



LPI-Japan主催

HTML5プロフェッショナル認定試験 レベル1 ポイント解説無料セミナー

2014年11月8日

アシアル株式会社 生形可奈子



事業内容

HTML5アプリ開発環境（Monaca）、ネイティブアプリ開発、
サーバーサイド開発、インフラ、教育事業など



■ HTML5プロフェッショナル認定試験とは

- 概要
- 試験範囲

■ カテゴリ毎の頻出ポイント解説

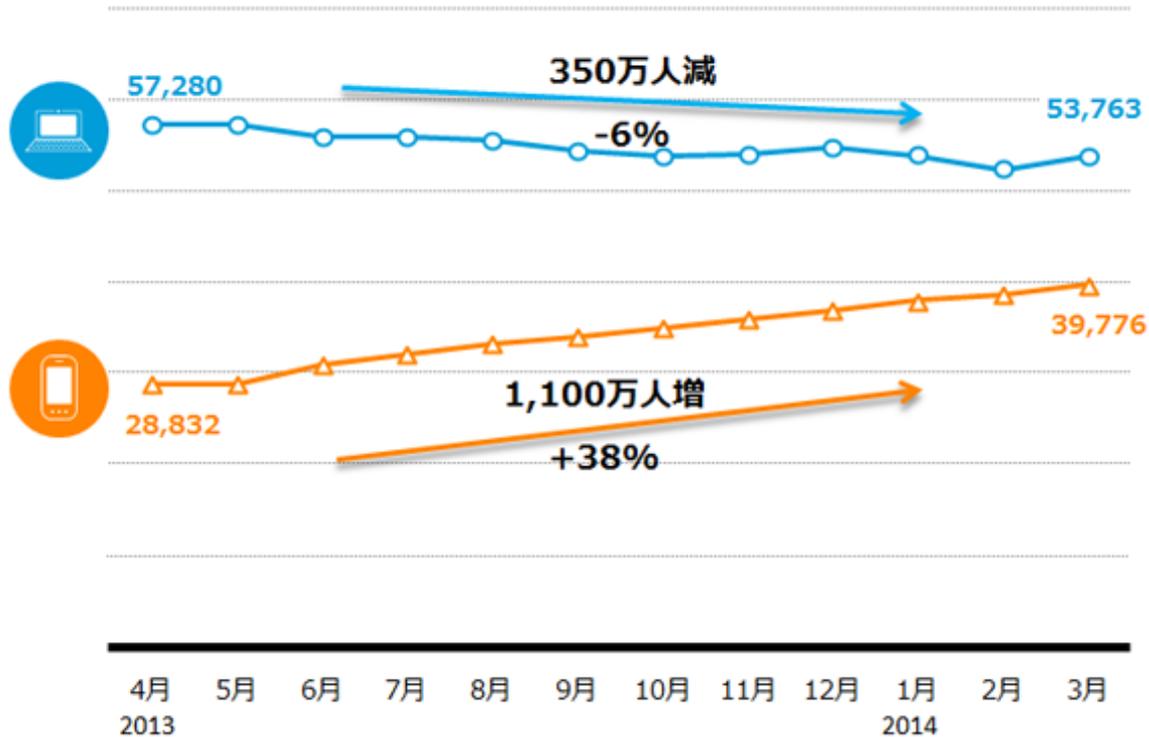
- Webの基礎知識
- HTML要素
- CSS3
- レスポンシブWebデザイン
- オフラインWebアプリケーション

HTML5 について



- 2014年正式勧告予定
- マルチデバイス・マルチメディア対応
- リッチクライアント・アプリケーションのプラットフォーム
- 広義ではCSS3やJavaScriptによる3Dグラフィック、WebSocket、デバイスアクセス、クライアントストレージ等も含む

インターネット利用者数推移



ニールセン株式会社 2013年度（2013年4月～2014年3月）のネット利用動向



ブラウザからマルチプラットフォームへ

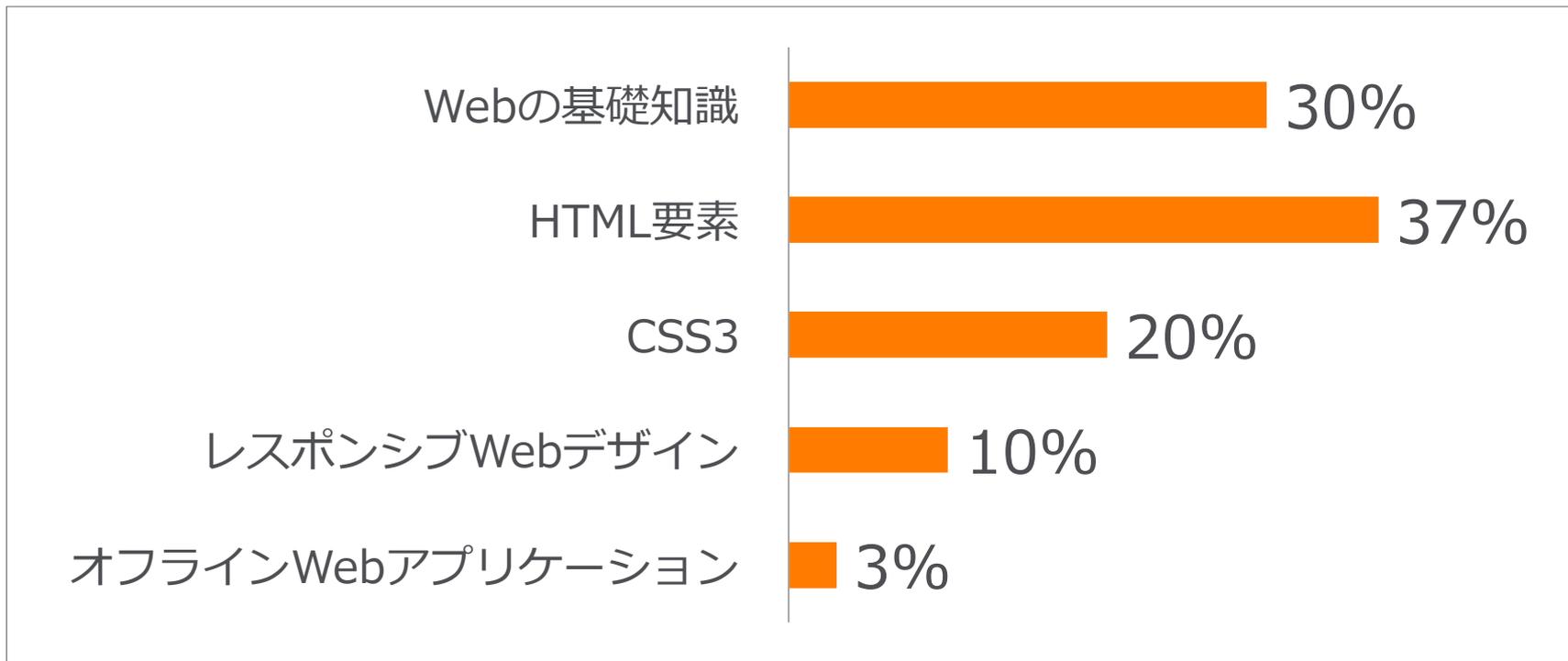
HTML5プロフェッショナル認定試験



HTML5プロフェッショナル認定試験とは

- 特定非営利活動法人LPI-Japanが実施する、HTML5および周辺技術の知識レベルを測る認定制度です。
- 試験の難易度を示す2種類のレベルがあり、段階的に受験します。
 - Level1
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、作成できるレベル
 - Level2
システム間連携や最新のマルチメディア術に対応したWebアプリケーションや動的Webコンテンツの開発・設計ができるレベル

Level 1 の試験範囲



※ 出題率は目安であり、実際の試験では変動します。

- 試験方式はコンピュータベーステスト（CBT）です。試験配信会社の「ピアソンVUE」を通して受験します。
 - 全国のテストセンターで通年受験可能
 - 合否結果はその場でわかる
 - 試験の詳細は以下の通り

問題数	約65問
試験時間	90分
合格ライン	約7割
回答方式	殆どが選択式（複数回答あり） 記述式も1問程度
受験料	¥15,000（税抜）

サンプル問題

■ HTML 5 で廃止されたタグは以下のうちどれですか？

- A : ``
- B : ``
- C : `<big>`
- D : `<small>`

■ input要素のtype属性に指定できない値は以下のうちどれですか？

- A : tel
- B : url
- C : color
- D : address

Webの基礎知識

■ メソッドの種類

- GET/POST/HEAD/PUT/DELETE など

■ リクエストヘッダの種類

- User-Agent/Referer/Content-Type など

■ ステータスコードの種類

- 1xx 情報
- 2xx 成功
- 3xx 転送
- 4xx クライアントエラー
- 5xx サーバーエラー

レイヤ	階層名	説明	例
7	アプリケーション層	アプリケーション間のデータ交換	HTTP
6	プレゼンテーション層	データ変換や圧縮	HTTP
5	セッション層	セッションの開始・終了	HTTP, SSL/TLS
4	トランスポート層	エラー訂正や再送処理	TCP, UDP
3	ネットワーク層	通信経路の選択・中継	IP, ICMP
2	データリンク層	直接接続されている機器間の通信	イーサネット
1	物理層	物理的な接続	光ケーブル

■ どのような脆弱性をついた攻撃なのか・サイト開発者はどのような対策が必要なのか、の2点を理解すること

- 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)
- HTTPヘッダインジェクション
- ディレクトリトラバーサル



SQLインジェクション

■ SQLインジェクションとは？

- データベースサーバに発行する問い合わせに、悪意のあるSQL（データの不正取得、削除など）を挿入する攻撃

■ 正常な問合せ



①一般ユーザーの操作

「ID」を表示

`http://shop.example.com/item.php?id=5`



RDB

②データベース側での処理

```
SELECT * FROM items WHERE id = '5';
```

■ 不正な問合せ



①攻撃者の操作
「ID」を表示

`http://shop.example.com/item.php?id=' OR 1 = 1;--`



②データベース側での処理

`SELECT * FROM items WHERE id = " OR 1 = 1;--";`

idが空文字か1==1（実質検索条件なし）、--以降は無効化

全データが抽出されてしまう！

■ 対策方法①

- 特殊文字をエスケープする
 - 開発言語が用意しているエスケープ関数などを利用する

■ 対策方法②

- あらかじめSQLを用意しておき、可変の部分だけを置き換える「プリペアドステートメント」という仕組みを利用する

■ XSSとは？

- 掲示板などの、ユーザーの登録した文字列が公開される場所にスクリプトを埋め込む攻撃
- 主な被害
 - 正規ユーザーのクッキー盗難
 - フィッシングサイトへの誘導
 - サイト改ざん
- 正式名称
 - Cross Site Scripting

■ XSSの例

掲示板

②スクリプトを含んだ文章が掲載される

```
<script>  
  location.href = "http://crack.example.com"  
</script>
```

①スクリプトを投稿



③閲覧



別のサイトに誘導
されてしまう！

■ 対策方法

- HTMLの生成時に、サニタイズ（タグの構成文字等の特殊文字をHTMLエンティティに変換すること）を行う
- HTMLエンティティ一覧

&	→	&
<	→	<
>	→	>
'	→	'
"	→	"

■ CSRFとは？

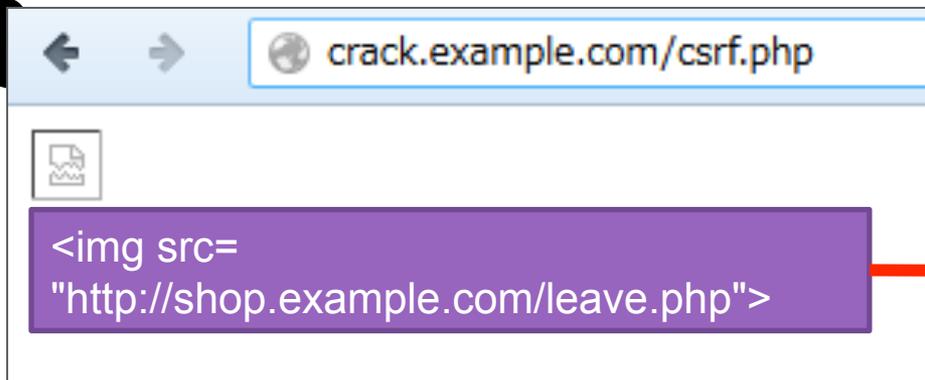
- ログイン済みユーザに対して、外部から不正操作を強要する攻撃
- 主な被害
 - 投稿処理
 - 退会処理
 - 登録情報の変更
- 正式名称
 - Cross Site Request Forgery

■ CSRFの仕組み

- ・ ログイン済みサイトに対して、img要素やJavaScriptによってリクエストを送信する



罨サイト



ログイン済みサイト

shop.example.com

leave.php
(退会URL)

正規のユーザーとして退会処理が実行されてしまう！

■ 対策方法

- パスワード入力によって本人の意思を確認する
- 直前のページにトークンを埋め込み、外部サイトからのアクセスではなく正しい導線に基づいた要求であるかを判断する

■ 間違った対策方法

- リファラーをチェックする
 - リファラーは偽装が可能である

■ ネットワーク・サーバ関連

- ・ ドメイン/DNS/プロキシ/ロードバランサ/ファイアーウォール

■ データベース

- ・ SQLの種類 (CREATE/DROP/SELECT/INSERT/DELETE/UPDATE)

■ データ表現技術

- ・ マイクロデータ/RDFa/データURL

■ Web広告

- ・ ペイパークリック/アドワーズ/アフィリエイト/コンバージョンレート

HTML要素

■ HTML5の例

```
<!DOCTYPE html>
<html lang="ja">
  <head>
    <meta charset="UTF-8">
    <title>サンプル</title>
  </head>
  <body>
    <p>本文</p>
  </body>
</html>
```

- 省略できる属性が増え、シンプルなシンタックスに
- セマンティックWeb対応（見た目ではなく、意味を重要視する）

■ DOCTYPE宣言

```
<!DOCTYPE html>
```

■ 文字エンコーディング指定

- 文書の先頭にBOMを付加するか、以下のどちらかを<head>内に指定

```
<meta charset="UTF-8">
```

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

セクショニング要素 (HTML5で追加)

header (ヘッダ)

nav (ナビゲーションリンク)

section (汎用的なセクション)

article (記事として独立したセクション)

article (記事として独立したセクション)

aside
(サイドバーや
広告など)

footer (フッタ)

■ article要素とsection要素

- 記事として独立しているセクションは<article>、それ以外の汎用的なセクションは<section>
- <article>、<section>要素はどちらが親要素となっても構わない

■ header要素とfooter要素

- <header>、<footer>は<article>、<section>の中に入れることもできる
- <header>内には通常<h1>~<h6>を含むが、含まなくても構わない

figure/figcaption要素（HTML5で追加）

■ 説明文のついた図表などを表す

- 要素は本文から参照されるが、本文とは別のページなどに切り離すことができる

```
<p>クランウェルツノガエルは、ツノガエル科の蛙です。<br>同科のベルツノ  
ガエルと比較するとツノが尖っているのが特徴です。<br><a href="#photo">  
クランウェルツノガエルの画像</a></p>
```

```
<figure id="photo">  
    
  <figcaption>クランウェルツノガエル</figcaption>  
</figure>
```

video/audio/source要素（HTML5で追加）

■ 動画（video）、音声（audio）の再生

- どちらの要素も使い方はほぼ同じ

```
<video src="video.mp4" controls autoplay loop></video>
```

```
<audio src="audio.mp3" controls autoplay loop></audio>
```

- クロスブラウザ対応するには<source>要素を子要素に使う

```
<audio controls autoplay loop>
```

```
<!--ブラウザが対応しているファイルタイプに適合するファイルを使用-->
```

```
<source src="audio.ogg" type="audio/ogg" media="all">
```

```
<source src="audio.mp3" type="audio/mp3" media="all">
```

```
<p>ご利用中のブラウザでは再生できません</p>
```

```
</audio>
```

■ メディアファイルに字幕をつける

- WebVTT形式で作成したテキストトラックを、video要素などに埋め込む

```
<video src="video.mp4">  
  <track src="ja.vtt" srclang="ja" label="日本語" kind="subtitles"  
  default>  
  <track src="en.vtt" srclang="en" label="英語" kind="subtitles">  
</video>
```

WEBVTT

```
00:00:01.000 --> 00:00:05.000  
こんにちは
```

```
00:00:06.000 --> 00:00:11.000  
はじめまして
```

embed要素（HTML5で追加）

■ プラグインが必要なコンテンツの埋め込み

- 以前からブラウザに実装されていた要素だが、仕様として組み込まれたのはHTML5から
- QuickTimeプラグインが必要なファイルを埋め込む例

```
<embed src="file.mov">
```

- プラグイン不要な外部リソース（SVG、HTMLなど）を埋め込む場合は<object>要素を使う

```
<object src="image.svg"></object>
```

mark要素（HTML5で追加）

■ 重要箇所を表す

- 制作者ではなく、「ユーザーにとって」重要な箇所（検索結果など）をマークアップする
- 多くのブラウザでは黄色く表示される

Monacaは、`<mark>HTML5</mark>`によるモバイルアプリの開発環境です。

Monacaは、**HTML5**によるモバイルアプリの開発環境です。

progress/meter要素 (HTML5で追加)

■ <progress>

- ・ タスクの進捗状況を表す

```
<progress value="75" max="100">100%中75%まで完了</progress>
```

■ <meter>

- ・ 範囲内の数、量、割合などを表す

```
<meter value="75" max="100" min="0">100人中75人が回答</meter>
```

- ・ 見た目はどちらもほぼ同じ



■ <time>

- 日付や時刻を表す
- `datetime`属性（省略時は内容テキスト）をコンピュータが認識可能な形式で指定する

```
<time datetime="2001-05-15 19:00">May 15</time>
```

■ <data>

- 日付以外のデータを表す
- `value`属性をコンピュータが認識可能な形式で指定する

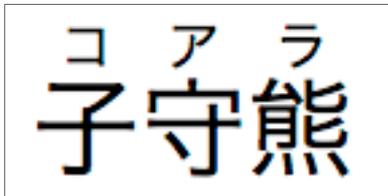
```
<data value="30">三十歳</data>
```

■ ルビを付与する

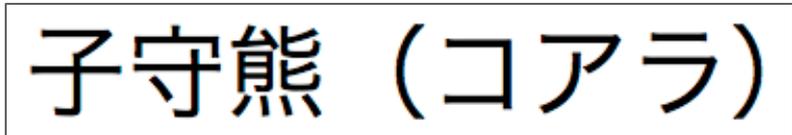
- `<ruby>` 対象テキストをマークアップ
- `<rt>` ルビテキストを指定
- `<rp>` ルビ未対応ブラウザでのみ表示

```
<ruby>子守熊<rp> (</rp><rt>コアラ</rt><rp>) </rp></ruby>
```

ルビ対応ブラウザ



ルビ未対応ブラウザ



bdi要素 (HTML5で追加)

■ 向きが異なるテキストを埋め込む

- アラビア文字などの、右から左に記述する自然言語をマークアップすることによって前後のテキストが入れ替わって配置される現象を回避する

```
<ul>
  <li>User Name: <bdi>james</bdi> 2014/5/28</li>
  <li>User Name: <bdi>steve</bdi> 2014/6/19</li>
  <li>User Name: <bdi>إيمان</bdi> 2014/7/20</li>
</ul>
```

■ テキストが改行されても良い位置を指定する

- 一般的なブラウザはスペースを含まないアルファベットや記号の並びは改行せずに表示する
- ブラウザの横幅が狭く文章を表示しきれない場合に、`<wbr>`が挿入されている箇所を改行する

```
section,nav,article,aside,header,footer,main,<wbr>figure,figcaption,  
video,audio,source,canvas,mark,time,data,ruby,rt,rp,rb,bdi,progress,  
meter,output,datalist,keygen
```

canvas要素（HTML5で追加）

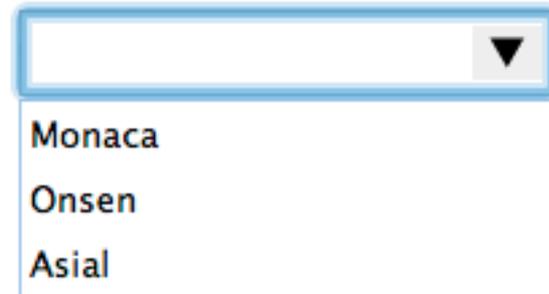
- JavaScriptでビットマップのグラフィックを描画する要素
 - 具体的な描画方法はLevel2の範囲
 - Level1ではキャンバスで描画できる図形の種類を知っておくこと
- 四角形/多角形/円弧/画像

datalist要素 (HTML5で追加)

■ フォームの入力候補を定義する

- 定義した入力候補はテキストボックスなどに付与することができる

```
<datalist id="keywords">  
  <option value="Monaca">  
  <option value="Onsen">  
  <option value="Asial">  
</datalist>  
  
<input type="text" list="keywords">
```



keygen要素（HTML5で追加）

■ 公開鍵と秘密鍵を生成する

- フォーム送信時に公開鍵と秘密鍵を生成する
- 秘密鍵はローカルに保存され、公開鍵のみがサーバーに送信される
- サーバーサイドでクライアント証明書を生成するなど、クライアントの信頼性を確保する目的での用途が考えられる

```
<form action="keygen.php" method="post">  
  <keygen name="key" keytype="rsa">  
  <input type="submit" value="キーを送信する">  
</form>
```

output要素（HTML5で追加）

■ スクリプトによる計算結果などを表示する

- ユーザーに対して表示することのみを目的とする（フォーム送信されない）

```
<form oninput="result.value = parseInt(price.value * 1.08)">
  <input type="number" name="price" value="0">
  税込：<output name="result">0</output>円
</form>
```

 税込：1080円

HTML5で削除された要素

要素名	説明
basefont	基準となるフォント、サイズ、色を指定する
big	テキストサイズを一回り大きくする
center	コンテンツを中央に配置する
font	テキストのフォント、サイズ、色を指定する
strike	取り消し線を引く
tt	等幅フォントを指定する
frame	フレーム分割されたウィンドウに表示するページを指定する
frameset	ウィンドウをフレーム分割する
noframes	frame要素が利用できないブラウザでの表示内容を記述する

HTML5で**変更**された要素

要素名	変更前	変更後
b	太字表示	他と区別したいテキストを表す
i	斜体表示	専門用語などを表す
s	取り消し線	もう正確でないテキストを表す
u	下線	伝わりにくい、または本来と異なる表記を表す
hr	罫線	意味的な段落を表す
small	テキストを小さく表示する	注釈や細目を表す
strong	テキストを強調表示する	重要性を表す

■ バリデーション属性

- `<input>`要素に付与すると、submit時にチェックを行ってくれる
 - `required` 必須
 - `pattern` 正規表現
 - `min` 最小値
 - `max` 最大値
 - `maxlength` 最長文字数

名前：`<input type="text" required>`

郵便番号：`<input type="text" pattern="^[0-9]{3}-[0-9]{4}$">`

年齢：`<input type="number" min="18" max="99">`

ID：`<input type="text" maxlength="6">`

■ input type属性値

- urlやemailを指定した場合、フォーム送信時にURL、メールアドレスの形式として正しいかバリデーションが行われる
- rangeを指定するとレンジバーが、colorを指定するとカラーパレットが表示されるなど、UIの拡張が行われる

```
URL : <input type="url">  
メールアドレス : <input type="email">  
電話番号 : <input type="tel">  
日付 : <input type="date">  
範囲 : <input type="range">  
色 : <input type="color">
```

■ 独自データ属性

- スクリプトで利用する値を要素に保持するために、data-で始まる属性を自由に定義することができる

```
<ul>
  <li data-id="0001">りんご</li>
  <li data-id="0002">みかん</li>
  <li data-id="0003">いちご</li>
</ul>
```

CSS3

■ セレクト、疑似要素、疑似クラスの種類

- CSS2も試験範囲に含まれる
- idセレクト/要素型セレクト/クラスセレクト/属性セレクト
- 子セレクト/子孫セレクト/隣接セレクト/間接セレクト
- 疑似要素 (:before/:after/:first-lineなど)
- 疑似クラス (:first-child/:active/:link/:hover/:target/:notなど)

■ スタイルの継承

- スタイルは基本的に子孫要素に継承される
- 一部の継承されないプロパティを継承するには、`inherit`を指定する

```
div {  
    border : solid 1px #fff;  
}  
  
div * {  
    border: inherit;    /* div要素のborderを子孫要素に継承 */  
}
```

カスケード（優先順位）

■ スタイルが競合している場合の優先順位の決定

1. CSSの種類

!important付きユーザCSS > !important付き文書作成者CSS
> 文書作成者CSS > ユーザCSS > ブラウザ標準CSS

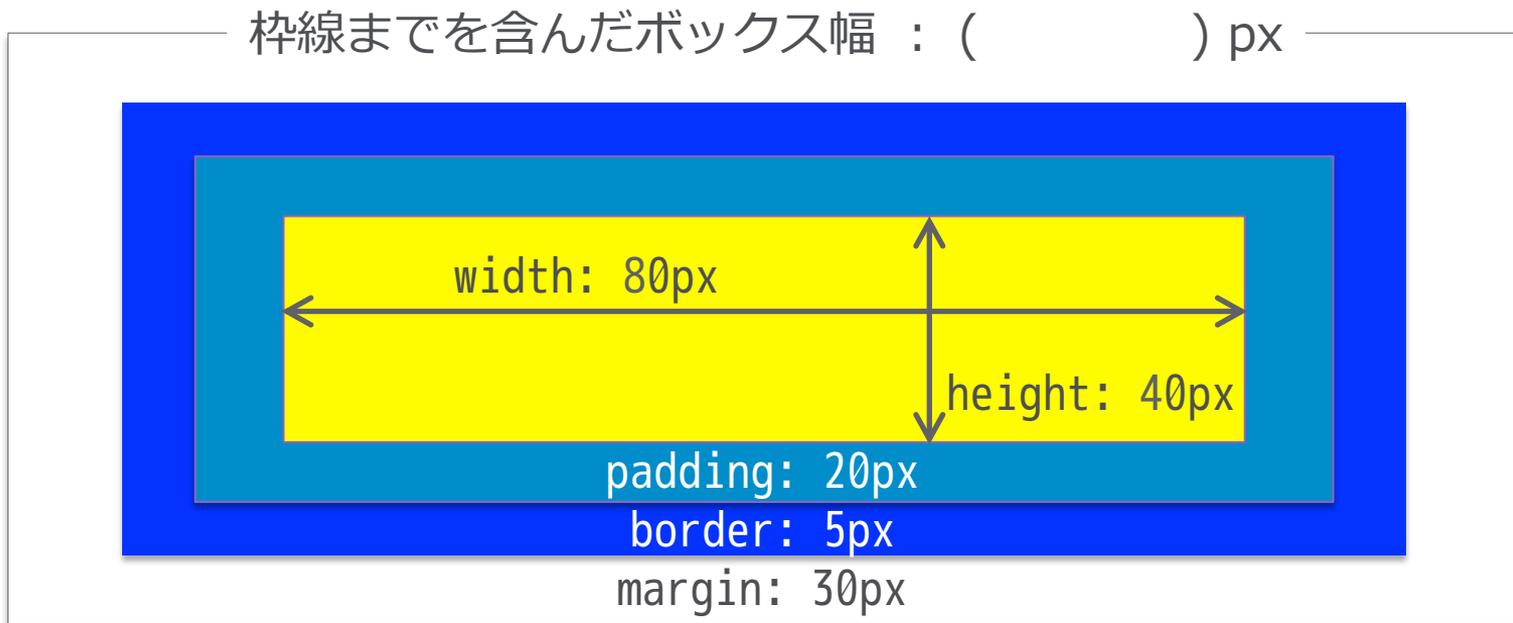
2. セレクタの種類と数による重み付け

1. style属性で指定したスタイル
2. idセレクタの数が多いものを優先
3. クラスセレクタ or 属性セレクタ or 疑似クラス の数が多いものを優先
4. 要素型セレクタ or 疑似要素 の数が多いものを優先

3. 出現順

1. 最後に出現する記述を優先

- ボックスは コンテンツ領域 (width,height) 、内側の余白 (padding) 枠線 (border) 、外側の余白 (margin) によって構成される



■ CSS3での変更点

- **box-sizing**プロパティによって、**width**、**height**の指定値を**border**まで含めた値にすることが可能
 - **box-sizing : content-box** 従来通り
 - **box-sizing : border-box** borderまでを幅と高さを含める

■ border/paddingプロパティの値

- 値が1つ：上下左右すべて同じ値

```
margin: 10px;
```

- 値が2つ：上下 と 左右

```
margin: 10px 20px; /* 上下10px, 左右20px */
```

- 値が3つ：上 左右 下

```
margin: 10px 20px 30px; /* 上10px, 左右20px, 下30px */
```

- 値が4つ：上、右、下、左（時計回りと覚える）

```
margin: 10px 20px 30px 40px; /* 上10px, 右20px, 下30px, 左40px */
```

■ マルチカラムレイアウト（段組みレイアウト）

- 長い文章などを指定したカラム数に分割して表示することができる

マルチカラムレイアウトは、長い文章などを指定したカラム数

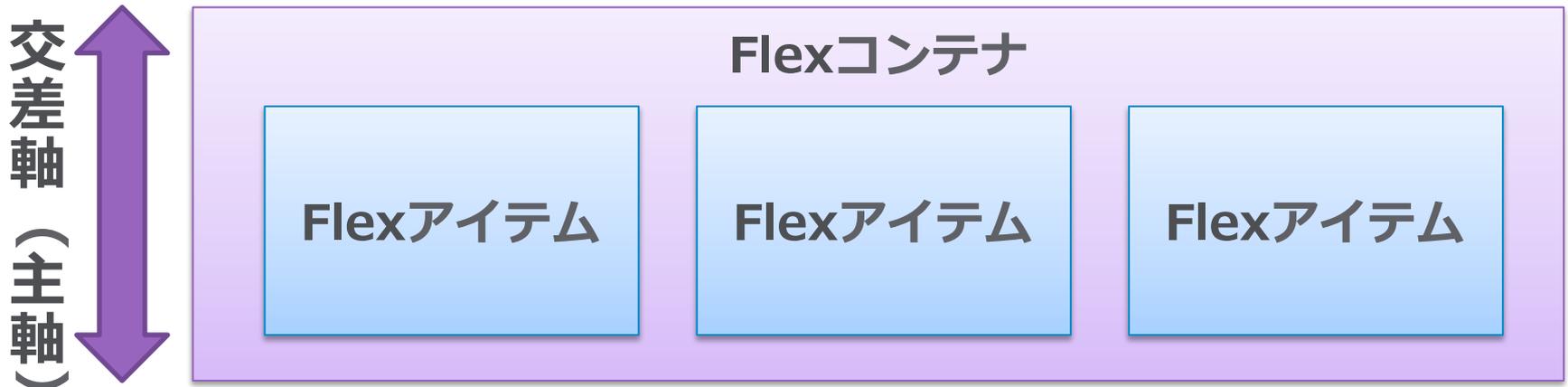
に分割して表示することができます。これは3つのカラムに分

割したイメージです。

■ Flexible Box (可変ボックスレイアウト)

- 要素の中央揃えや均等揃えなどの配置を簡単に行うことができる

主軸 (交差軸)



■ transitionプロパティ

- スタイルを変更した際に、変化の過程をアニメーションで表示する
 - 例：要素にマウスカーソルが乗ったら、背景色を赤から青に変化させる

■ animationプロパティ

- transitionとの違いは、アニメーションを連続で実行することができる
 - 例：要素にマウスカーソルが乗ったら、背景色を赤→青→緑→黄色のように変化させる

■ transformプロパティ

- 要素に対して移動、回転、拡大、傾斜を行うことができる

```
img {  
  transform: rotate(90deg);    /* 画像を90度回転 */  
}
```



CSS適用



■ ベンダプリフィックス

- -webkit- / -moz- / -o- / -ms-

■ ボックスモデル

- 表示の制御 (display/visibility)
- 影 (box-shadow) / 切り抜き (clip) / 角丸(border-radius)

■ 色

- カラーコード / カラーネーム / rgb / rgba

■ フォント

- Webフォント / レイアウト (単語間隔や行高さなど) / 装飾 (色や影)

レスポンスイブWebデザイン

■ メディアクエリ

- デバイスの特性（ブラウザ幅など）に応じてCSSを切り替える方法

横幅800px
未満



横幅800px
以上



メディアクエリの指定方法

1. CSSファイル内でスタイルを指定

```
@media screen and (min-width: 800px) {  
  body{  
    background-color: #000  
  }  
}
```

2. <link>でCSSファイルを読み込むときに指定

```
<link rel="stylesheet" media="screen and (min-width: 800px)"  
href="style.css">
```

■ フルードデザイン、フルードイメージ

- 画面の幅に合わせて横幅を調整する
- コンテンツのサイズを%単位で指定する

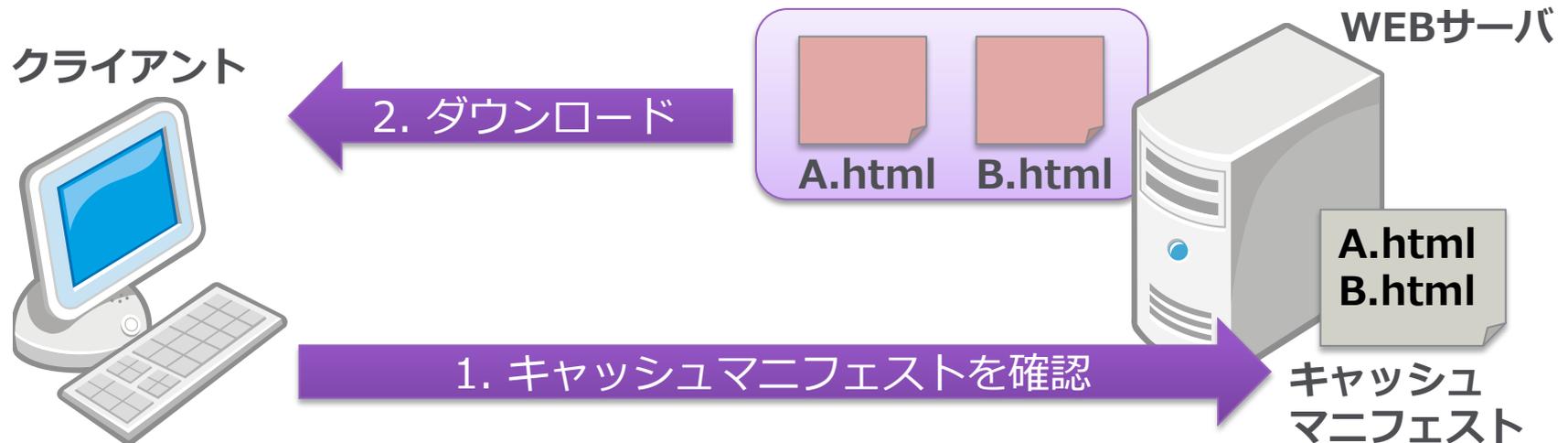
■ グリッドレイアウト

- ページをグリッドで分割し、グリッド枠に沿ってコンテンツを配置
- 画面サイズにあわせてグリッドの幅や数を調整する
- グリッドレイアウトを実現するための代表的なフレームワークに
Bootstrapなどがある

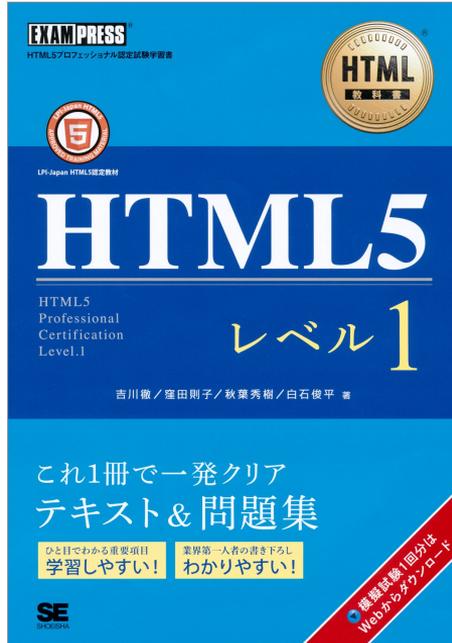
オフラインWebアプリケーション

■ アプリケーションキャッシュ

- ・ キャッシュマニフェストに記述されたファイルをローカル環境に保存し、オフライン状態ではローカル環境に保存されたファイルを表示する



受験対策



HTML教科書
HTML5レベル1



HTML5プロフェッショナル
認定資格 レベル1 教科書

LPI-JAPAN HTML5 Professional Certification

Open the Future with **HTML5**.