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

株式会社クリーク・アンド・リバー社 認定講師

高井 歩

本日の概要

- HTML5 レベル1 試験について
- レスポンシブWebデザイン
- マルチデバイス対応ページの作成
- メディアクエリ
- スマートフォンサイト最適化

HTML5プロフェッショナル認定資格とは

- 次世代のWebプロフェッショナルのスキルの向上に貢献するために、HTML5を活用したWebページやWebアプリケーションなどのデザイン、設計、構築に関する体系だった知識とスキルを備えたHTML5のプロフェッショナルを中立的な立場で公平かつ厳正に認定する資格制度です。

- Webデザイナー、Webプログラマー、スマートフォンアプリ開発者、サーバーサイドエンジニアなどの、Web開発プロジェクトやWebサービスに関わるあらゆるプロフェッショナルが対象です。

- 多くの企業が推進する次世代Web言語の認定資格として、HTML5のプロフェッショナルのスキルの向上に役立ちます。
また、企業内や研修機関での『技術力を担保する客観的基準』としても活用できます。

二つのレベル

HTML5 Level.1

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

対象

- Webデザイナー
- グラフィックデザイナー
- フロントエンドプログラマー
- HTMLコーダー
- Webディレクター
- Webシステム開発者
- スマートフォンアプリ開発者
- サーバーサイドエンジニア

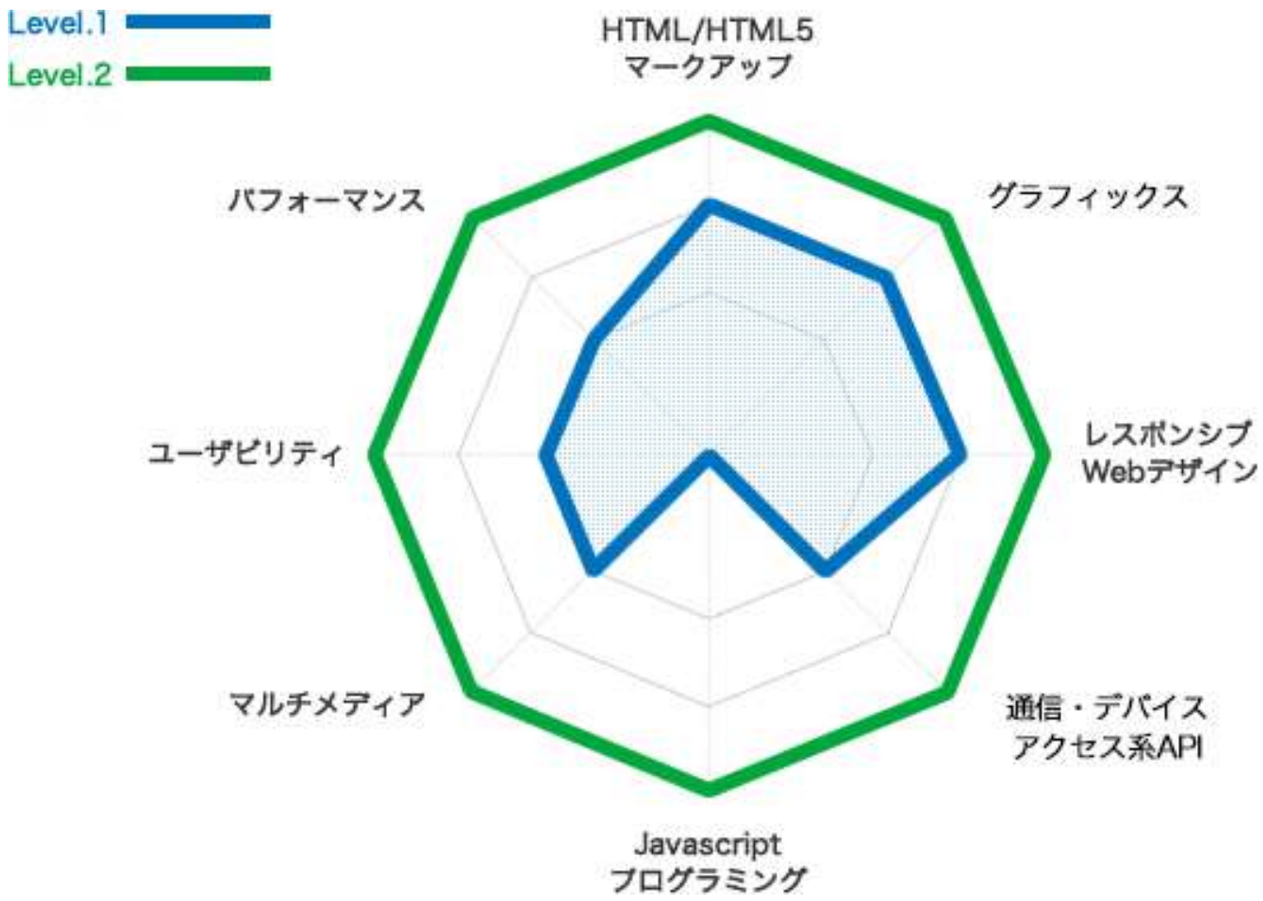
HTML5 Level.2

最新のAPIを駆使したWebアプリケーションを設計・開発できる。

対象

- Webデザイナー
- フロントエンドプログラマー
- HTMLコーダー
- Webディレクター
- Webシステム開発者
- スマートフォンアプリ開発者
- サーバーサイドエンジニア

レベル1とレベル2の資格体系



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

レスポンシブ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

Javascriptプログラミング

Javascriptを使って、動的なWebコンテンツを作成する技術

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良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

レベル1とレベル2の資格体系

ベーシックレベル
HTML5プロフェッショナル向け

所要時間：90分
試験問題数：約60問
受験料：￥15,000 （税別）
認定条件： HTML5 レベル1試験に合格すること
認定の有意性の期限：5年間

認定名： HTML5 Level1 (Markup Professional)
試験名： HTML5 Level1 Exam

この資格の認定者は、下記のスキルと知識を持つWebプロフェッショナルであることを証明できます。

- **HTML5を使ってWebコンテンツを作成することができる。**
- **ユーザー体験を考慮したWEBコンテンツを設計・作成することができる。**
- **スマートフォンや組み込み機器など、ブラウザが利用可能な様々なデバイスに対応したコンテンツを制作できる。**
- **HTML5で何ができるか、どういった技術を使うべきかの広範囲の基礎知識を有する。**

アドバンストレベル
HTML5プロフェッショナル向け

所要時間：90分
試験問題数：40〜45問
受験料：￥15,000(税別)
認定条件：HTML5 レベル2試験に合格し、かつ有意なHTML5レベル1認定を保有していること。
認定の有意性の期限：5年間

認定名： HTML5 Level2 (Application Development Professional)
試験名： HTML5 Level2 Exam

この資格の認定者は、下記のスキルと知識を持つWebプロフェッショナルであることを証明できます。

- **動的に動作させて高いユーザビリティを実現するリッチユーザインターフェイスアプリケーションを作成することができる。**
- **マルチデバイスに対応し高パフォーマンスで動作する動的コンテンツを作成することができる。**
- **システム間連携を行いリアルタイムな情報を提供するアプリケーションを作成することがでる。**
- **スマートフォンなどでネイティブアプリに近い機能を組み込んだ先端のWebアプリケーションに近い機能を組み込んだ先端のWebアプリケーションを開発することができる。**
- **APIのセキュリティモデルを理解したうえで開発することができる。**

出題範囲

主題	項目
Webの基礎知識	・HTTP,HTTPSプロトコル(☆8) ・HTMLの書式(☆9) ・Web関連技術の概要(☆6)
CSS	・スタイルシートの基本(☆7) ・CSSデザイン(☆9) ・カスケード（優先順位）(☆2)
要素	・要素と属性の意味(☆10) ・メディア要素(☆6) ・インタラクティブ要素(☆7)
レスポンシブWebデザイン	**・マルチデバイス対応ページの作成(☆4)** **・メディアクエリ(☆5)** **・スマートフォンサイト最適化(☆3)**
APIの基礎知識	・マルチメディア・グラフィックス系API概要(☆5) ・デバイスアクセス系API概要(☆4) ・オフライン・ストレージ系API概要(☆8) ・通信系API概要(☆3)

レスポンシブWebデザイン

レスポンシブWebデザイン

スクリーンサイズに応じてレイアウト、デザインが変更されるWebページのデザイン

max-width:1080px　　max-width:768px

岩波ホール：https://www.iwanami-hall.com

マルチデバイス対応ページの作成

知識問題 コードリーティング問題 記述問題

マルチデバイス対応ページの作成

- リセットCSS
- レイアウト
  - 固定レイアウト
  - 可変レイアウト
    - Fluid Grid
- Fluid Image

viewportは
「スマートフォンサイト最適化」
で解説

リセットCSS

- 各Webブラウザが持つ**デフォルトのCSSをリセット**するためのCSS

- marginやpadding、list-styleなどのスタイルを除去するものが多い
  (残しつつ共通化するものはノーマライズCSSと呼ばれる)

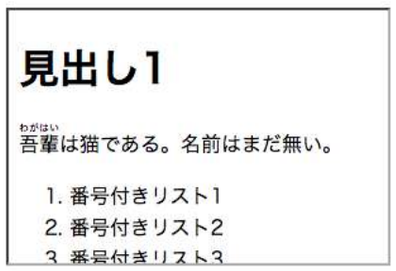
- 様々なリセットCSSが公開されている。
  - Eric Meyer’s “Reset CSS” 2.0
  - HTML5 Doctor CSS Reset
  - Yahoo! (YUI 3) Reset CSS

  など(上記3つはhttp://cssreset.comの集計による2016年DL数top3)

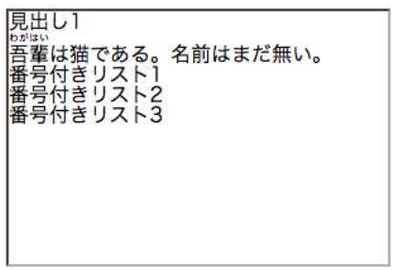
リセットCSSの例

Firefox

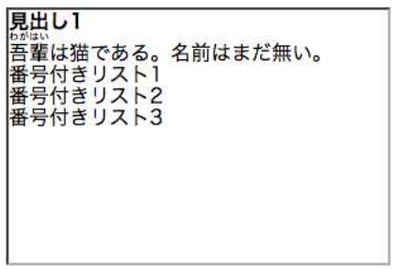
CSSなし(ユーザエージェントスタイルシート)



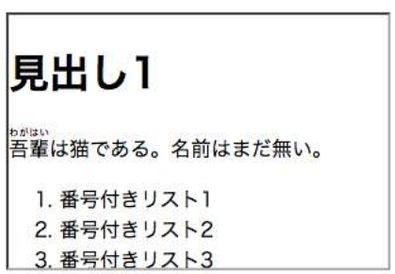
Eric Meyer's "Reset CSS" 2.0



HTML5 Doctor CSS Reset

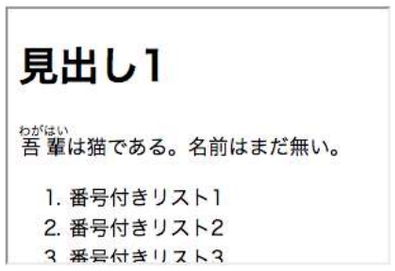


sanitize.css v4.0.0

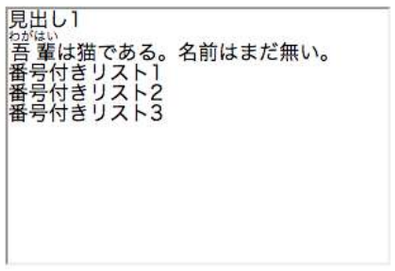


Chrome

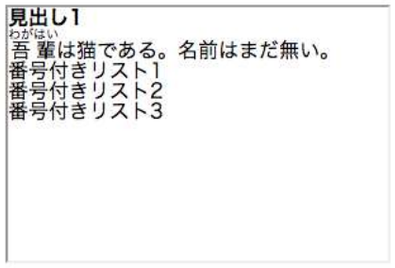
CSSなし(ユーザエージェントスタイルシート)



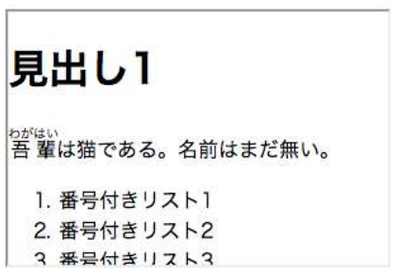
Eric Meyer's "Reset CSS" 2.0



HTML5 Doctor CSS Reset

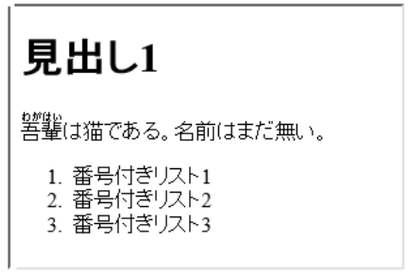


sanitize.css v4.0.0

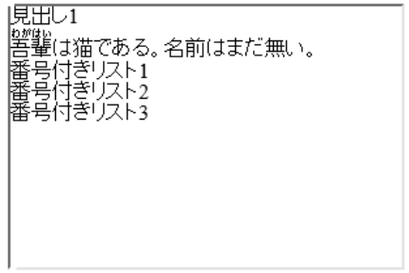


IE11

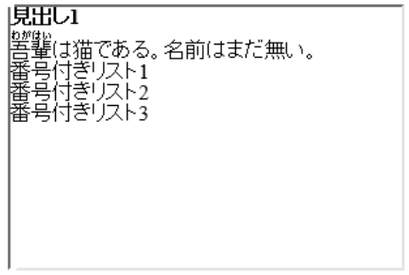
CSSなし(ユーザエージェントスタイルシート)



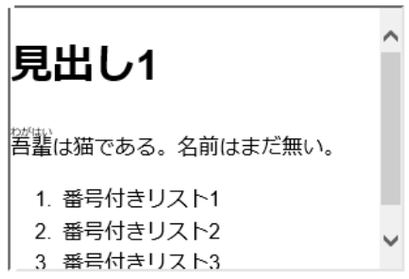
Eric Meyer's "Reset CSS" 2.0



HTML5 Doctor CSS Reset



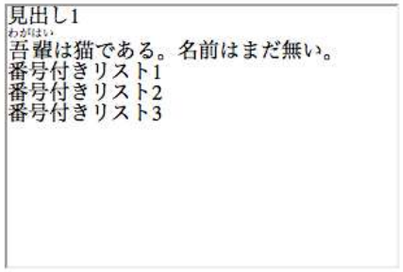
sanitize.css v4.0.0



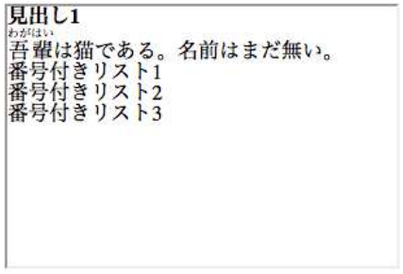
Safari

CSSなし(ユーザエージェントスタイルシート)

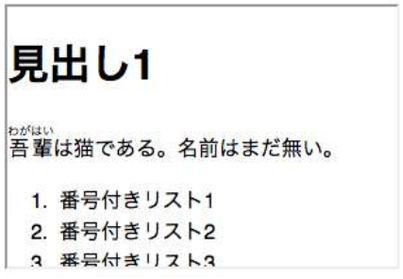
Eric Meyer's "Reset CSS" 2.0



HTML5 Doctor CSS Reset



sanitize.css v4.0.0



練習問題1

リセットCSSの説明として正しくないものを選択しなさい。

A. Webブラウザ間の差異を吸収する

B. ユーザエージェントスタイルシートをリセットする

C. 既存のスタイルをできるだけ残す

D. リセットCSSの後で独自のスタイルをセットする

レイアウト

マルチデバイス対応時のページレイアウトの課題

- 物理的な画面サイズの違い
  - モバイル端末の縦(portrait)横(landscape)
- 画面解像度の違い
- マウス、タッチなど操作方法の違い

固定レイアウト

- レイアウトが固定で、画面サイズが小さくなるとスクロールバーが表示される。

- サイズ指定、位置指定はpx単位で行なう。

スクロールバーが表示される

財務省:http://www.mof.go.jp/index.htm

可変レイアウト

- 固定レイアウトに対して、表示領域のサイズに応じてコンテンツやレイアウトが変化するWebサイトの作りかた。

- **レスポンシブWebデザイン**や**FluidGrid**など様々な手法、概念がある。

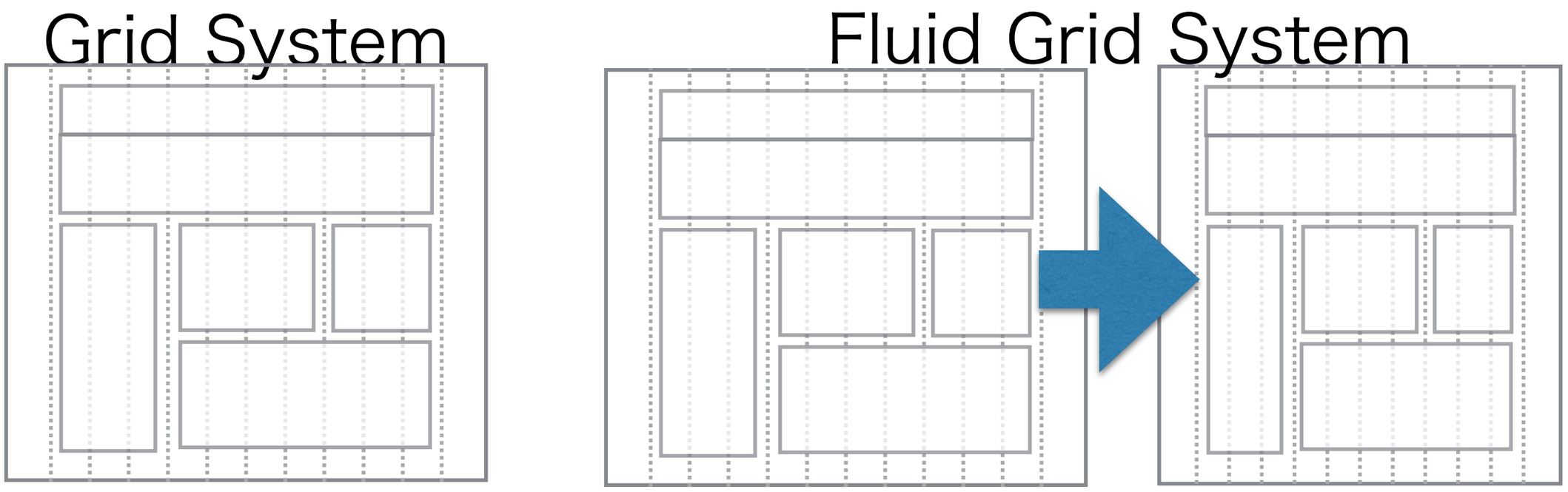
- サイズ指定、位置指定は%やem単位のような**相対値**で行なうことが多い。

- **メディアクエリ**を使用して**CSSを切り替える**こともある。

Fluid Grid

- 表示領域を格子状に区切り、その格子に合せてコンテンツなどを配置していく手法をグリッドシステムと呼ぶ。(グリッドシステムは固定/可変どちらも可能)

- グリッドシステムをウィンドウサイズなどからの相対値にすることで、デバイスに応じたレイアウトにする手法がFluid Grid。



Fluid Image

- 通常、画像のサイズは表示領域の大きさにかかわらず一定。

- 画像のサイズも表示領域の大きさの相対値にすることで、画像だけ(相対的に)大きく表示されたりする問題を防ぐ手法がFluid Image。

- 以下のようにmax-widthを指定することで親要素のサイズを越えないようにできる。

```
img {
  max-width: 100%;
}
```

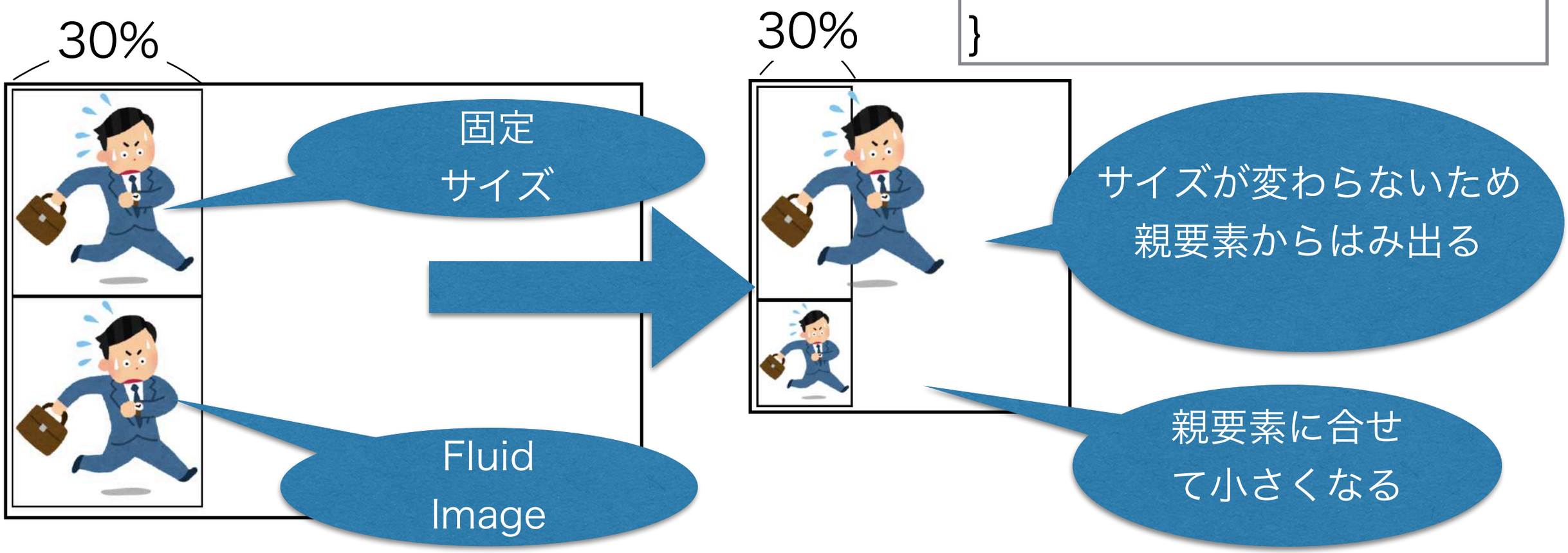
Fluid Imageの例

HTML

```
<div class="nonfluid">
  <img src="images/img.png" width="300">
</div>
<div class="fluid">
  <img src="images/img.png">
</div>
```

CSS

```
div {
  border:solid 3px black;
  width:30%;
}
.fluid img {
  max-width: 100%;
}
```



HTML5.1の画像の取り扱い 試験範囲外

- **HTML5.1 は 2016.11.1に勧告**された。

- picture要素

```
<picture>
 <source srcset="largeLogo.png" media="(min-width: 600px)">
 <img src="smallLogo.png" alt="Logo">
</picture>
```

- img要素のsrcset属性

- 使い分けの指針
  - サイズ(解像度)変更なら srcset属性
  - 内容やレイアウトの変更なら picture要素

練習問題2

Fluid Gridの特徴として正しくないものを2つ選択しなさい。

A. Gridの幅を相対値で指定する。

B. 同サイズのタイル状のパネルで配置する。

C. Gridの幅をpxで指定する。

D. 表示領域を格子状に区切る。

メディアクエリ

知識問題 コードリーティング問題 記述問題

メディアクエリ

- メディアクエリ
- メディアタイプ
- メディア特性
- 単位(ピクセル,dpi,dpcm)

メディアクエリ

- デバイスの特性(スクリーンの大きさや種類)に応じてCSSを切り替える技術。
- CSSを読み込むlink要素のmedia属性または、CSS内で@media規則で指定。
- **メディアタイプ**と**メディア特性**を組み合わせて適用条件を指定する。

media属性での指定例

<link href="(URL)" rel="stylesheet" **media="(メディアクエリ)"**>

@media規則での指定例

@media (メディアクエリ) {　body { font-size: 10pt } }

メディアタイプ

- 出力デバイスの種類を条件にする

メディアタイプ	概要
all	すべてのデバイスに適合
print	印刷、印刷プレビュー
screen	カラーのコンピュータ画面
speech	スピーチシンセサイザ (読み上げソフト)

Media Queries Level 4(ドラフト)で非推奨

tty	固定幅の表示端末
tv	低解像度なテレビ
projection	プロジェクタ
handheld	携帯デバイス
braille	点字ディスプレイ
embossed	点字印刷
aural	音声出力

メディア特性

・幅や高さなどの量や値を条件にすることができる。

特性	概要	特性	概要
width	表示領域の幅	device-aspect-ratio	出力領域の縦横比
height	表示領域の高さ	color	色要素毎のビット数
device-width	出力領域の幅	color-index	使用可能パレット数
device-height	出力領域の高さ	monochrome	モノクロのビット数
orientation	縦/横表示	resolution	デバイスの解像度
aspect-ratio	表示領域の縦横比	scan	テレビの操作方式
		grid	文字表示/画像表示

はmin/max接頭辞を付けることができる(min-widthなど)

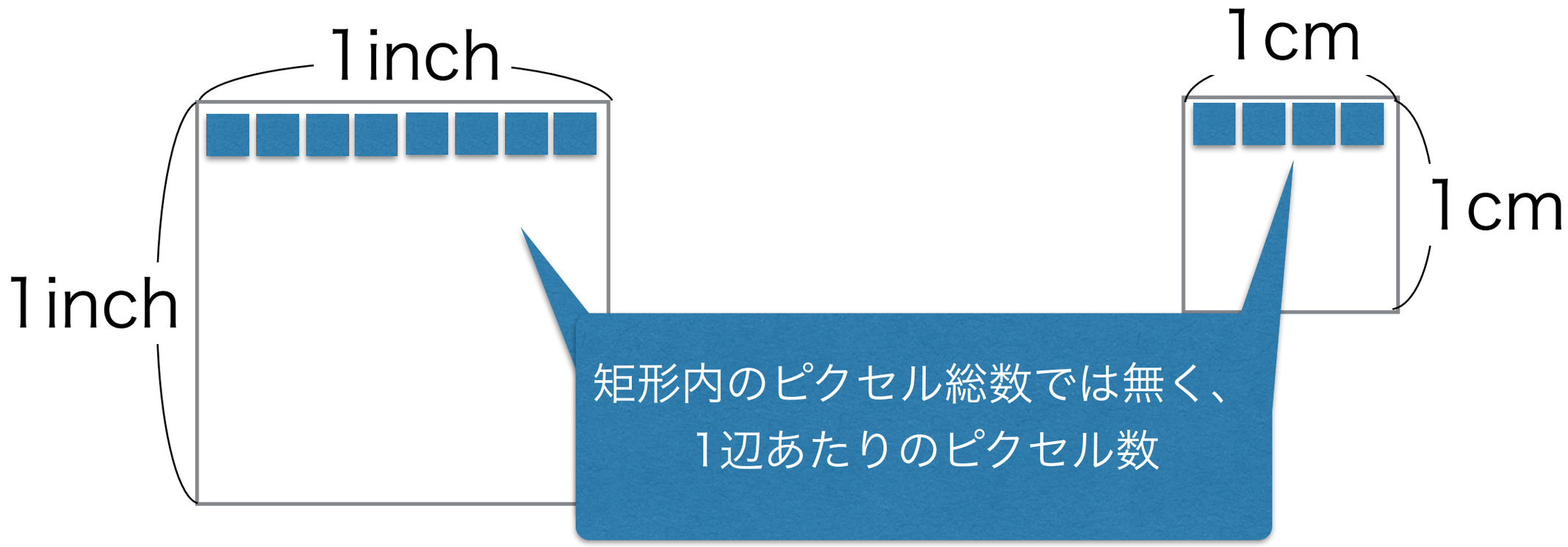
メディアクエリで使える演算子

- 複数のメディアタイプやメディア特性の条件を結合してより複雑な条件を表現する

演算子	概要
and	二つのクエリが共にtrueの時true
not	クエリの結果を否定
only	メディアクエリ非対応ブラウザから スタイルシートを隠す
,	or に相当 二つのクエリの少なくとも片方がtrueならtrue

単位(ピクセル,dpi,dpcm)

- px … 表示画面の点ひとつ、画素、ドット
- dpi … Dots Per Inch 1インチあたりの画素数
- dpcm … Dots Per CentiMeter 1cmあたりの画素数



メディアクエリの例

幅800px**以下**

(max-width: 800px)

max-,min-はどちらもその値を含む。
max-width:800pxは800px以下、
min-width:800pxは800px以上。

幅700px以上**かつ**横画面

(min-width: 700px) **and** (orientation: landscape)

幅700px以上**もしくは**横画面

(min-width: 700px) , (orientation: landscape)

全てのモノクロデバイス(全てのデバイス**かつ**カラーでは**無い**)

all **and** (**not** color)

練習問題3

メディアタイプとして正しく無いものを選択しなさい。

A. screen

B. smartphone

C. print

D. speech

スマートフォンサイト最適化

知識問題 コードリーティング問題 記述問題

スマートフォンサイト最適化

- CSSスプライト
- 高解像度画面向け対応
- viewport,density,initial-scale
- ファビコン、アイコン設定
- スタンドアローンモード
- 電話番号へのリンク
- script要素のasync属性、defer属性

CSSスプライト

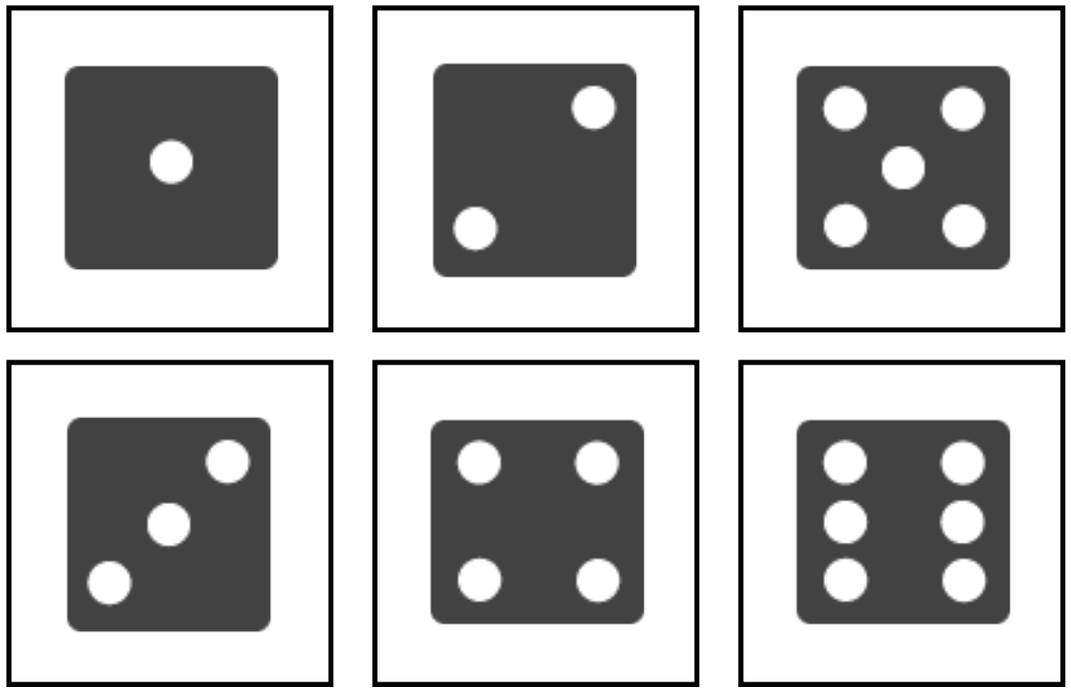
- 複数の小さなアイコンなどを**ひとつの画像にまとめる**ことで、**ファイルサイズを小さく、ファイルの転送回数を減らす**技法。

- 一般に**background-image**として表示させる。

- それぞれの画像は**background-position**と**width,height**を指定して表示させる。

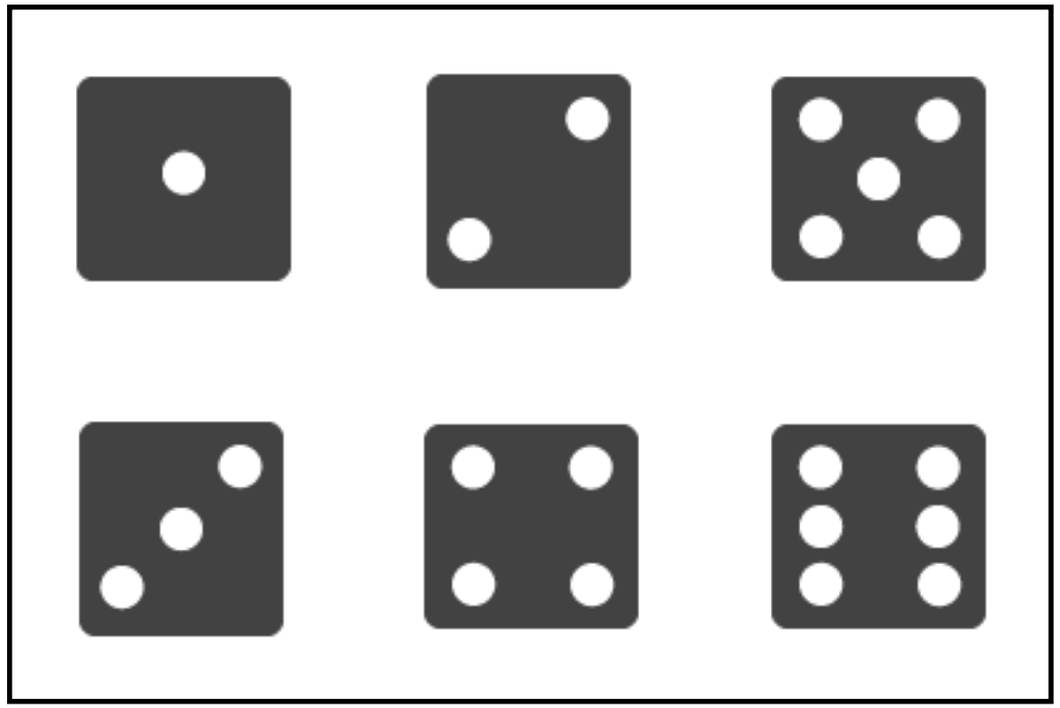
CSSスプライトの例

6回Webサーバへの
リクエストが必要



約4Kb×6=>24Kb

1回Webサーバへの
リクエストが必要



9Kb

http://spritegen.website-performance.org
でCSSスプライト化

CSSスプライトのCSS

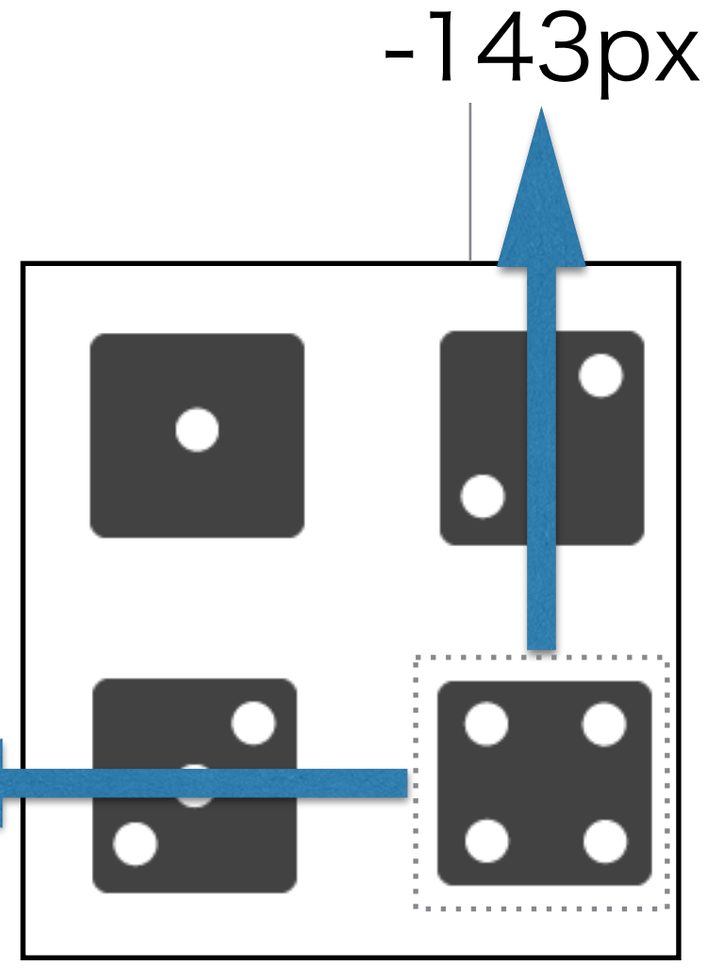
HTML

```
<span class="sprite sprite-dice-4"></span>
```

CSS

```
.sprite {
    background-image: url(csssprite.png);
    background-repeat: no-repeat;
    display: block;
}

.sprite-dice-4 {
    width: 128px;
    height: 128px;
    background-position: -143px -143px;
}
```



練習問題4

CSSスプライトのメリットでは無いものを選択しなさい。

A. ファイルサイズの総計を減らす。

B. ダウンロード回数を減らす。

C. CSSを単純化する。

D. キャッシュを有効活用する。

高解像度画面向け対応

- iOSデバイスや、android端末などのモバイルデバイスでは、一般的なPCにくらべて高解像度の画面を備えている(iOSのRetinaディスプレイなど)。
- **実際のピクセル数と論理的なCSSピクセル数が異なることがある。**(縦横2画素で1ピクセルを表現することで文字の表示を滑らかにするなど)

density

- CSSピクセルと物理ピクセルの比率(密度)
- @media (**-webkit-min-device-pixel-ratio: 2**){
  ...
  }
  iOSの場合、値が2ならRetinaディスプレイ。
  Androidの場合、値は様々。
- resolutionというメディア特性も用意されている。
  (単位はdpi/dpcm)

-webkitとついているが
今ではApple以外のブラウザでも対応している。

viewport

- モバイル機器などで、表示領域のサイズを表わす。
- デフォルト値は**980px**。
- Webブラウザは要素をレイアウトする際にviewportを基準にする。
(PCのWebブラウザのウィンドウ内の表示領域のように)

initial-scale

- <meta name="viewport">で使用するプロパティ。
- content属性に記述する。
  例) content="initial-scale=1.0"
- **表示初期状態でのズーム倍率。**
- **通常は1.0にすることが多い。**
- 関連するプロパティ
  - **minimum-scale**は最小倍率
  - **maximum-scale**は最大倍率
  - **user-scalable**=yesならユーザによるズーム可、noならズーム不可

Webブラウザにより対応に
差があるため注意

viewportの例

<meta name="viewport"
content="**width=device-width,initial-scale=1**">

- モバイルデバイスでは、画面の幅とviewportを一致させることが多い。(**width=device-width**)
- 初期の表示倍率は1倍
- **Webブラウザの種類、バージョンによって動作が異なるので検証が必要。**

練習問題5

viewportの正しい指定を選択しなさい。

A. <viewport initial-scale="1.0">

B. <meta name="viewport" initial-scale="1.0">

C. <meta name="viewport" content="initial-scale=1.0">

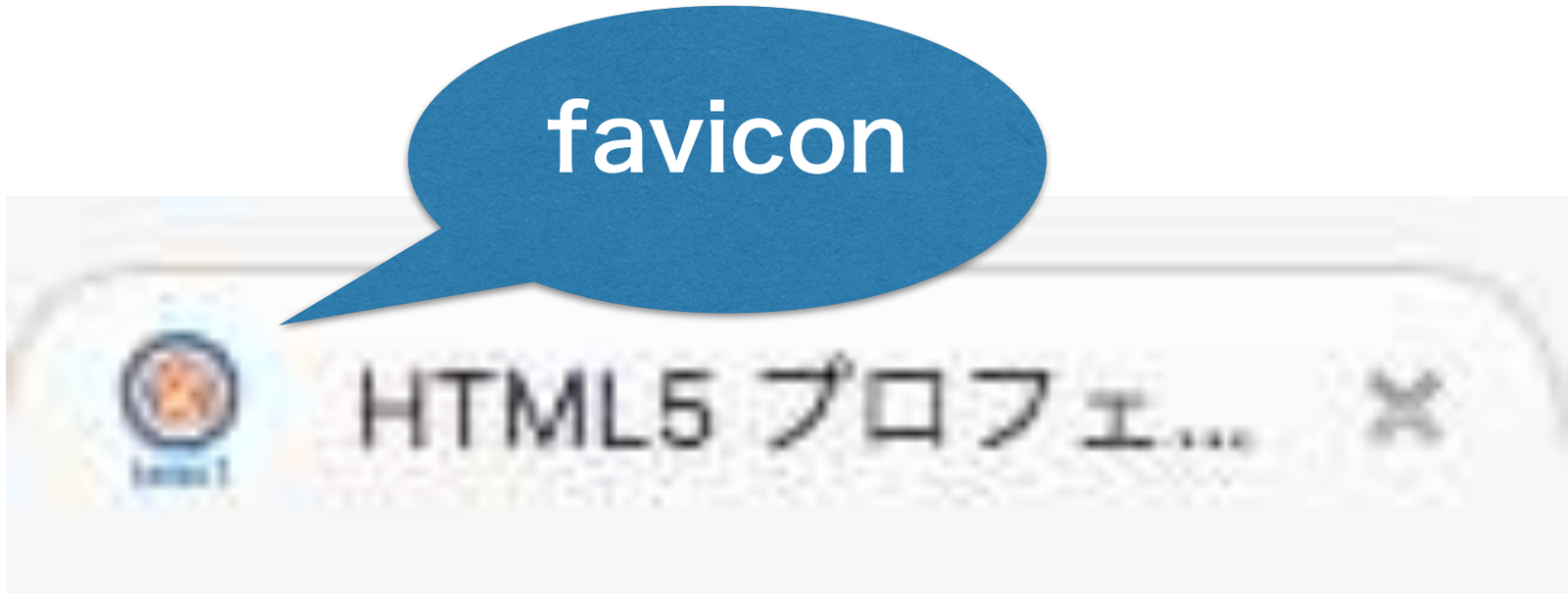
D. <viewport content="initial-scale=1.0">

ファビコン

- Webブラウザのタブに表示されるアイコン

- <link **rel="icon"** href="images/favicon.ico" type="image/vnd.microsoft.icon">

- 大きな画像を用意しておいて、ファビコン生成をしてくれるWebサイトを利用すると簡単に作成できる。



https://ao-system.net/favicongenerator/ など

アイコン設定(iOS)

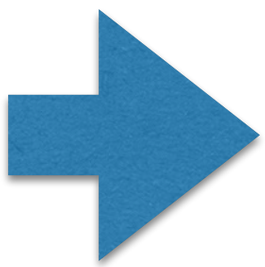
iOSデバイスで、Webページを"ホーム画面に追加"した際にホームに表示するアイコンを指定することができる。

デバイスに合わせる

「**apple-touch-icon.png**」を**サイトのルートディレクトリ**に設置
もしくは、
<link rel="**apple-touch-icon**" href="アイコンファイル名" **sizes="120x120"**>
をHTMLファイルのヘッド要素内に記述する(サイズ別に複数記述可)。

画像ファイル名に"**-precomposed**"を付けると加工されずにそのままアイコンに使用される(iOS6までは光沢処理が行なわれていたがiOS7からなし)。

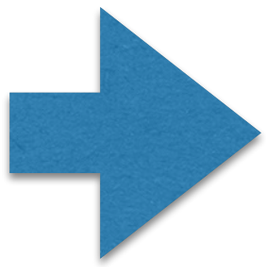
元画像



アイコン設定(Android)

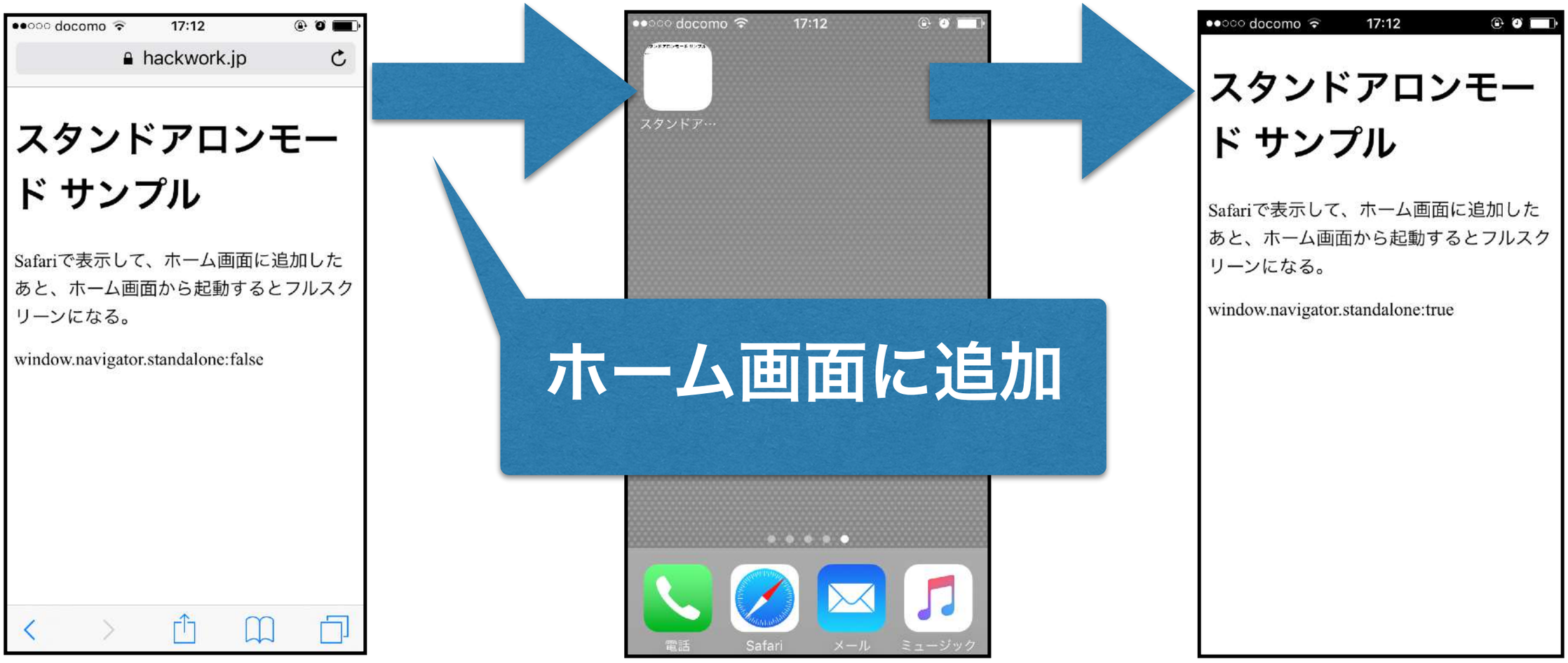
Androidの場合、iOSと同様の指定も可能だが、存在しなければファビコンがホーム画面のアイコンに使用される。

元画像



スタンドアローンモード

- **iOS**のMobile Safariに実装されている機能
- フルスクリーンモードでネイティブアプリのように見せる
- <meta name="**apple-mobile-web-app-capable**" content="**yes**">



電話番号へのリンク

- 電話番号指定したリンクをタップするとその電話番号に発信する
- 時報(有料)

練習問題6

電話の発信ができるHTMLの記述を選択しなさい。

※(電話番号)は有効な電話番号とする

A. 電話を掛ける

B. <input type="tel" value="(電話番号)">

C. 電話を掛ける

D. <input type="phone" value="(電話番号)">

script要素のasync属性、defer属性

JavaScriptの実行によるブロック問題

- scriptタグが読み込まれると、HTML要素のレンダリングやDOM構築がブロックされるため、読み込みや処理に時間が掛ると表示が遅く見える

- body終了タグの直前にscriptを記述することで、ブロックの影響を低減する手法が一般化

- body.onloadイベントやjQueryのreadyメソッドなど、DOM構築完了後にまとめて実行するニーズ

→async,defer属性で解決できる。

javascriptによるブロック実例

```
<h1>async属性なし</h1>
<script src="js/large.js"></script>
<script src="js/small.js"></script>
<img src="images/img.png">
```

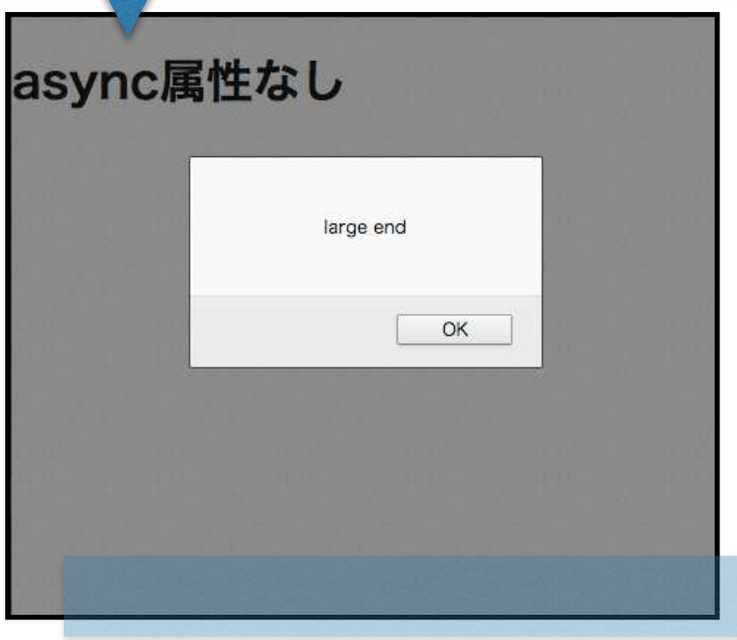
ファイルサイズ(大)
"large script"とアラート

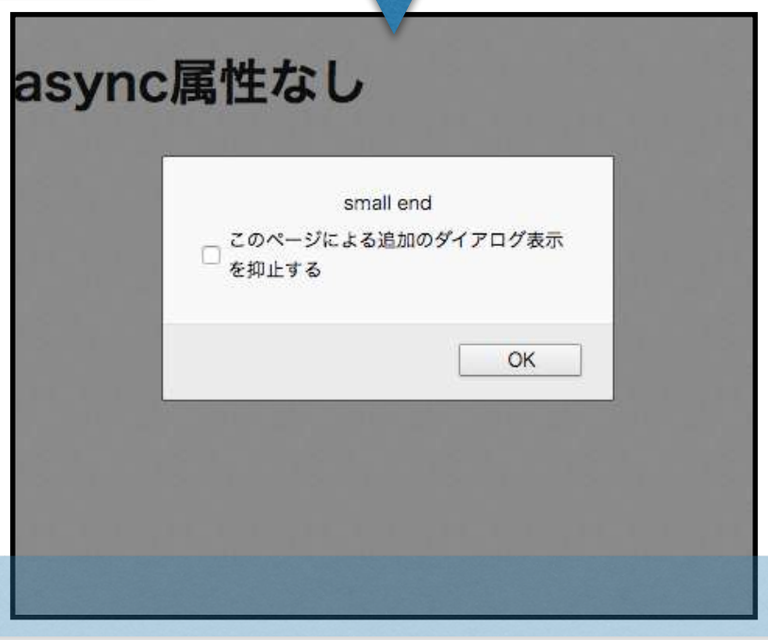
ファイルサイズ(小)
"small script"とアラート

h1は表示されている
"large script"とアラート(OK待ち)
imgは表示されていない(ブロック)

h1は表示されている
"small script"とアラート(OK待ち)
imgは表示されていない(ブロック)

img表示





async属性

- script要素にasync属性を付けることで、他の処理をブロックせずにスクリプトを読み込める。

- インラインスクリプトは実行できない。

- document.writeは実行できない。

- スクリプトファイルを読み込み終った順に実行される可能性があるため、**スクリプトの実行順序は維持されない。**

async属性 実例

```
<h1>async属性あり</h1>
<script src="js/large.js" async></script>
<script src="js/small.js" async></script>
<img src="images/img.png">
```

ファイルサイズ(大)
"large script"とアラート

ファイルサイズ(小)
"small script"とアラート

h1は表示されている
imgは表示されている(ブロックなし)
"small script"とアラート(OK待ち)

h1は表示されている
imgは表示されている(ブロックなし)
"large script"とアラート(OK待ち)

先に読み込まれた
small.jsが先に実行された

defer属性

- async属性同様、スクリプト読み込みを非同期に行ない、**onloadイベント発生直前**にスクリプトを実行する。

→bodyの終了タグ直前にスクリプトファイルを読み込む必要がない。

- スクリプトの**実行順序は維持**される。

- インラインスクリプトには適用されない。

defer属性 実例

```
<h1>defer属性あり</h1>
<script src="js/large.js" defer></script>
<script src="js/small.js" defer></script>
<img src="images/img.png">
```

ファイルサイズ(大)
"large script"とアラート

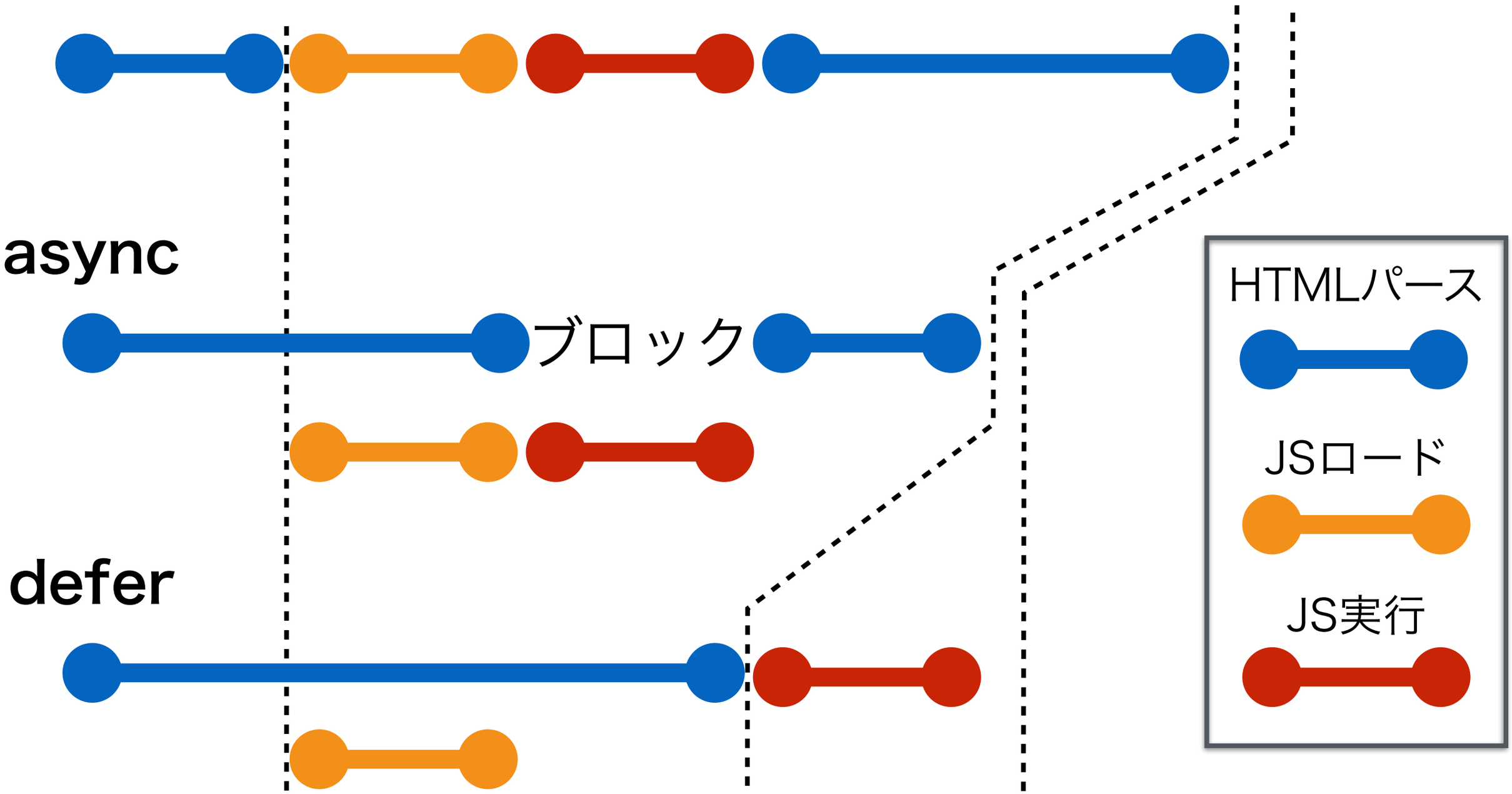
ファイルサイズ(小)
"small script"とアラート

h1は表示されている
imgは表示されている(ブロックなし)
"large script"とアラート(OK待ち)

h1は表示されている
imgは表示されている(ブロックなし)
"small script"とアラート(OK待ち)

async / defer のタイミング

通常の処理



練習問題7

script要素のasync属性について間違っている説明を選択しなさい。

A. スクリプトの実行順序は不定である。

B. body.onloadの直前に実行される。

C. 非同期に実行される。

D. インラインスクリプトには適用されない。

本日の概要

- HTML5 レベル1 試験について
- レスポンシブWebデザイン
- マルチデバイス対応ページの作成
- メディアクエリ
- スマートフォンサイト最適化

練習問題正解

- 問題1: **C**　ノーマライズCSSの説明です。
- 問題2: **B,C** タイルレイアウト及び固定レイアウト。
- 問題3: **B** smartphoneはありません。
- 問題4: **C** 一般にCSSは複雑化します。
- 問題5: **C** name属性とcontent属性で指定します。
- 問題6: **A** href属性にtel:電話番号で指定します。
- 問題7: **B** defer属性の説明です。