



オープンソースカンファレンス 2019 Hokkaido

これから始める！ HTML5プロフェッショナル認定試験 レベル1

資格を取るために大切なことと、
資格を取った後に大切なこと

2019年6月1日（土）@OSC 2019 Hokkaido

F() R K 株式会社フォーク
川谷 晴



自己紹介



自己紹介

川谷 晴

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

2017

システムエンジニアからweb業界への転職を志し、株式会社フォーク入社
webページ制作などの案件に携わりながら、
HTML5プロフェッショナル認定試験レベル1取得

2018

業務内容を徐々にレベルアップし、javascript開発などを担当

2019

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: **40**名

Lv2: **7**名

※2018年6月時点



Lv1: **41**名

Lv2: **13**名

※2019年6月時点

僕が所属する札幌ブランチでは、
7名がレベル1に合格しています



F()RK

<https://www.fork.co.jp/>
<https://4009.jp/>



本日解説する内容

- ゼロから始めるHTML5
 - ゼロからってどこから？
 - なぜ資格を取得したのか
 - どうやって勉強したのか
 - 試験概要
- 資格を取って終わりにしないために
 - 資格を取るメリット
 - 資格を活用する
 - 実務で体感したメリット
 - 次のステップへ



ゼロから始めるHTML5



本日のメインターゲット

「ゼロから」な人

- 「非IT業界からの転職」
- 「なんとなくweb業界に興味がある」
- 「異世界から転生したらwebエンジニアだった」

そんなに「ゼロから」でもない人

- 「IT業界の異業種から転職」
- 「実務で使っているけど、この資格はよく知らない」



なぜ資格に挑戦したのか

web系の知識がなかったから

- 簡単なhtmlやcssくらいは書けるけど、実務なんてとても...
- 勉強して資格を取ることができれば、ある程度の知識は保証できるのでは？
- 会社で取得を推奨している 乗るしかないこのビックウェーブに



どうやって勉強したのか

まずは教本

- 試験範囲の解説と、練習問題が載っているタイプ
- 通勤時間や自宅で一通り読み込む

リファレンス

- web上にはHTML5の仕様を解説しているリファレンスがある
- 教本を読んでもわからない部分や気になる機能は、リファレンスをチェック

ひたすら問題を解く

- 教本に付属の練習問題をひたすら解く
- 公式サイトに掲載されている練習問題も、全問正解するまで何度も挑戦する



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資格



2つのレベル



HTML5 レベル1 (今回解説するのはこちら)

マルチデバイスに対応した静的なWebコンテンツをHTML5を使ってデザイン・作成できる。

主な内容：

- ・ HTML5
- ・ CSS3
- ・ レスポンシブwebデザイン



HTML5 レベル2

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。

⇒ javascriptに関する問題の比率が大きくなる



試験の概要

試験実施方式	コンピュータベーステスト（CBT） ※ほとんどの問題が選択式
合否結果	試験終了と同時
所要時間	90分（機密保持契約とアンケートの時間を含む）
問題数	約60問
合格ライン	約7割



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

1.1.1 HTTP, HTTPSプロトコル

Webの基本的な通信技術である、HTTP/HTTPSプロトコルに関する問題

1.1.2 HTMLの書式

HTMLの基本的な記述方法や、仕様に関する問題

1.1.3 Web関連技術の概要

画像ファイルやスクリプト言語など、HTMLの周辺技術に関する問題

POINT :

HTTPステータスコードの種類

HTMLの文書型宣言のバリエーション



1.2 CSS

1.2.1 スタイルシートの基本

HTMLの"見た目"を指定する、CSSの基本的な記述方法や文法に関する問題

1.2.2 CSSデザイン

要求された見た目(デザイン)を、どんな記述をすれば実現できるかを問う問題

1.2.3 カスケード(優先順位)

複数のCSSが同時に適用された場合の優先順位や計算方法に関する問題

POINT :

CSSのプロパティ(設定値)の種類

⇒ CSSでどのようなデザインを実現できるのか？

1.3.1 要素と属性の意味（セマンティクス）

HTMLを構成する要素の種類と、それぞれが持つ役割・意味に関する問題

1.3.2 メディア要素

動画や静止画を、HTMLのコンテンツとして活用するための知識を問う問題

1.3.3 インタラクティブ要素

フォームやリンクなど、ユーザーの操作を伴うHTML要素に関する問題

POINT :

HTML要素(タグ)の種類・役割

⇒ それぞれの要素に「文書」としての役割がある



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

様々な画面サイズやデバイスに対応したwebページの作成方法に関する問題

1.4.2 メディアクエリ

閲覧環境に応じてスタイルを切り替える "メディアクエリ"に関する問題

1.4.3 スマートフォンサイト最適化

スマートフォン環境を考慮したページ制作の技術や注意点に関する問題

POINT :

マルチデバイスに対応する方法それぞれのやり方や注意点



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

javascriptで、画像や音声などを適切に扱うための知識を問う問題

1.4.2 デバイスアクセス系API概要

端末に付属するセンサー類を、javascriptを使って扱うことに関する問題

1.4.3 オフライン・ストレージ系API概要

ブラウザ内へのデータ保存や、バックグラウンド処理を行う技術に関する問題

1.4.4 通信系API概要

javascriptを用いて通信を行う技術に関する問題

POINT :
javascript APIの種類と概要



資格を取って終わりにしないために



資格を取って終わりにしないために

資格を取得するメリット

「資格をとるとどうなる？」 「知らんのか？」



資格を取得するメリット

基本的な知識が身につく

- 比較的幅広い範囲の問題が出題される
- 基礎的な知識に関する問題の割合は多め



資格を取得するメリット

綺麗なコードを書くことができる

- ただ書くだけなら資格を取らなくても書ける
- 「正しく」「綺麗に」「早く」書くためには知識が必要



資格を取得するメリット

実現できること・できないことを把握できる

- 「できること」がわかれば、より良いアイデアを提案できる
- 「できないこと」がわかれば、代替案などの判断を素早く行える



資格を取得するメリット

フックが増える

- 勉強したからといってすべてを完璧に覚えられるわけではない
- 単語など簡単な情報だけでも覚えていれば、それを手がかり="フック"として、調べることはできる



資格を取って終わりにしないために

資格を活用する

ぶきはもっているだけじゃいみがないぞ！



資格を活用する

webページを作る

- 一番シンプルで直接的な活用法
- 実際に作ることで、知識に加えて経験も積める



資格を活用する

webページを作る"以外"のこと

- web業界はHTMLを書く職種だけではない
- 例)デザイナー⇒実際の実装をイメージしながらデザイン
- 例)ディレクター⇒作業量を見積りやすくなる
- etc..



誰かに教える

- 「教える機会」は「学ぶ機会」よりも貴重
- 周囲のレベルアップにつながる
- 自身も教えるために学びなおし、知識が定着していく



資格を取って終わりにしないために

実務で体感したメリット

何か良いことでもあったのかい？

正しさ・読みやすさを考えて書けるようになった

```
<p>おはし株式会社公式ホームページ</p>

<div>
  <p>ごあいさつ</p>
  <p>
    <span>楽しい箸をつくる企業、<span class="red">おはし株式会社</span>のホームページへようこそ！</span>
  </p>
</div>
```



```
<h1>おはし株式会社公式ホームページ</h1>

<section>
  <h2>ごあいさつ</h2>
  <div>
    <p>楽しい箸をつくる企業、<span class="red">おはし株式会社</span>のホームページへようこそ！</p>
  </div>
</section>
```

HTML5における要素(タグ)の分類 (1/2)

メタデータ・コンテンツ

HTMLファイルの情報や、他のファイルとの関係をあらわすもの
例) meta、script、style、title

フロー・コンテンツ

子孫にテキストなどを持つもの
例) a、p、header、h1、div、

セクショニング・コンテンツ

明示的に章を構成するもの
例) section、article

ヘッディング・コンテンツ

暗黙的に章を構成するもの
例) h1、h2、h3

■明示的な章

```
<section>  
  <h1>第1章：誕生</h1>  
  <p>小さな病院で生まれ...</p>  
</section>
```

■暗黙的な章

```
<h1>自己紹介</h1>  
<h2>誕生</h2>  
<p>雪国で生まれ...</p>  
<h2>育ての親</h2>  
<p>老夫婦に保護され...</p>  

```




HTML5における要素(タグ)の分類 (2/2)

フレーズング・コンテンツ

段落などの中に含まれる、文章を構成する要素

例) span、a、img、b

エンベディッド・コンテンツ

HTML内に他のリソースを埋め込むもの

例) iframe、img、audio

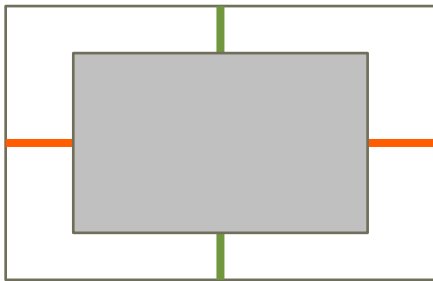
インタラクティブ・コンテンツ

閲覧しているユーザーによって操作ができるもの

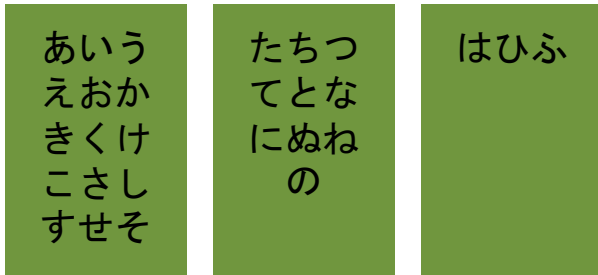
例) input、select、button

flexレイアウトを活用できるようになった

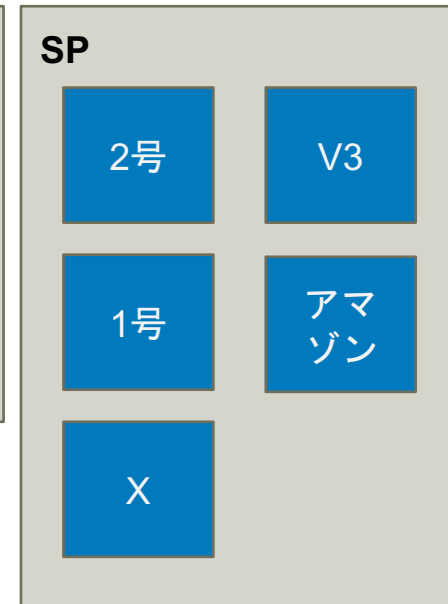
上下左右中央揃え



要素の高さを揃える



PCとスマホで順序が違う



だいたいflexでどうにかなる

display:flex;

指定した要素(コンテナ)の子要素(アイテム)を、設定したルールに従って整列させる

親要素 (コンテナ)

子要素 (アイテム)

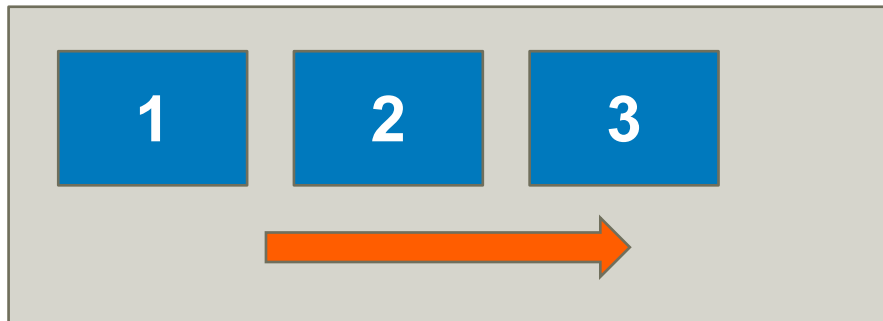
子要素 (アイテム)

子要素 (アイテム)

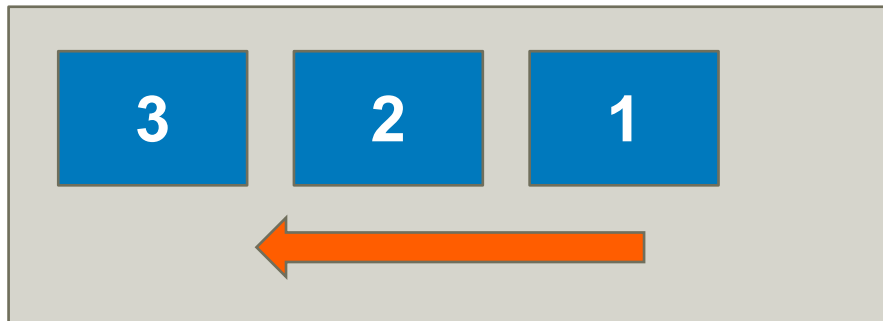
flex-direction

要素を並べる方向

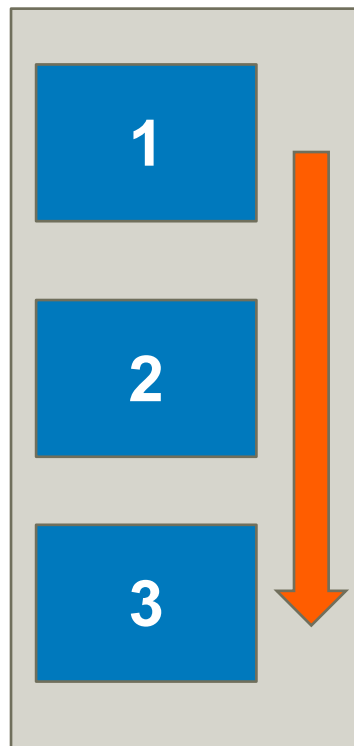
`flex-direction : row;`



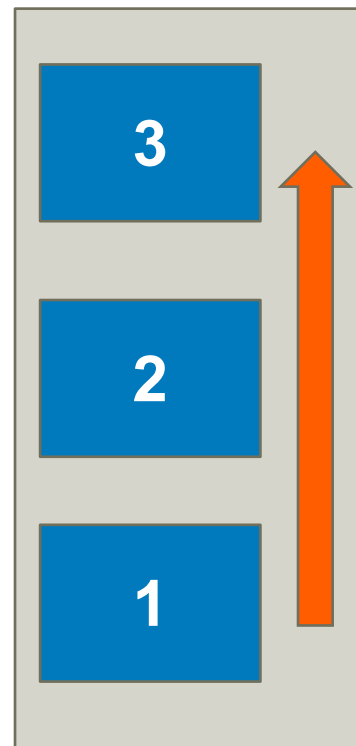
`flex-direction : row-reverse;`



`flex-direction : row;`



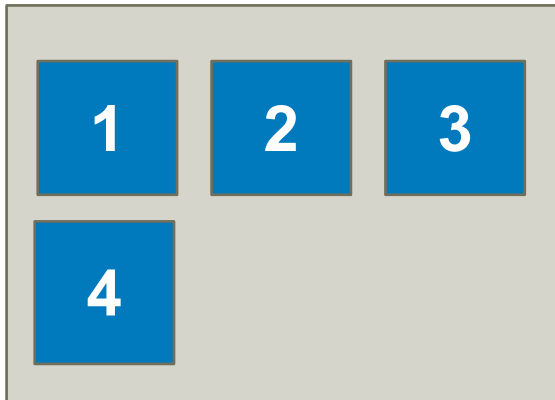
`flex-direction : row-reverse;`



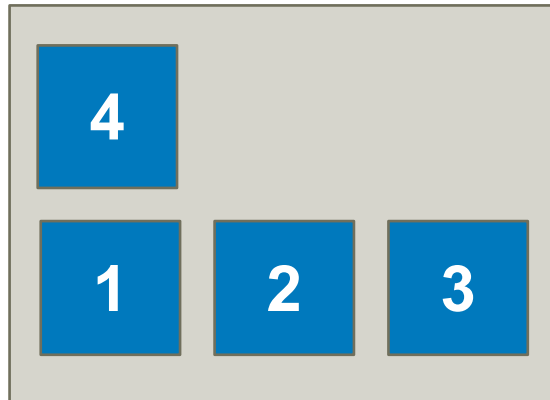
flex-wrap

子要素を折り返して並べるかどうか

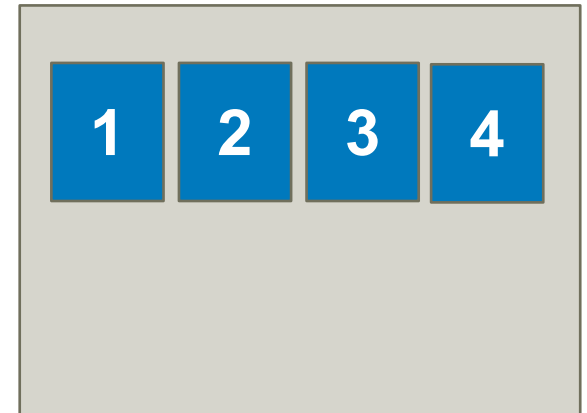
flex-wrap: wrap;



flex-wrap: wrap-reverse;



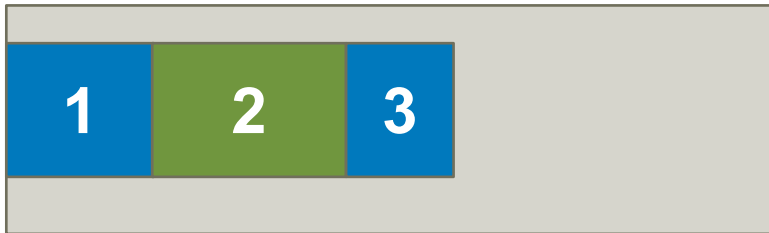
flex-wrap: nowrap;



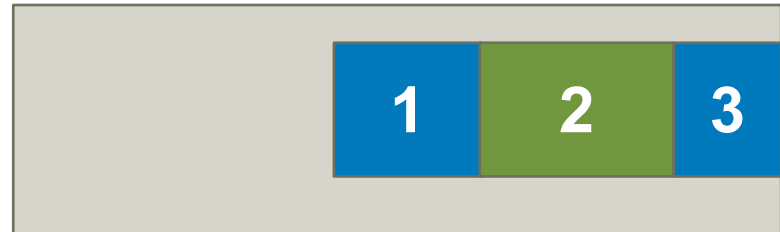
justify-content

アイテムとアイテムの、水平方向の揃え方

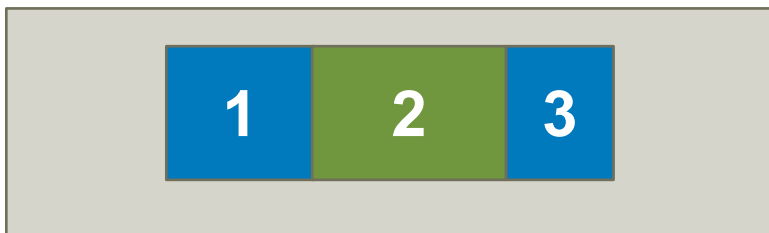
`justify-content:flex-start;`



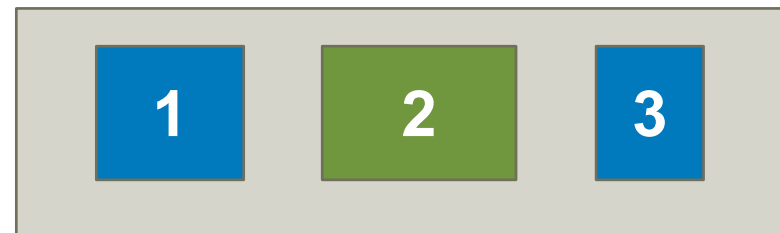
`justify-content:flex-end;`



`justify-content:center;`

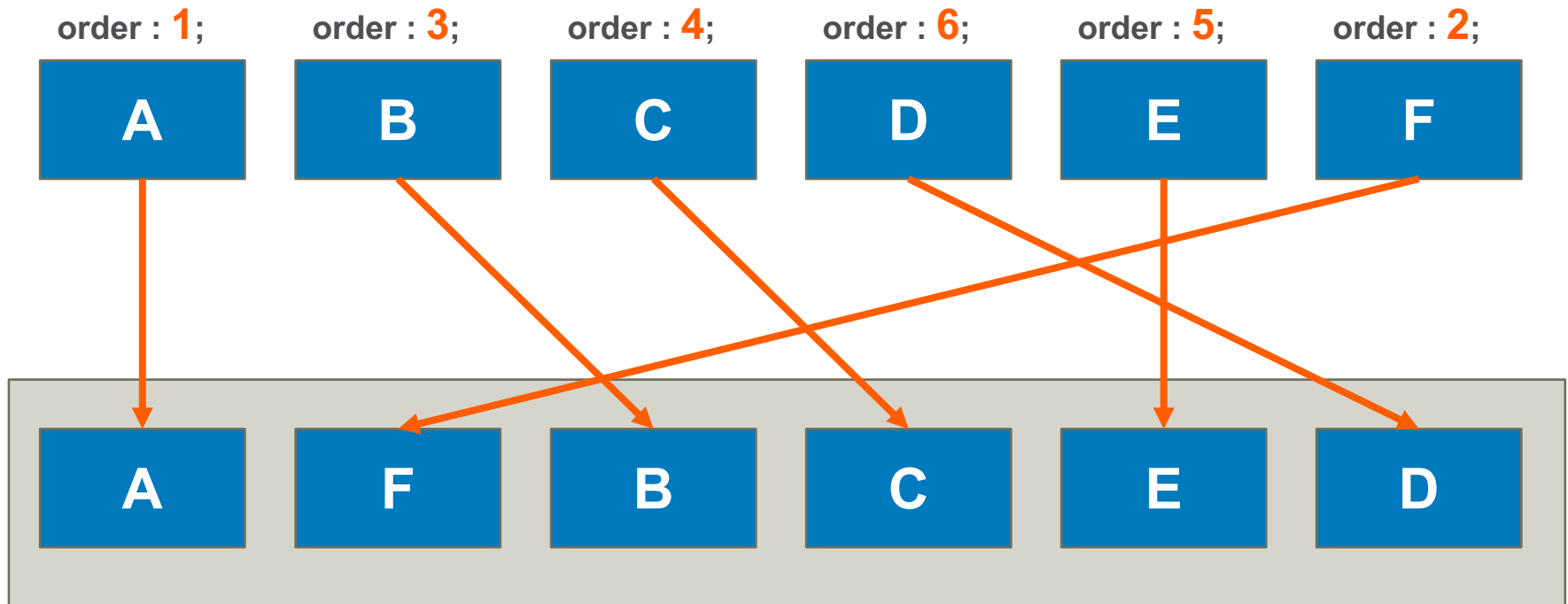


`justify-content:space-around;`



order

並び順を指定する



CSSの優先順位を理解できた

common.css

```
.text {  
  font-size: 12px;  
  font-weight: bold;  
  margin-top: 10px;  
}
```

base.css

```
p {  
  font-size: 10px;  
  font-weight: normal;  
  margin: 0;  
}
```

page.css

```
#top p {  
  font-size: 20px;  
  margin-top: 5px;  
}
```



```
<div id="top">
```

```
  <p class="text">一体どうなってしまうのか！？</p>
```

```
</div>
```




CSSの優先順位

優先順の基本

- htmlで読み込んだCSS > 閲覧者が設定しているCSS > ブラウザ標準スタイル
- 要素に[style=""]で直接設定 > head要素内に記述したCSS > 外部CSS
- 優先度同じの場合、あとから書いたほうが優先
- 奥の手 : !importantを付けたスタイルは優先される

もうひとつの優先順 ⇒ 点数制

CSSのセレクタ (要素を指定する単語)それぞれに点数が設定されている

⇒ 点数の合計が多いものが優先される

点数は、小数点で区切った4つの数値で表す (例 : 1.0.0.1)

セレクタ	例	点数
要素	p	0.0.0.1
クラス	.text	0.0.1.0
ID	#top	0.1.0.0
タグのstyle属性 (セレクタではない)	style="color:red;"	1.0.0.0



CSSの優先順位

優先度の例)

① **p.text.text-red#item { ~ }**

② **#top p.text#item { ~ }**

③ **p.text.ta.tb.tc.td.te.tf.tg.th.ti { ~ }**

優先度の例)

① p.text.text-red#item { ~ }

$0.0.0.1 + 0.0.1.0 + 0.0.1.0 + 0.1.0.0 = 0.1.2.1$

② #top p.text#item { ~ }

$0.1.0.0 + 0.0.0.1 + 0.0.1.0 + 0.1.0.0 = 0.2.1.1$

③ p.text.ta.tb.tc.td.te.tf.tg.th.ti { ~ }

$0.0.0.1 + 0.0.1.0 + (\text{中略}) + 0.0.1.0 = 0.0.10.1$

② > ① > ③



資格を取って終わりにしないために

Level1のその先へ

オレはようやくのぼりはじめたばかりだからな
このはてしなく遠いHTML5坂をよ...



Level1のその先へ

有意性の期限：5年間

- 有効期限ではない（過ぎても取得した資格は消えない）
- 変化の早いWeb技術の業界において、最新の技術を理解しているか否かの判断基準
- 期間内に再度同じ試験に合格するか、上位レベルの試験に合格すると更新される



Level1のその先へ

HTML5 レベル2

- レベル1の上位資格
- javascriptを活用したwebアプリケーション開発に関する問題が多くを占める
- レベル1で自信が付いたら是非
- 僕も頑張ります…

web業界は川のようなもの



外から見ると流れが速く見えるでも、流れが速いのは表面部分。底に近い所は流れが緩いし、地面はそう簡単には動かない。WEB業界も応用・発展的な技術は流れが速いが、基礎の基礎はすぐには変わらない。まずは川の中でしっかり立つところから始めてみよう。



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**.