンス 2023 Hokkaido

HTML5プロフェッショナル認定試験 レベル1解説セミナー

～ 新卒入社からの歩み ～

2023年6月17日（土）@OSC 2023 Hokkaido

F() R K 株式会社フォーク
牛島 啓太



自己紹介



自己紹介

牛島 啓太

株式会社フォーク 札幌ブランチ所属
フロントエンドエンジニア

2020

株式会社フォークに新卒で入社
HTML5プロフェッショナル認定試験レベル1を取得



2023

レベル2取得に向けて勉強中



自己紹介

株式会社フォーク

@渋谷、札幌

(HTML5プロフェッショナル アカデミック認定校)

HTML5プロフェッショナル認定試験の合格者

Lv1: **74**名

※2023年5月時点

私が所属する札幌ブランチでは、
13名中9名がレベル1に合格しています



F()R K

<https://www.fork.co.jp/>

<https://4009.jp/>



本日解説する内容

- 試験概要、学習方法、自分の体験
- 試験のポイント
- 実務で役に立ったCSS
- 最後に



試験概要



HTML5プロフェッショナル認定資格とは

特定非営利活動法人エルピーアイジャパン（LPI-Japan）が、
HTML5、CSS3、JavaScriptなど
最新のマークアップに関する技術力と知識を、
公平かつ厳正に、中立的な立場で認定する認定制度



2つのレベル

今回はこちら



HTML5 レベル1

HTML5、CSS3などの最新のマークアップを使ってマルチデバイスに対応したコンテンツをデザイン・制作できる

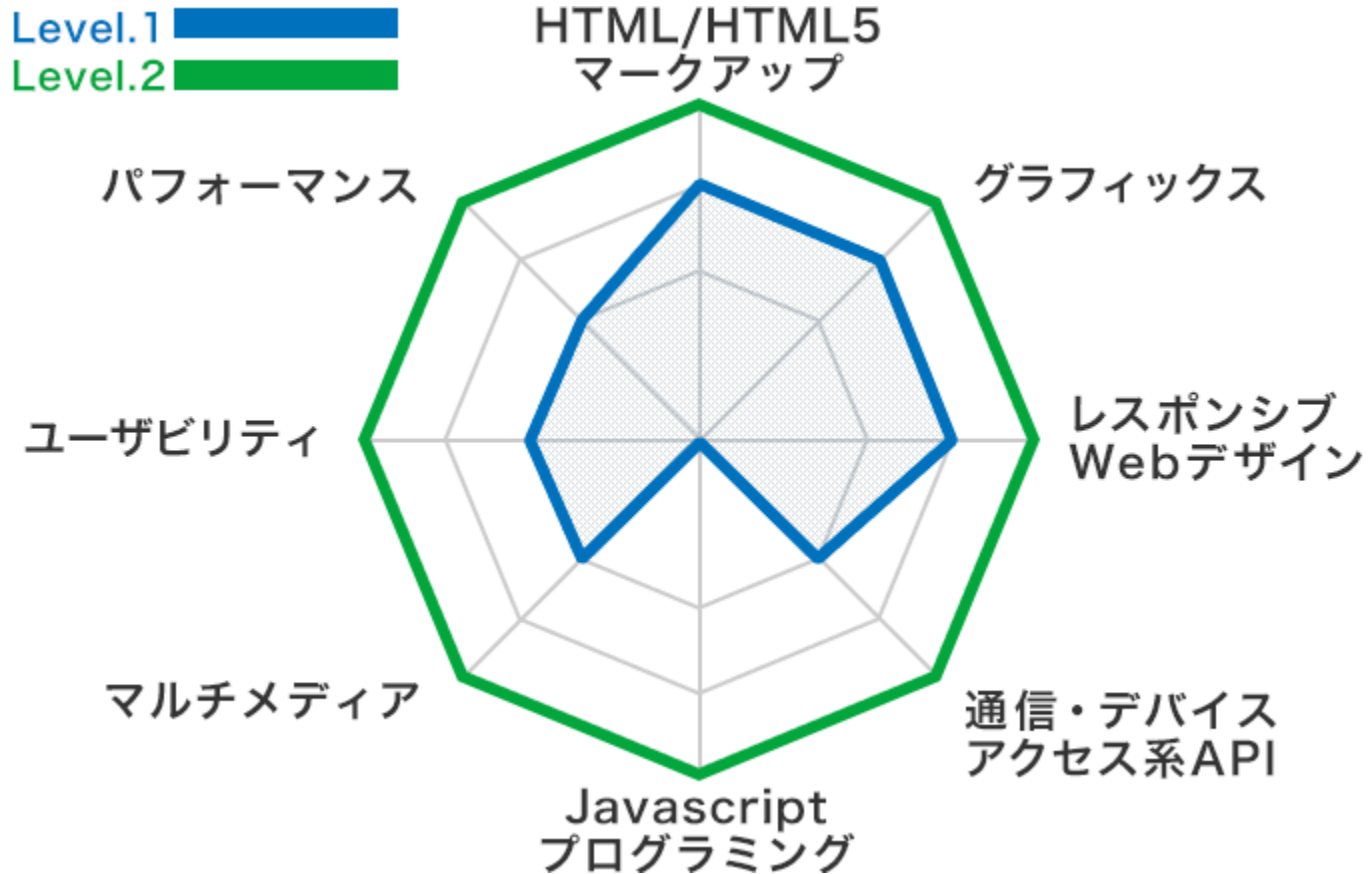


HTML5 レベル2

JavaScriptなどの最新のマークアップを使ってシステム間連携や最新のマルチメディア技術に対応したWebアプリケーションや動的Webコンテンツの開発・設計ができる。



2つのレベルと得られるスキル





スキル

HTML/HTML5

マークアップHTML5に関するタグの用途、構造の組み立て方に関する技術

グラフィックス

JavascriptやCSSなどを用いて、動的にグラフィックスを生成したりアニメーションを実現したりする技術

レスポンスWebデザイン

一つのソースで、スマートフォンなどの様々なデバイスの画面サイズに対応させるための技術

通信・デバイスアクセス系API

JavaScriptからクラウドと通信をして情報の送受信を行ったり、センサーなどのデバイスにアクセスしたりする技術

マルチメディア

3D・動画・音声ファイルなどのマルチメディアコンテンツの表示・再生に関する技術

ユーザビリティ

JavascriptやCSSなどを用いて、デザイン仕様に沿った見やすい表示や操作しやすいコンテンツを作成するための技術

パフォーマンス

ストレージや並列処理を使ってコンテンツを効率よく高速に動作させたり、オフラインでも動作する仕組みを作るための技術

JavaScriptプログラミング **Level.2**

JavaScriptを使って、動的なWebコンテンツを作成する技術。



レベル1試験について

所要時間	90分（機密保持契約とアンケートの時間を含む）
試験問題数	約60問
受験料	¥16,500（税込）
試験実施方式	コンピュータベーステスト（CBT） ※マウスによる選択方式がほとんどですが、入力問題も多少出題されます。
受験場所	全国各地の試験センターでの受験か、自宅や職場からのオンライン受験が可能
合否結果	試験終了と同時
優位性の期限	5年間



試験のバージョン

最新のレベル1の試験のバージョンはVer2.5です

Ver2.0 2017年4月1日～2022年7月31日

Ver2.5 2022年2月1日～

Ver2.0とVer2.5の違いについては公式サイトで案内されています

https://html5exam.jp/images/outline/html5_lv1_ver25.pdf



学習方法



学習方法

公式サイトでの学習方法を参考にする

セミナーの紹介や、メールマガジン、サンプル問題があるので活用する

<https://html5exam.jp/measures/>

教本を使う

認定教材を読み込む

サンプル問題があったら、全問正解するまで挑戦する

実際にコードを書く

[CodePen](#)などの、即座に確認できるものを使えば捗ります

わからないところは検索

HTML5の仕様などが掲載されているサイトがあるので、教本などと合わせて理解を深める



自分の体験



自分の体験

何故取得したか

弊社がLPI-Japan HTML5のアカデミック認定校ということもあり
新人研修の一環で、HTML5レベル1の試験を受験し合格することが
目標の1つであった。

学習期間

4月に入社し、その後6月から試験勉強、
7月1日に受験し合格したので、約1か月。
研修期間であったこともあり勉強の時間はしっかり設けられていた。
いつ受けるかという目安を一旦決めておくとよいかもかもしれない。
公式サイトにある期間目安は3か月～半年

試験勉強前の知識

情報系の大学に通っていたこともあり、
HTMLとCSSの基礎は学習済み。

初めてのHTML、CSSとして試験勉強を始めるよりは、
HTML、CSSの基礎学習の仕上げで受験するというのがよいかも。

実際にサイトを作りながら学んでゆく参考書がおすすめ。
基本有料だが動画で学べるサイトもある。



自分の体験

どのように学習したか

学習に利用したもの

- 対策テキスト&問題集

「HTML5プロフェッショナル認定試験レベル1 ver2.0対応版」

※現在はver2.5、LPI-Japan HTML5 認定教材

- 公式サイトサンプル問題

<https://html5exam.jp/measures/sample.html#lv1>

最初はテキストをじっくり1周読む。

その後、手を動かして学習。公式で「重要度」も設定されている。

後はとにかく問題を解く。



試験のポイント



出題範囲

1.1 Webの基礎知識

1.2 CSS

1.3 要素

1.4 レスポンシブWebデザイン

1.5 APIの基礎知識



1.1 Webの基礎知識

1.1 Webの基礎知識

1.1.1 HTTP, HTTPSプロトコル

1.1.2 HTMLの書式

1.1.3 Web関連技術の概要

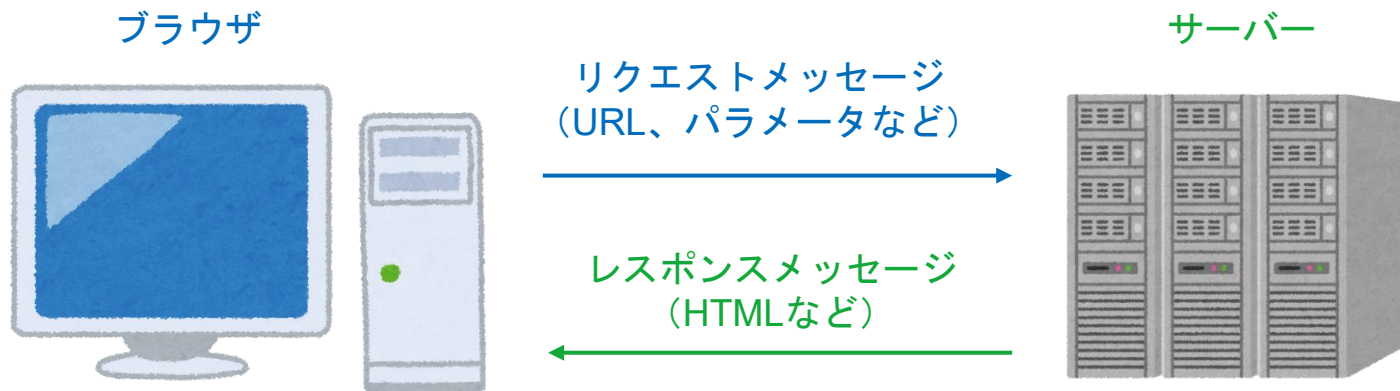
HTTP (HyperText Transfer Protocol) とは

サーバーとブラウザに代表されるクライアント間で

HTML文書などのテキストメッセージを受け渡すためのプロトコル。

※プロトコル=仕組み、手順

HTTPでの通信は以下の図のようになっており、
これを行うことでブラウザにページを表示する。



ステータスコード

レスポンスメッセージにはレスポンスの状態を3桁の数字で表す
ステータスコードが含まれる。

何に関するコードが何番台か覚えておくのがよい。例も覚えられるとなおよし

ステータスコード	概要
1xx	Informational (情報提供のコード)
2xx	Success (成功を表すコード) 例) 200 OK
3xx	Redirection (転送に関するコード) 例) 301 Moved Permanently, 307 Temporary Redirect
4xx	Client Error (クライアントエラーに関するコード) 例) 400 Bad Request, 404 Not Found
5xx	Server Error (サーバエラーに関するコード) 例) 500 Internal Server Error, 503 Service Unavailable

HTMLの全体構造

```
<!DOCTYPE html>
<html>
  <head> (開始タグ)
    . . .
    <title>文書のタイトル</title>
  </head> (終了タグ)

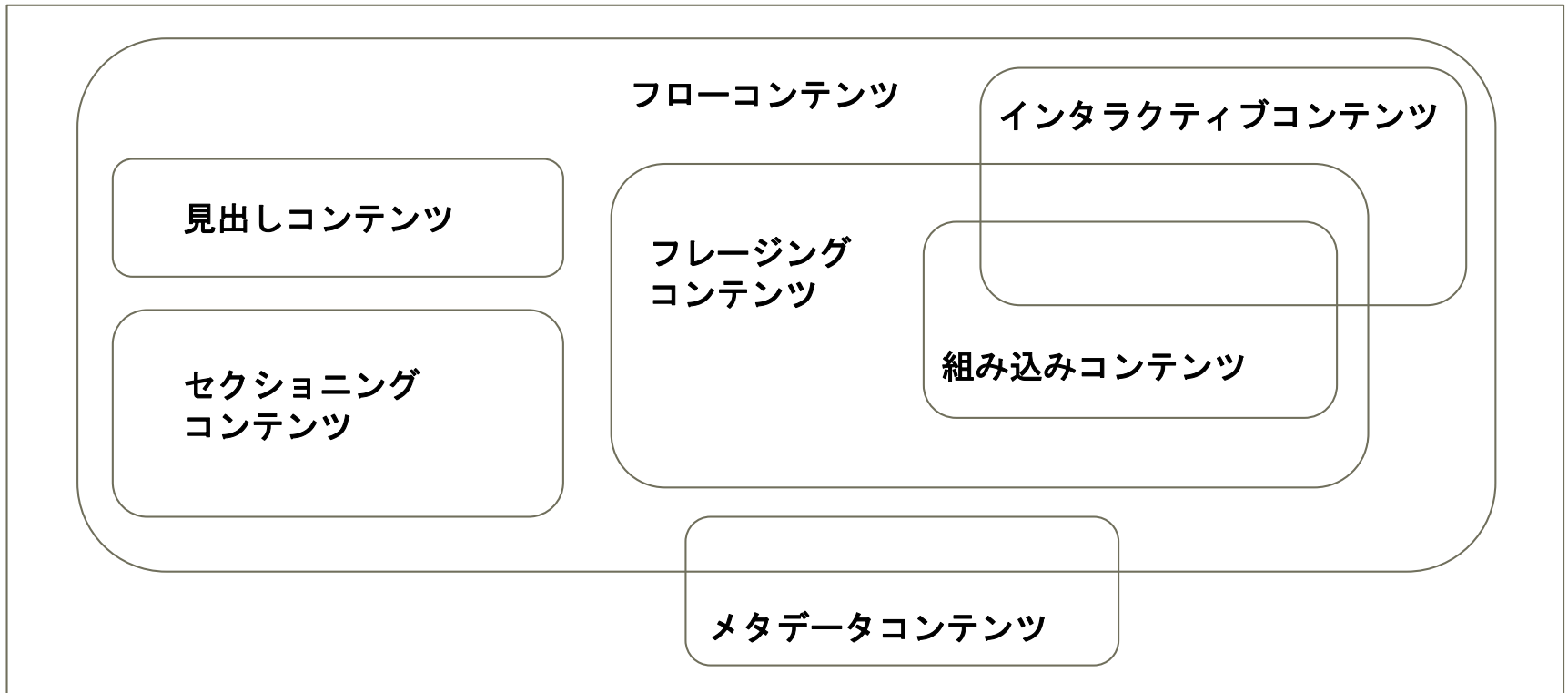
  <body>
    . . .
  </body>
</html>
```

先頭にはDOCTYPE宣言を配置する必要がある。

その後にはhtml要素を配置し、その子要素にhead要素とbody要素を順にひとつずつ配置する。

要素のカテゴリー

※多くの要素は複数のカテゴリーに該当するが、どれにも属さない要素もある。



7つのカテゴリー

HTMLの各要素には、子要素として入れられる要素が決まっており、そのルールのことをコンテンツ・モデルと呼ぶ。
親要素のコンテンツ・モデルがそのまま自分のものになる特別な要素をトランスペアレントと呼ぶ。例) a要素

正)

```
<p>
```

```
  <a href="#">a要素は<em>トランス・ペアレント</em>で
```

```
  す</a>
```

```
</p>
```

誤)

```
<em>
```

```
  <a href="#"><p>トランス・ペアレント</p></a>
```

```
</em>
```

p要素はフローコンテンツ。コンテンツ・モデルはフレージングコンテンツ。
em要素はフレージングコンテンツ。コンテンツ・モデルはフレージングコンテンツ。



1.1.3 Web関連技術の概要

XHTML

HTMLをXMLの文法で定義しなおしたものの。ルールが厳密。

- ・ 文章の先頭にXML宣言文が必要
- ・ 文字の大文字、小文字が厳密に区別される。
- ・ 要素は必ず開始タグ、終了タグでくくる。

Webサイトへの不正な攻撃手法

- ・ SQLインジェクション
- ・ クロスサイト・スクリプティング (XSS)
- ・ クロスサイト・リクエスト・フォージェリ (CSRF)
- ・ ディレクトリ・トラバーサル
- ・ HTTPヘッダインジェクション



1.1.3 Web関連技術の概要

画像ファイルフォーマット

- ・ BMP
- ・ GIF
- ・ PNG
- ・ JPEG
- ・ SVG

それぞれに特徴があるので覚えておくとよい。

その他

セッション、Ajax、Base64、DOMなどなど



1.2 CSS

1.2 CSS

1.2.1 スタイルシートの基本

1.2.2 CSSデザイン

1.2.3 カスケード(優先順位)

CSSの基本的な書式と名称

(セレクタ)

```
h1 {
```

(宣言ブロック)

```
    color : white;  
    font-size : 24px;  
    text-shadow : 1px 1px 2px black;
```

```
}
```

(プロパティ名) : (プロパティ値)

```
h1 { color : white }
```

セレクタは

コンマで区切ると複数指定

```
h1, h2, p {
```

```
}
```

空白を入れると子要素指定

```
a span {
```

```
}
```



1.2.1 スタイルシートの基本

セレクタの種類

セレクタには複数の種類がある。

タイプセレクタ

指定した要素が対象

```
h1 { color : white }
```

ユニバーサルセレクタ

要素名の代わりに*を指定することで全要素が対象になる。

```
* { color : white }
```



1.2.1 スタイルシートの基本

以降に記述するセレクタは、
タイプセレクタまたはユニバーサルセレクタに
順不同で必要な数だけ連結できる。

クラスセレクタ

ピリオドの後にクラス名を指定することで、
そのクラス名を持つ要素が対象になる。

```
.subtitle { color : white }
```

```
p.subtitle { color : white }
```

IDセレクタ

#の後にid名を指定することで、そのid名を持つ要素が対象になる。

```
#subtitle { color : white }
```


1.2.1 スタイルシートの基本

属性セレクタ

特定の属性が指定されている要素、または特定の属性に特定の値が指定されている要素を適応対象として指定できる。

```
img[src$=".jpg"] { border : 3px solid black }
```

リンク関連の疑似クラス

ある要素が特定の状態にあるときなどに限定して適用対象にする。

```
a:hover { color : blue }
```

:hoverはマウスカーソルが上に乗っている状態。

その他

:link（リンク先をまだ見ていないa要素） :visited（リンク先をすでにみているa要素） :active（マウスのボタン等が押されている状態）がある。

その他の疑似クラス

例) :nth-of-type(式)・・・先頭から○個目の要素から△個おきに適応（指定した要素と同じ要素のみ対象）

```
p:nth-of-type(2n+1) { color : red }
```

```
div:nth-of-type(1) { color : blue }
```

他にも様々な疑似クラスがあるので書き方と適応対象を覚えておくとよい。

```
<div>
    <p>pの1番目</p>
    <p>pの2番目</p>
    <div>divの1番目</div>
    <p>pの3番目</p>
    <p>pの4番目</p>
    <p>pの5番目</p>
</div>
```

長さを表す単位

CSSでの長さを表す際に使用する単位には、

「絶対的な長さを表す単位」と「相対的な長さを表す単位」がある。

px	ピクセル（1/96インチを1とする単位）	絶対
em	要素のフォントサイズを1とする単位	相対
rem	html要素のフォントサイズを1とする単位	相対
vw	ビューポートの幅をの1%を1とする単位	相対
vh	ビューポートの高さの1%を1とする単位	相対
%	親要素の幅、高さ、フォントサイズに対して何%の大きさか ※厳密には長さの単位の分類ではない	相対

※ビューポートとはWebページを表示させる領域のこと
要するに画面全体

1.2.3 カスケード（優先順位）

CSSは同じ指定元の中では後に書いたものが上書きする形で優先される。

※後に書いたものが優先されるのは詳細度が同じ場合に限る。

!importantを付けると順序や詳細度にかかわらず優先される。

例) index.css内での記述

```
011 h1.title {  
    font-size : 24px;  
}  
001 h1 {  
    font-size : 30px; !important  
}
```

!important

011

001



セレクタの詳細度による優先順位

1桁目 → タイプセレクタの個数

2桁目 → クラスセレクタの個数 + 属性セレクタの個数 + 擬似クラスの個数

3桁目 → IDセレクタの個数

例)

body div#wrapper	102
div#wrapper	101
#wrapper	100
p.subtitle	011
.subtitle	010
body h1	002
h1	001
*	000





1.3 要素

1.3 要素

1.3.1 要素と属性の意味（セマンティクス）

1.3.2 メディア要素

1.3.3 インタラクティブ要素

セクションと見出しの関係

セクションとは見出しとそれに対応する文章全体を含む範囲。

例えば長い文章における「章」「節」、新聞やブログなどの「1記事」。

セクションを作り出すセクショニングコンテンツに分類される要素は4つ。

section → 一般的なセクション。他3つは特定の用途向けなのでそれ以外で使用

article → 「それだけで全部の / それだけで完結している」セクション。

ブログの記事や学術論文などでも使用できる。

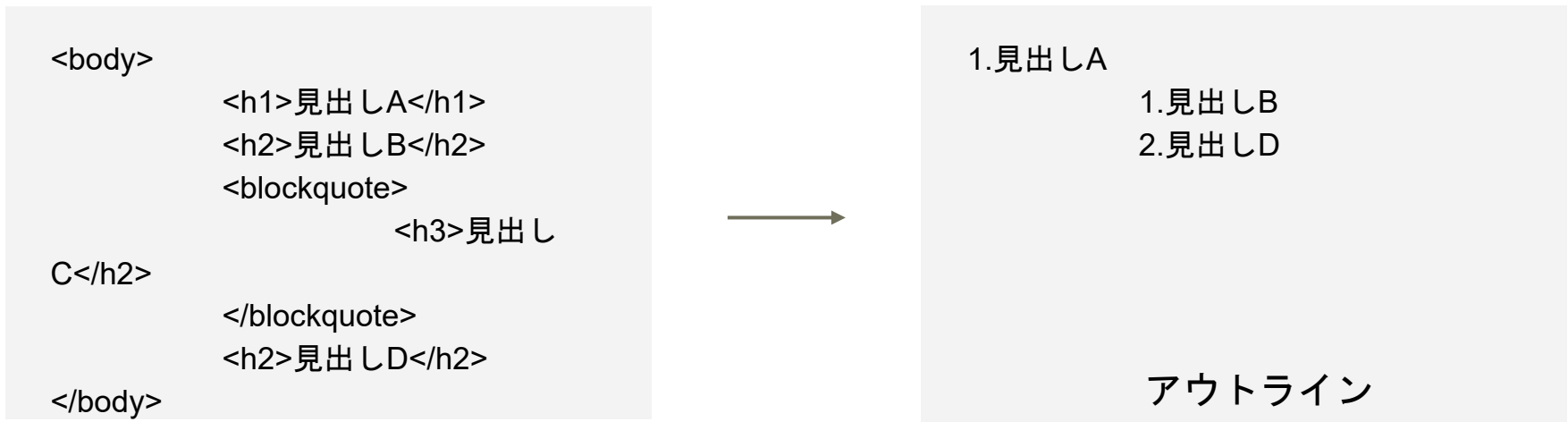
nav → ナビゲーション部分のセクションを示すための要素。

aside → メインコンテンツの内容と関係が薄く別扱いにしたほうがよさそうな

セクション。例えば広告、補足記事、新聞で記事と別枠で囲って掲載する内容。

1.3.2 要素と属性の意味（セマンティクス）

セクションニングコンテンツは、直接見た目に出るものではありません。
しかし、**アウトライン**というのが形成され、検索エンジンの解釈に影響がでます。



セクションニング・ルート

独立したセクションを持ち、アウトラインに影響を与えなくなる要素。

body要素 blockquote要素 fieldset要素 figure要素 td要素 details要素

1.3.2 メディア要素

画像・動画・音声・その他（イメージマップ、外部データなど）を表示、再生させる要素。

画像

img要素、picture要素、source要素

動画

video要素、track要素(video要素内で使用)

音声

audio要素、
track要素(audio要素内で使用)

その他

embed要素、map要素、area要素、
object要素、param要素

指定できる属性がいろいろあり、特に画像・動画・音声に関する要素は指定できる属性とその属性の働きを把握しておくといよい。

例)

```

```



1.3.2 インタラクティブ要素

a要素

href属性にリンク先の値を指定すると要素がリンクになる。

target属性に_blankを指定すると別タブで開くリンクになる。

a要素は使用頻度がすごく高いので他属性なども含め覚えておくとよい。

例) `別タブリンク`

アルファベット1文字系の要素は惑わせられるので注意

q要素、b要素、i要素、s要素、u要素、p要素

アルファベットが何の略なのか理解しておくと覚えやすい

qはquote、bはbold、iはitalic、sはstrike-through、uはunderline、pはparagraph



1.4 レスポンシブWebデザイン

1.4 レスポンシブWebデザイン

1.4.1 マルチデバイス対応ページの作成

1.4.2 メディアクエリ

1.4.3 スマートフォンサイト最適化



1.4.1 マルチデバイス対応ページの作成

レスポンシブWebデザインとは

1つのHTMLで、デバイスの特性（主に画面幅）に応じてレイアウトやデザインを変更する手法。

メリット

- ・ 1つのHTMLで完結するためリダイレクトの設定が不要になり、サイトの読み込み時間を短縮できる
- ・ URLが同じになるためSEO的に有利

デメリット

- ・ モバイル向けには動画や画像を切り替えるケースもあり実装が難しくなることがある
- ・ HTML/CSSのサイズが増加し、場合によってはページの表示や動作が重くなる



1.4.2 メディアクエリ

レスポンシブWebデザインの実現には様々な手段が利用されるが、主に「**メディアクエリ**」というものが利用される。

メディアクエリとは、デバイスの特性によってCSSを切り替えるものになります。

例) (HTMLファイルでの記述)

```
<link rel="stylesheet" media="screen and (max-width: 480px)" href="style.css">
```

上記は、幅が480pxまでのデバイスに、style.cssを適用することができるようになります。

また、CSSに切り替えの条件を記述する方法もある。

```
@media screen and (max-width: 480px){  
    .title{  
        color : red;  
    }  
}
```

Fluid Grid

ウィンドウ幅に合わせてコンテンツの数や幅を変更する技術となります。

Fluid Gridの実現には、コンテンツ幅は、絶対的な長さを表す単位（pxなど）ではなく相対的な長さを表す単位（%など）で指定する。

PC



スマートフォン





1.5 APIの基礎知識

1.5 APIの基礎知識

1.5.1 マルチメディア・グラフィックス系API概要

1.5.2 デバイスアクセス系API概要

1.5.3 オフライン・ストレージ系API概要

1.5.4 通信系API概要

マルチメディア

HTML5で追加されたaudio要素とvideo要素により、簡単に音声や動画といったマルチメディアコンテンツを埋め込み可能となった。

埋め込んだマルチメディアコンテンツはJavaScriptを使って制御できる。

JavaScriptで可能な操作

- ・コンテンツの再読み込み
- ・再生開始
- ・再生中断

JavaScriptで取得し利用可能なデータ

- ・コンテンツが再生・早送り等可能な状態か
- ・ネットワークの状態
- ・エラー
- ・コンテンツの場所
- ・コンテンツの長さ
- ・音量
- ・トラック情報

グラフィックス

HTML5ではCanvasと呼ばれるJavaScriptを使って画像を描画するための仕様を利用できるようになった。

Canvasは画像をビットマップ形式で描画する。

そのため拡大縮小すると画像は粗くなる。

Canvasでは以下の描画操作などが可能。

- ・ 線を描く
- ・ 円、四角を書く
- ・ 色を塗る
- ・ テキストを書く
- ・ 画像ファイルを読み込む
- ・ 拡大、縮小、回転する

Geolocation API

ユーザーの位置情報を扱うためのAPI。

無線LAN、WiFi、携帯電話基地局、GPS、IPアドレスといったソースから位置情報を取得。

取得できる情報は以下

緯度、経度、高度、緯度経度の精度、高度の精度、方角、速度

DeviceOrientation Event

デバイスの方角や傾きが変化したときに発生するイベント。

取得できる情報は以下

デバイスの頭が指す方角、デバイスの上下左右方向の傾き

デバイスの向きによって表示内容を変更するコンパスアプリなどに利用。



1.5.2 デバイスアクセスAPI

Touch Events

タッチパネルなどの画面を触るなど、**画面を指で操作している間に、その状態が変化すると発生するイベント。**

イベントは以下がある

画面をタッチ、タッチしたまま動かす、画面から離す、タッチへの割り込み

DOM3 Events (UI Events)

マウスやキーボードなどの入力操作を取り扱うためのイベント。

イベントは以下がある

UIイベント、フォーカスイベント、マウスイベント、ホイールイベント、入力イベント、キーボードイベント、間接的テキスト入力イベント

Web Storage

キーと値の組み合わせでブラウザにデータを保存し、利用するAPI。

Web Storageにはセッションストレージとローカルストレージの2種類がある。

cookieに似てはいますが、保存容量と有効期限が違います。

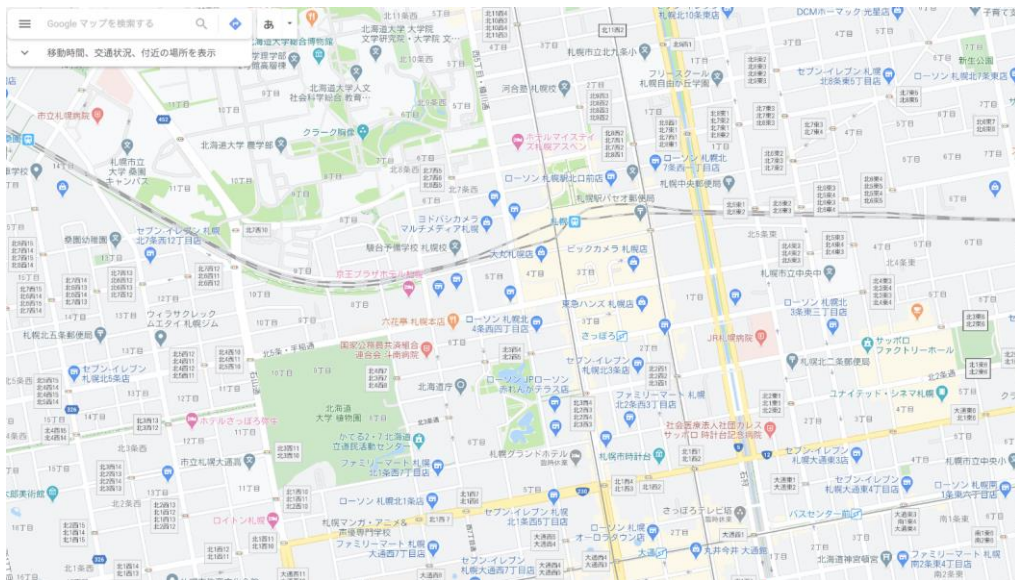
	容量	有効期限
cookie	4KB	指定した期限まで
セッションストレージ	5MB	ウィンドウやタブを閉じるまで
ローカルストレージ	5MB	永続的に有効

XMLHttpRequest

JavaScriptでのHTTP通信を実現するAPI。

現在表示しているページから画面遷移を伴わずHTTP通信を行い各種データを取得する。

↓地図アプリケーションはこの技術が利用されている例





実務で役に立ったCSS

font-family プロパティ

フォントの種類を設定するプロパティ。

例)

```
body {  
    font-family : “メイリオ”, “ヒラギノ角ゴ Pro”, Helvetica, sans-  
    serif ;  
}
```

フォントはユーザーの閲覧環境に依存するので、できるだけ多くの環境に対応したフォントを指定しておく必要がある。OSによって標準で搭載されているデバイスフォントには違いがあるが、共通のものもあるのでそういったものを使うようにするとよい。

基本的に保険のような感じで複数設定しておくことが多く、最初に書いたものから優先度が高い。

デバイスフォントは

font-familyのみで利用できるが、それ以外のサーバーにアップロードされている

または自分でサーバーにアップロードしたWebフォントは@font-faceという

書式を使用して指定する必要がある。

例)

```
@font-face {  
    font-family : MyFont ;  
    src : url(http://example.com/fonts/FontRegular.woff) ;  
}  
  
p {  
    font-family : MyFont, sans-serif ;  
}
```

——— フォントの名前を付ける

——— フォントのアドレスを指定

CSSのみでアニメーションを実現するにはまずキーフレームという特別な書式を使用してどのタイミングでどの値がどう変化するのか設定する。

```
@keyframes 名前 {  
  0% {  
    プロパティ: 値;  
  }  
  〇% {  
    プロパティ: 値;  
  }  
  100% {  
    プロパティ: 値;  
  }  
}
```

%はアニメーション全体の長さ（再生時間）の何%の時点かを表す。

※0%はfrom、100%はtoと書くこともできる。

キーフレームで設定した動きを実行させるには
animation-name プロパティで **キーフレームの名前を指定し実行させる**、
animation-duration プロパティで **再生時間を設定する**。

例)

```
.sample:hover{  
    animation-name :  
    jumping ;  
    animation-duration : 5s ;  
}
```

他にもオプショナルなプロパティがある。 **animation** プロパティで一括指定もできる。

animation-timing-function プロパティ
再生速度の変化のパターン

animation-direction プロパティ
逆再生させるかとそのタイミング

animation-delay プロパティ
再生の開始を遅らせる

animation-fill-mode プロパティ
delayによって遅らせている間および再生終了後の表示

animation-iteration-count プロパティ
何回繰り返して再生させるのか

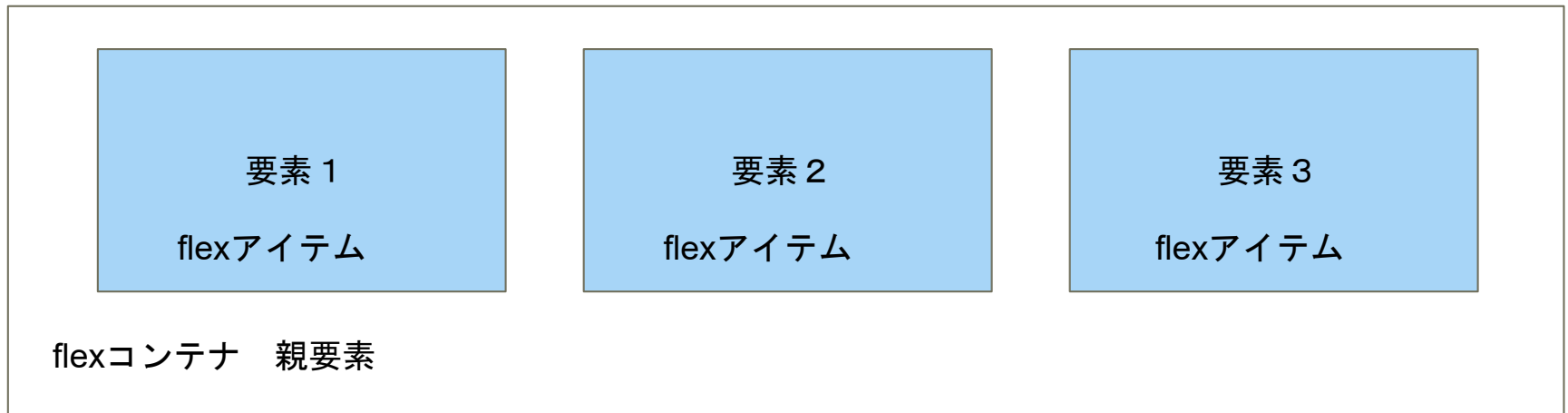
animation-play-state プロパティ
再生を一時停止させる

HTMLの要素はシンプルな状態だと縦に積まれていく。

横並びにするにはdisplayプロパティのプロパティ値 flexを使うことが多い。

```
<div class="container">
  <div>要素 1 </div>
  <div>要素 2 </div>
  <div>要素 3 </div>
</div>
```

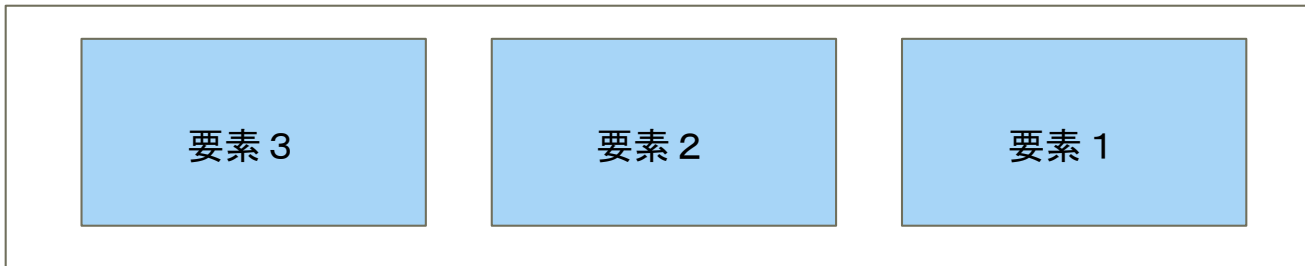
```
.container {
  display : flex;
}
```



flex-direction

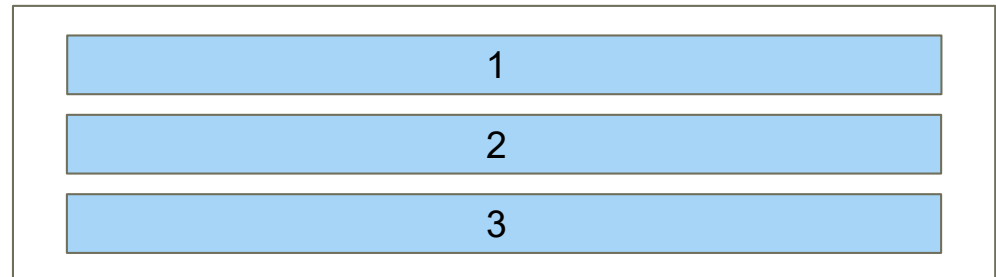
フレックスアイテムの並ぶ方向を設定する

flex-direction : row-reverse; 横方向逆並び



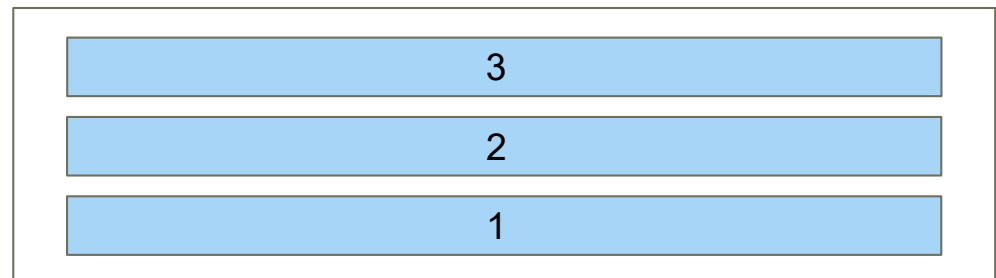
flex-direction : column;

縦方向並び



flex-direction : column-reverse;

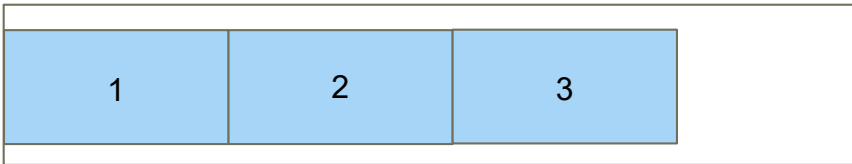
縦方向逆並び



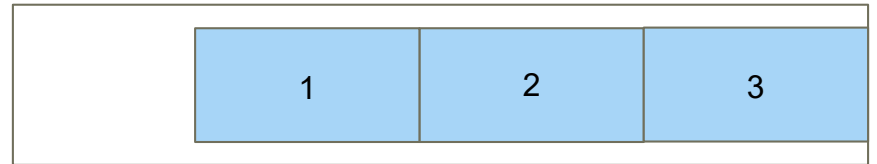
justify-content

水平方向の配置

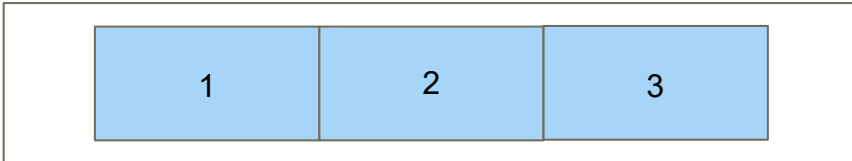
justify-content : flex-start;



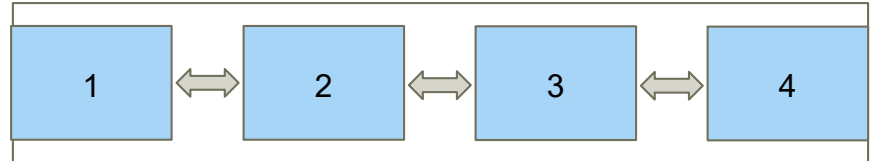
justify-content : flex-end;



justify-content : center;

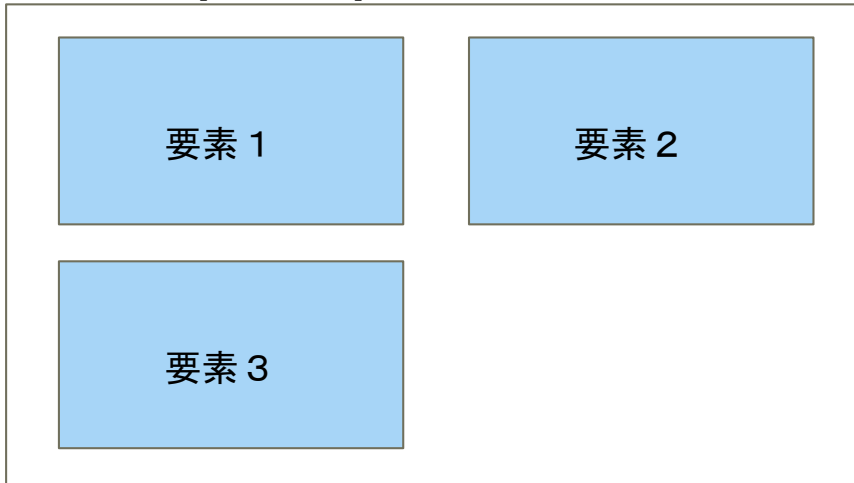


justify-content : space-between;



flex-wrap

flex-wrap : wrap; flexアイテムの折り返し

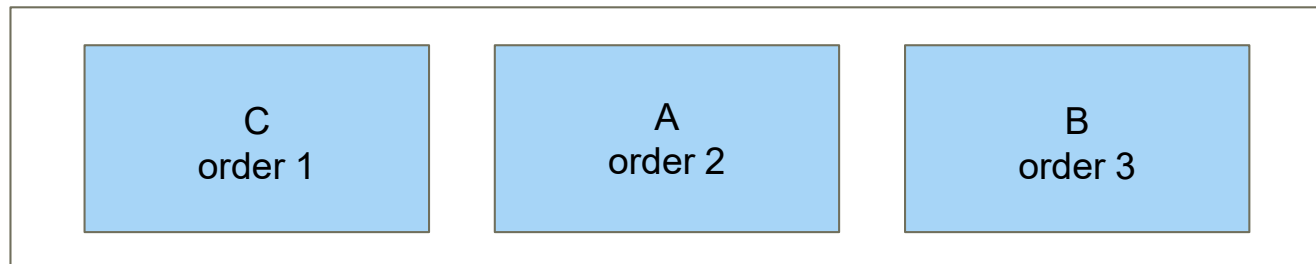


order

flexアイテム（子要素）に

order : 数字;

と指定することで順番を入れ替えられる





最後に

プログラミングの道へ進むとこれ以上に技術や知識がいっぱい

全てを覚えて完璧に使いこなすなんて無理！！

人間だもの.. 難しい言葉ばっかだし...



実務でも分からなければその時調べながら作業するし

誰かに聞いてみたりする



焦らずに自分のペースで頑張っていきましょう！

ご清聴ありがとうございました。



ご質問



公式アカウント、URL

HTML5プロフェッショナル認定試験のサイト

<http://www.html5exam.jp/>

Twitter

@html5Cert

Facebook

<https://www.facebook.com/html5exam>

過去のセミナーでの配布資料や、イベント情報などが配信されています。

LPI-JAPAN HTML5 Professional Certification

Open the Future with **HTML5**.