

webデザイナーが今覚えておくべき Flexboxとレスポンシブイメージ

平成28年7月28日(木) @ レバレジーズ本社

株式会社ICS 鹿野 壮

#ヒカラボ

自己紹介



株式会社ICS
インターラクティションデザイナー
鹿野 壮 (かの たけし)



@tonkotsuboy_com

自己紹介

モバイルアプリ制作

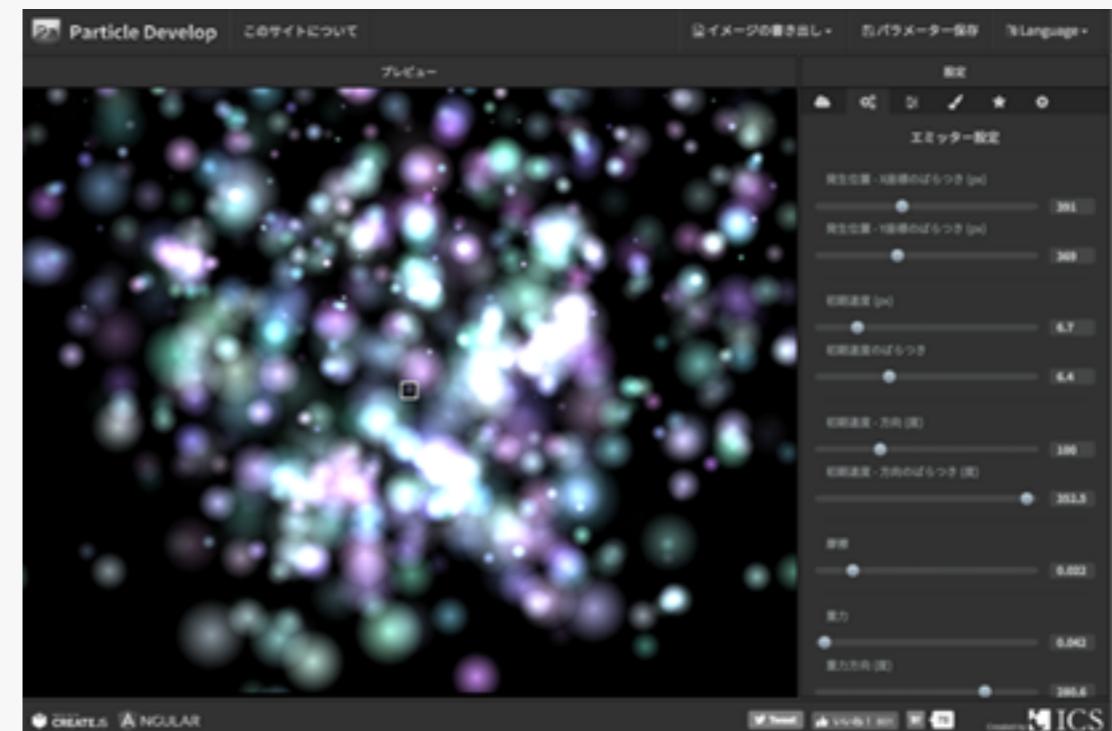
webサイト制作



パズ億～爽快パズルゲーム

Google Play 2014年ベストゲーム

<https://goo.gl/4Fx7T2>



Particle Develop

2016年Chrome Experiments

<http://ics-web.jp/projects/particle-develop/>

自己紹介

技術記事の執筆

The screenshot shows the homepage of ICS MEDIA. At the top, there's a dark banner with the text "ICS MEDIA" and "interspace". Below it is a large purple and black background image featuring the text "ics.media" and "インタラクションデザイン × ウェブテクノロジー". A subtext below reads: "最先端のリッテラチュアやノバーション、制作に熱を注ぐ新しいトピックスを紹介する ICS MEDIAによる新規メディア" (Introducing new topics in literature and innovation, focusing on the latest technologies). The main navigation menu includes "Web制作", "3D表現", "制作効率化", "デザイン", "IoT", and "アプリ開発". Below the menu, a section titled "話題の記事" displays three articles with thumbnails and titles: "LINEのアニメーションスタンプはこう作る！APNGファイルの作り方を徹底解説" (Design), "今どきの入力フォームはこう書く！HTMLコーダーが抑えるべきinputタグの書き方まとめ" (Web制作), and "CSS3とHTML5 Canvasで作るモーショングラフィックイブラリ「ParticleJS」を公開" (Web制作).



The screenshot shows a Qiita profile page for the user "かの" (Kano). The profile picture is a cartoon character with black hair and red cheeks. The user's name is "鹿野 壮" (@tonkotsuboy_com) and they have contributions, followers, and items. The contributions chart shows a steady increase from July 8th to July 26th. Below the chart, there are two popular posts: "これを選んでおけば間違いない2016年流行りのCSS・JavaScript・タスクランナー" (575 contributions) and "AJAX通信をするときはタイムアウト処理を必ず入れてほしい (切実)" (482 contributions). The user's bio mentions they work at ICS Inc., are based in Tokyo, Japan, and are a part-time programmer.

ICS MEDIA
オウンドメディア

<https://ics.media/>

Qiita

3700 Contribution

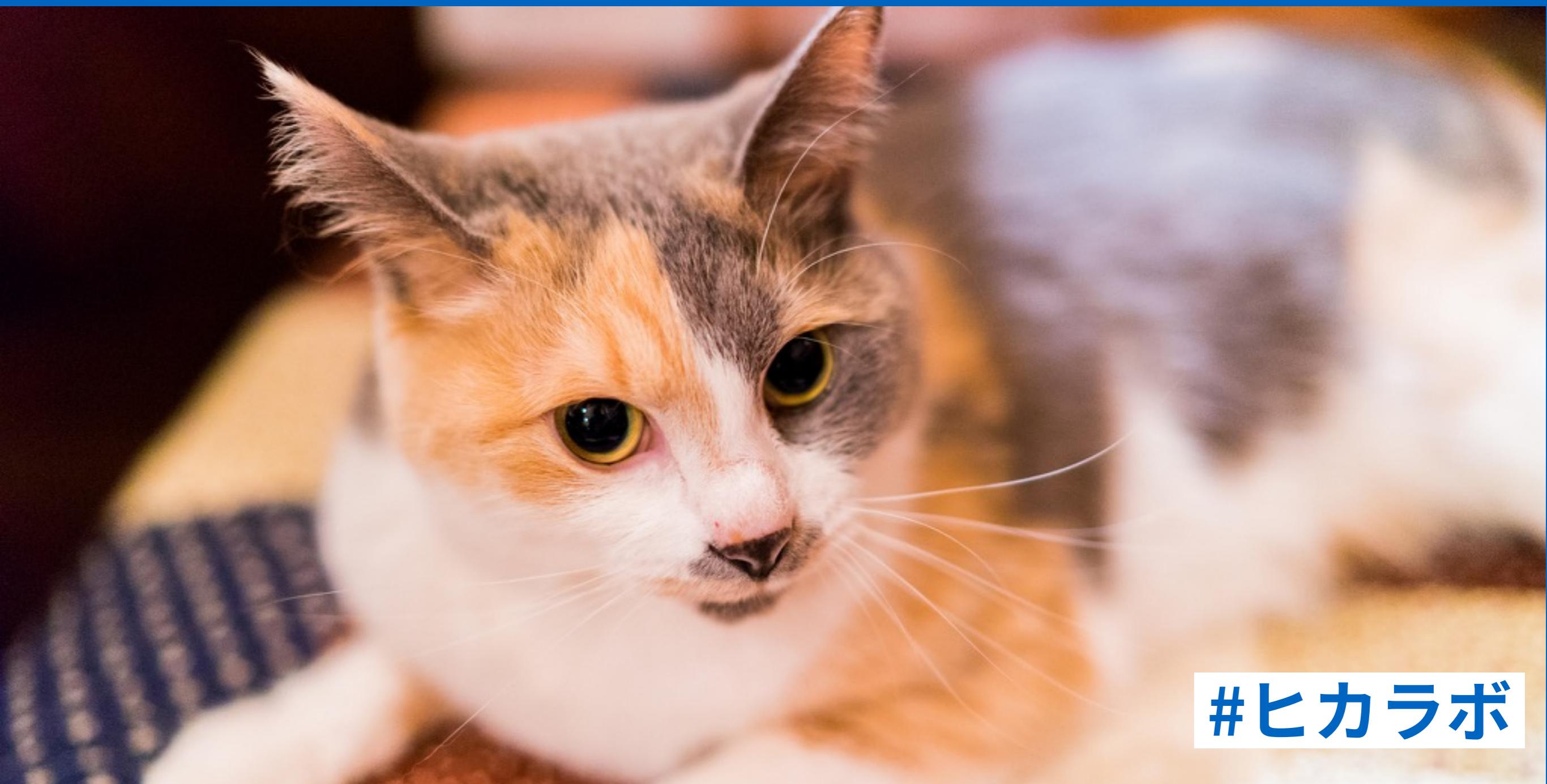
http://qiita.com/tonkotsuboy_com

本日の流れ

1. 2016年のフロントエンド事情
2. イマドキのボックスレイアウト
「Flexbox」
- (休憩)
3. 画像表示の新常識
「レスポンシブイメージ」

Section 1

2016年のフロントエンド事情



#ヒカラボ

2016年のフロントエンド事情

3つの大きな変化

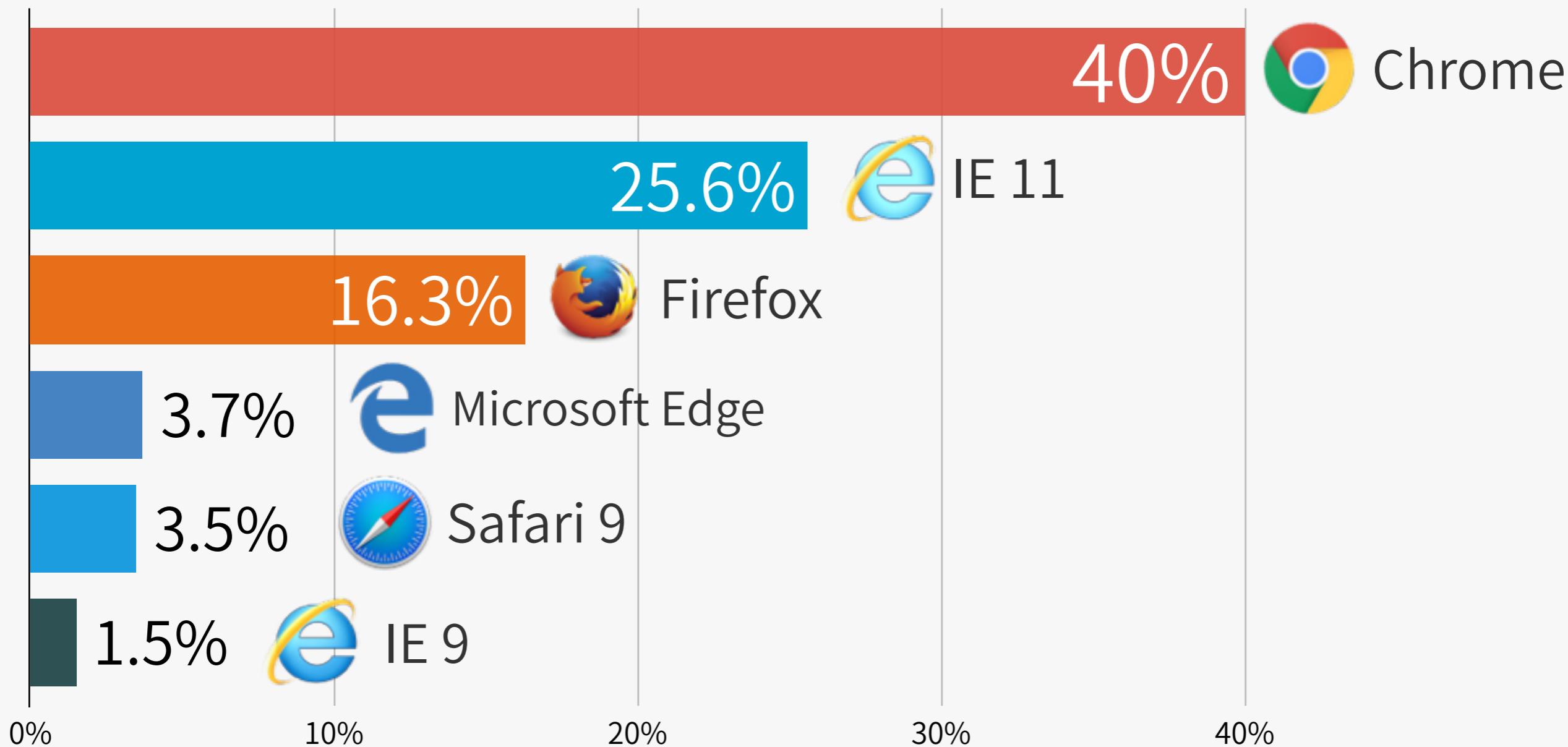
1. 古いIEのシェアの縮小
2. スマートフォンブラウザの割合の拡大
3. HTML5・CSS3の勧告

2016年のフロントエンド事情

1. 古いIEのシェアの縮小

日本におけるデスクトップブラウザ別シェア

(2016/4~2016/6 [Stat Counter](#) 調べ)

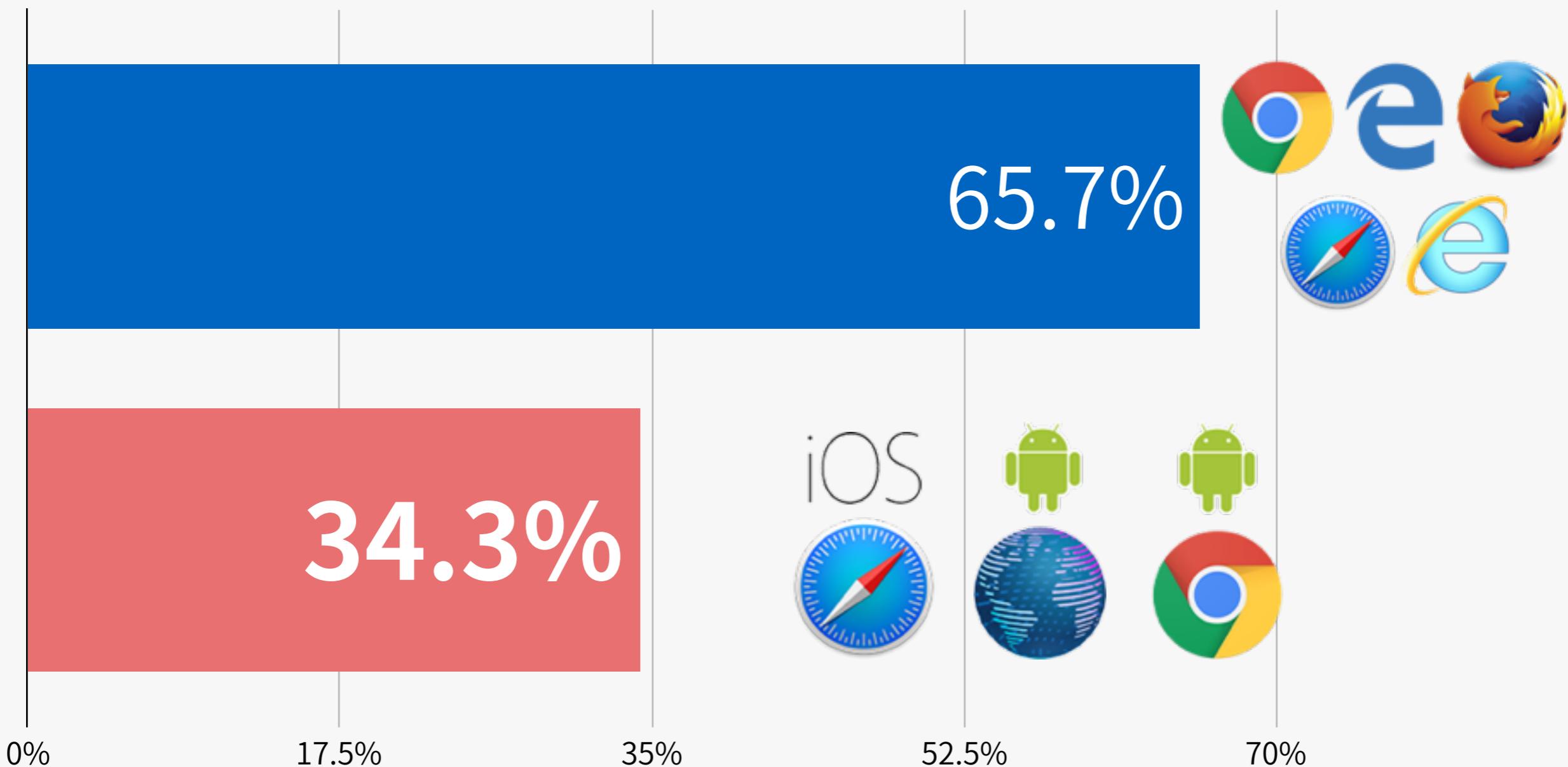


2016年のフロントエンド事情

2. スマートフォンブラウザの割合の拡大

日本におけるプラットフォーム別シェア

(2016/4~2016/6 [Stat Counter](#) 調べ)



2016年のフロントエンド事情

3. HTML5の勧告、CSS3の勧告

- HTML5は2014年10月28日に勧告
- 今秋にはHTML5.1が勧告予定([HTML 5.1 - W3C](#))
- CSS3は段階的に勧告
([CSS current work - W3C](#))



2016年のフロントエンド事情

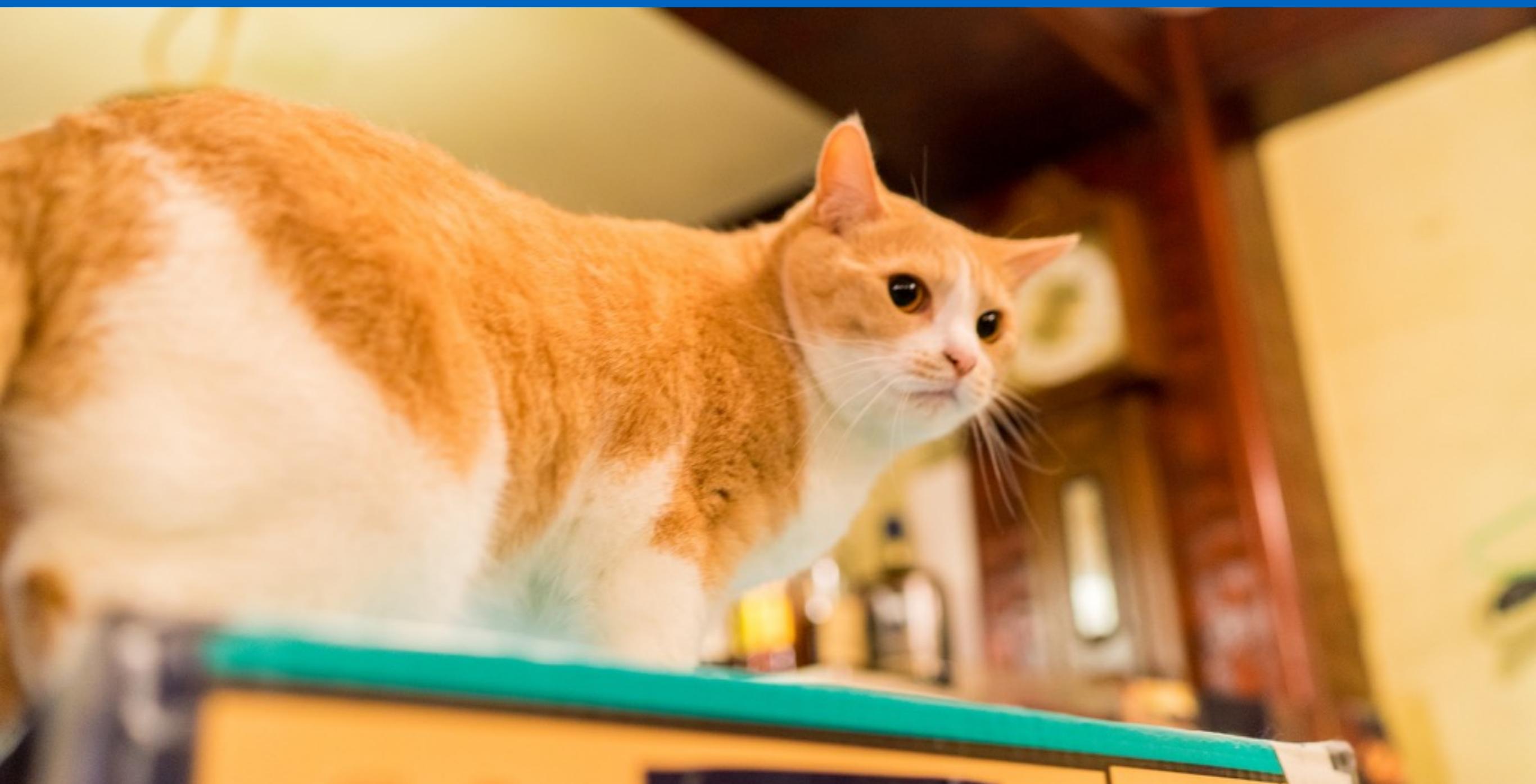
今覚えておきたい2つの技術

1. Flexbox

2. レスポンシブイメージ

Section 2

イマドキのボックスレイアウト Flexbox



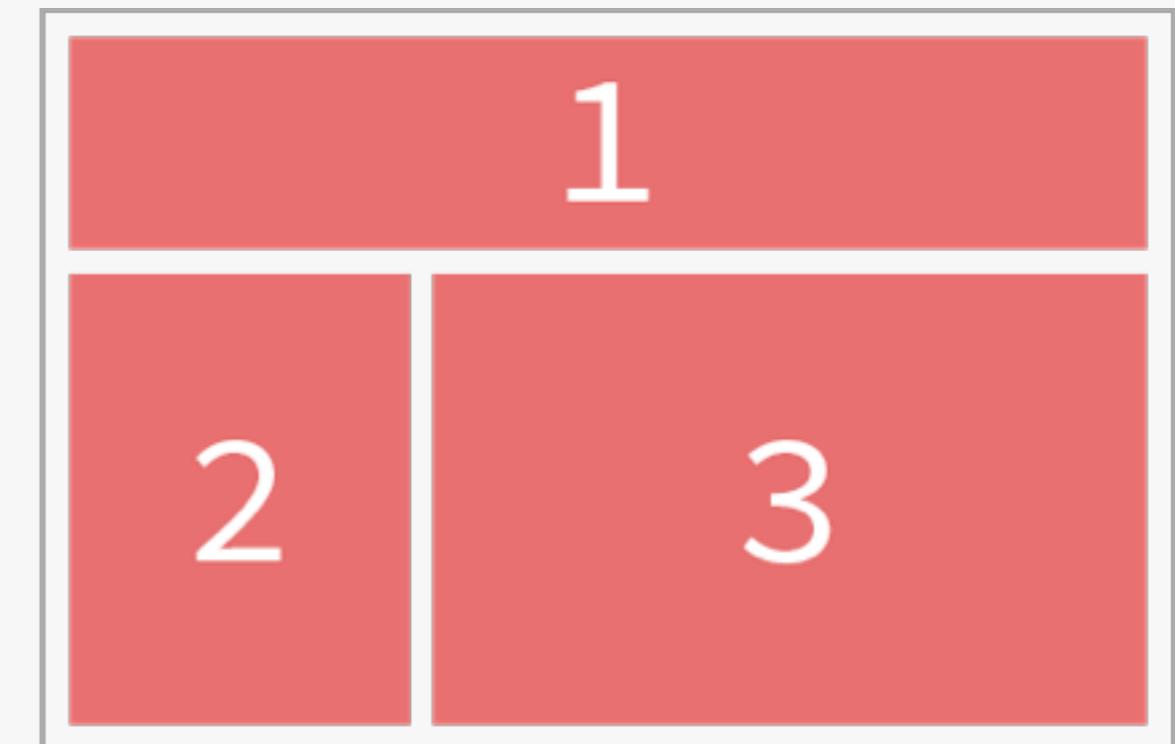
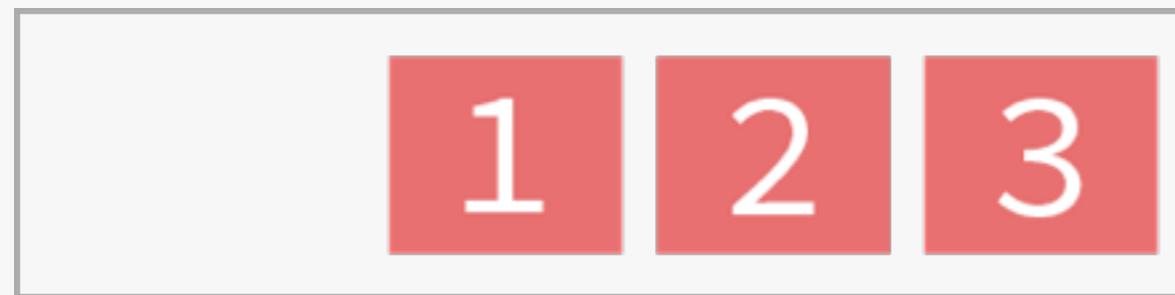
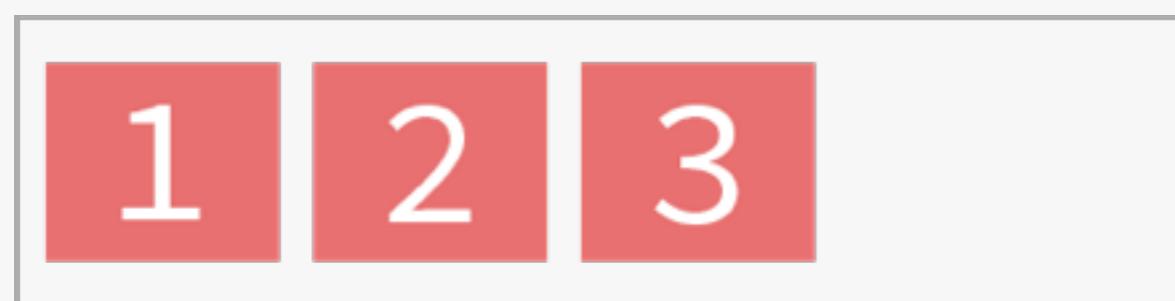
Section 2-1

Flexboxとは

Flexboxとは

Flexbox

- CSS3のボックスレイアウト用機能
- 横並び、右・左寄せ、縦中央揃え

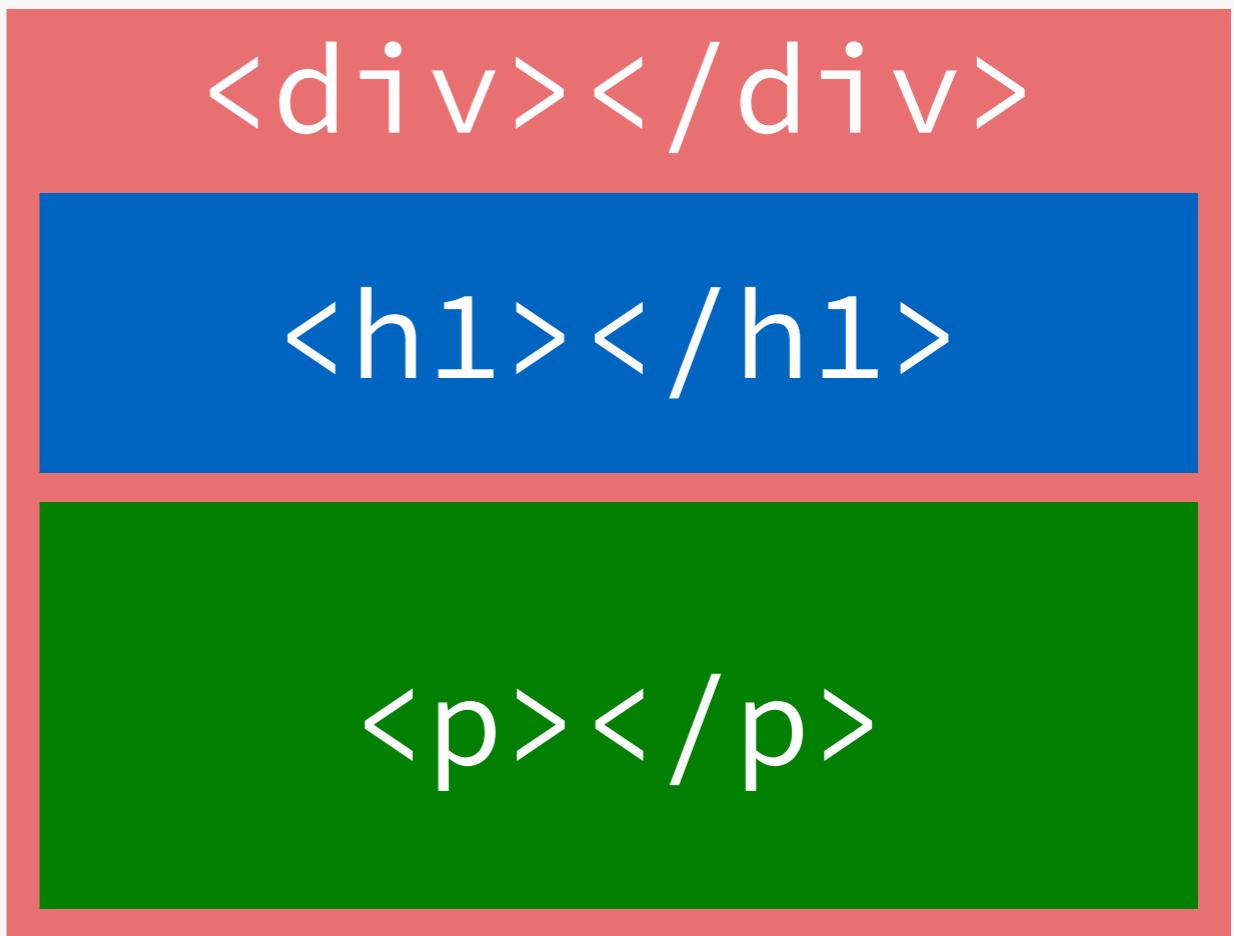


Flexboxとは

「ボックス」

- ・ HTML上の各要素を囲む領域のこと
- ・ divタグやpタグがボックスを形成する

```
<div>
  <h1>タイトル</h1>
  <p>テキスト</p>
</div>
```



Flexboxとは

大手フレームワークで採用



Bootstrap 4



<http://v4->

alpha.getbootstrap.com/



Ionic 2

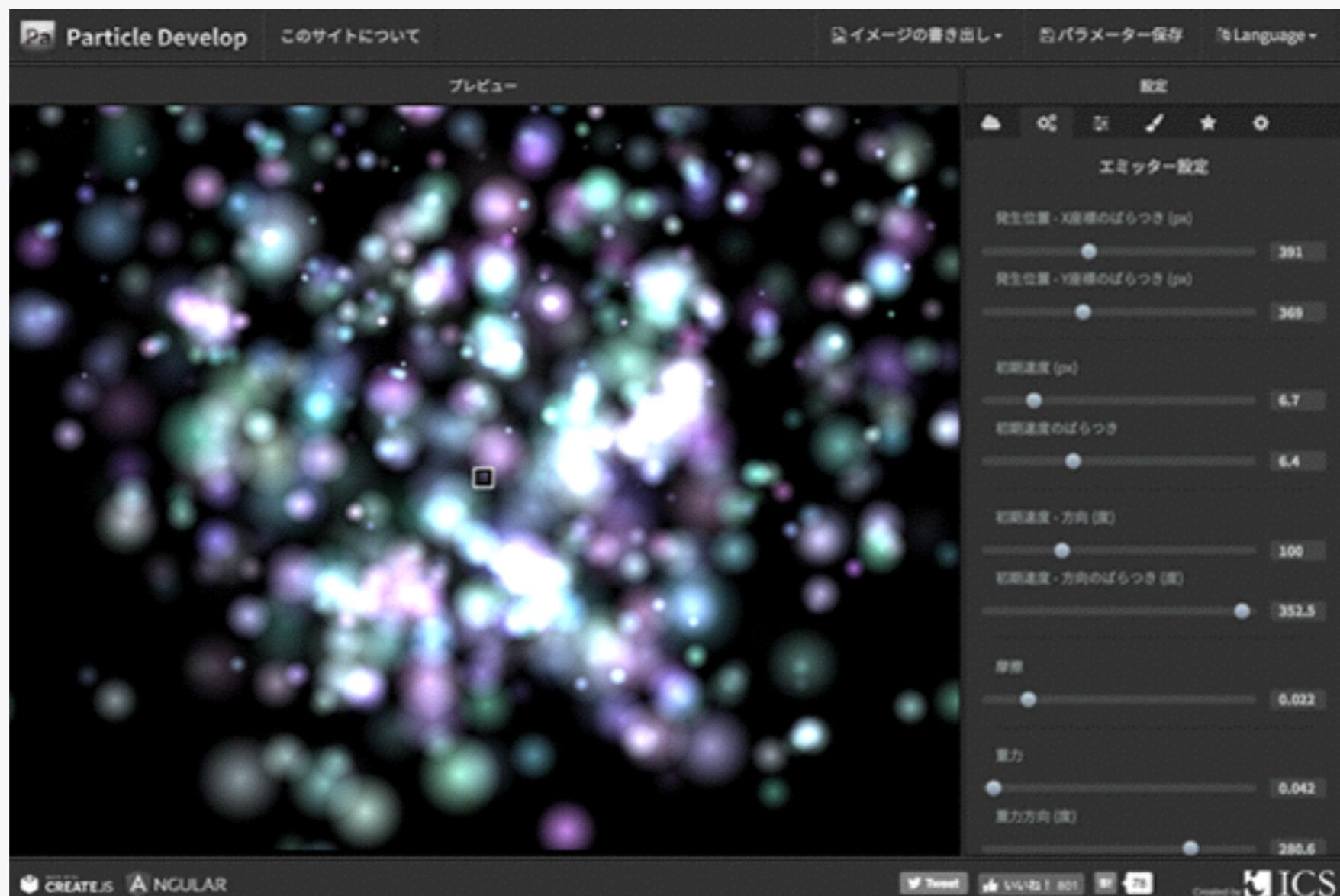


<http://ionic.io/2>

Flexboxとは

Particle Develop

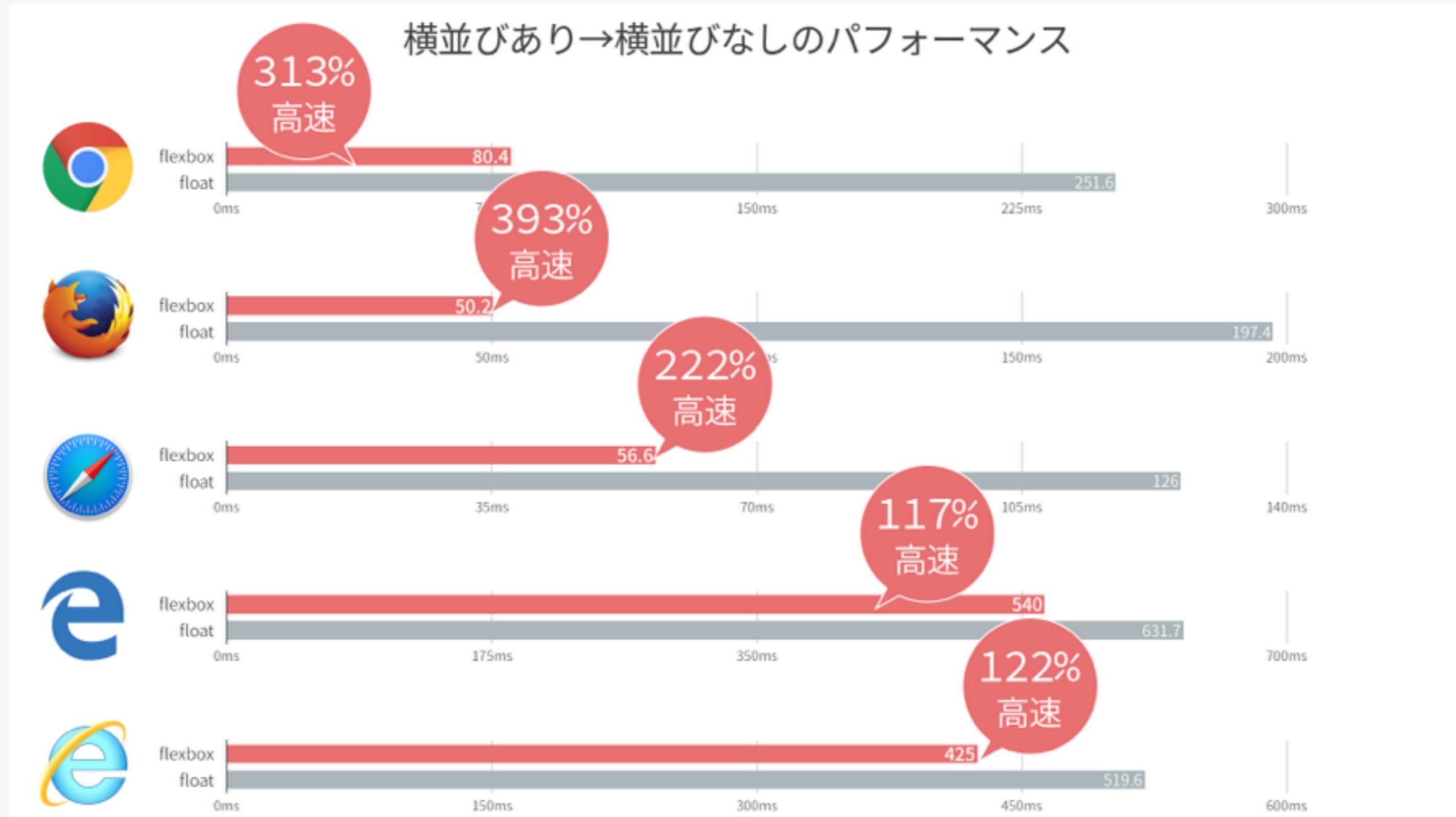
floatを使わずにレイアウト



<http://ics-web.jp/projects/particle-develop/>

Flexboxとは

floatよりも高いパフォーマンス



最適なCSSの横並びはどっち!? Flexboxとfloatのパフォーマンス比較

Section 2-2

Flexboxで レイアウトする

Flexboxでのレイアウト



menu1 menu2 menu3

animal1



Lorem ipsum dolor sit amet, consectetur adipisicing elit. Harum, repellendus.

animal2



Cum dolorem id nemo quas recusandae repellendus totam vel voluptates?

animal3



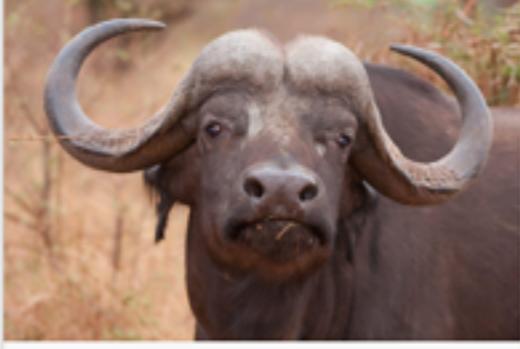
Animi eum labore rem? Atque libero nihil vel vero vitae!

animal4



Ab amet cum eos esse fugit maiores molestias neque sint.

animal5



Animi ducimus ea, eum laborum nostrum nulla officia quo voluptate.

animal6



Amet deserunt excepturi necessitatibus numquam, quidem repellat repellendus similique

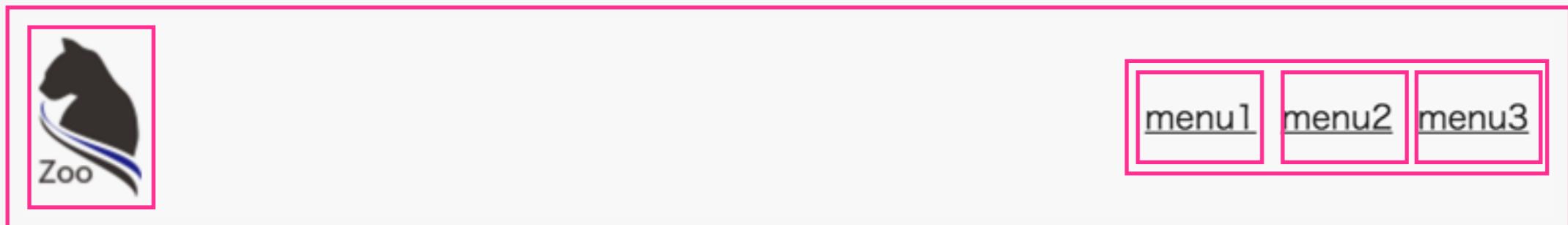
ヘッダー

3カラムの
カード

Flexboxでのレイアウト

ヘッダー部分の横並び・縦中央揃え

1. 横並び
2. 左右に振り分け
3. 上下中央揃え



Flexboxでのレイアウト

1. Flexboxのレイアウトを適用する
2. 各アイテムの配置方法を設定する

display: flex;

justify-content: (水平方向の並び方)

align-items: (垂直方向の並び方)

Flexboxでのレイアウト

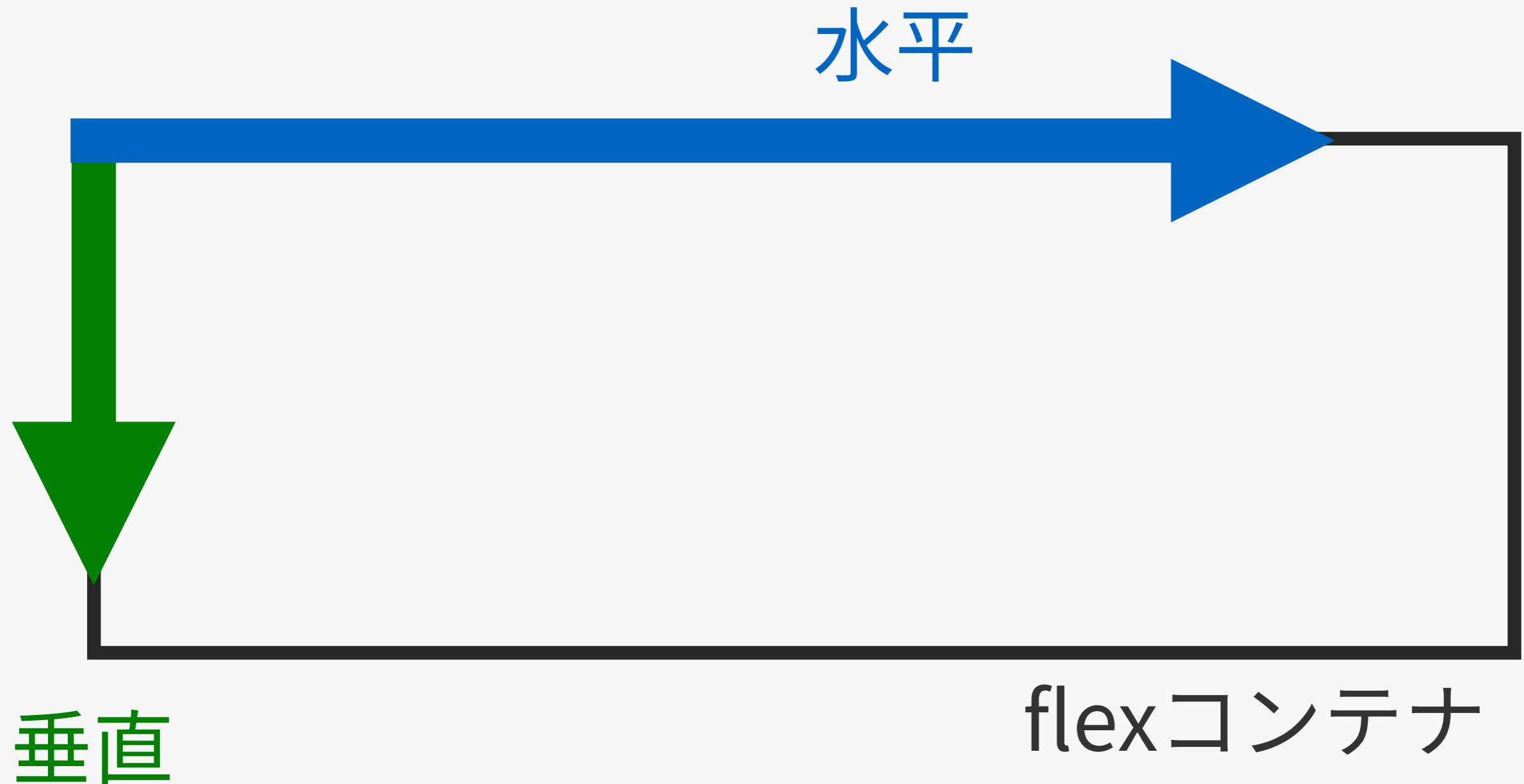
Flexboxの基本

`display : flex`

flexコンテナ

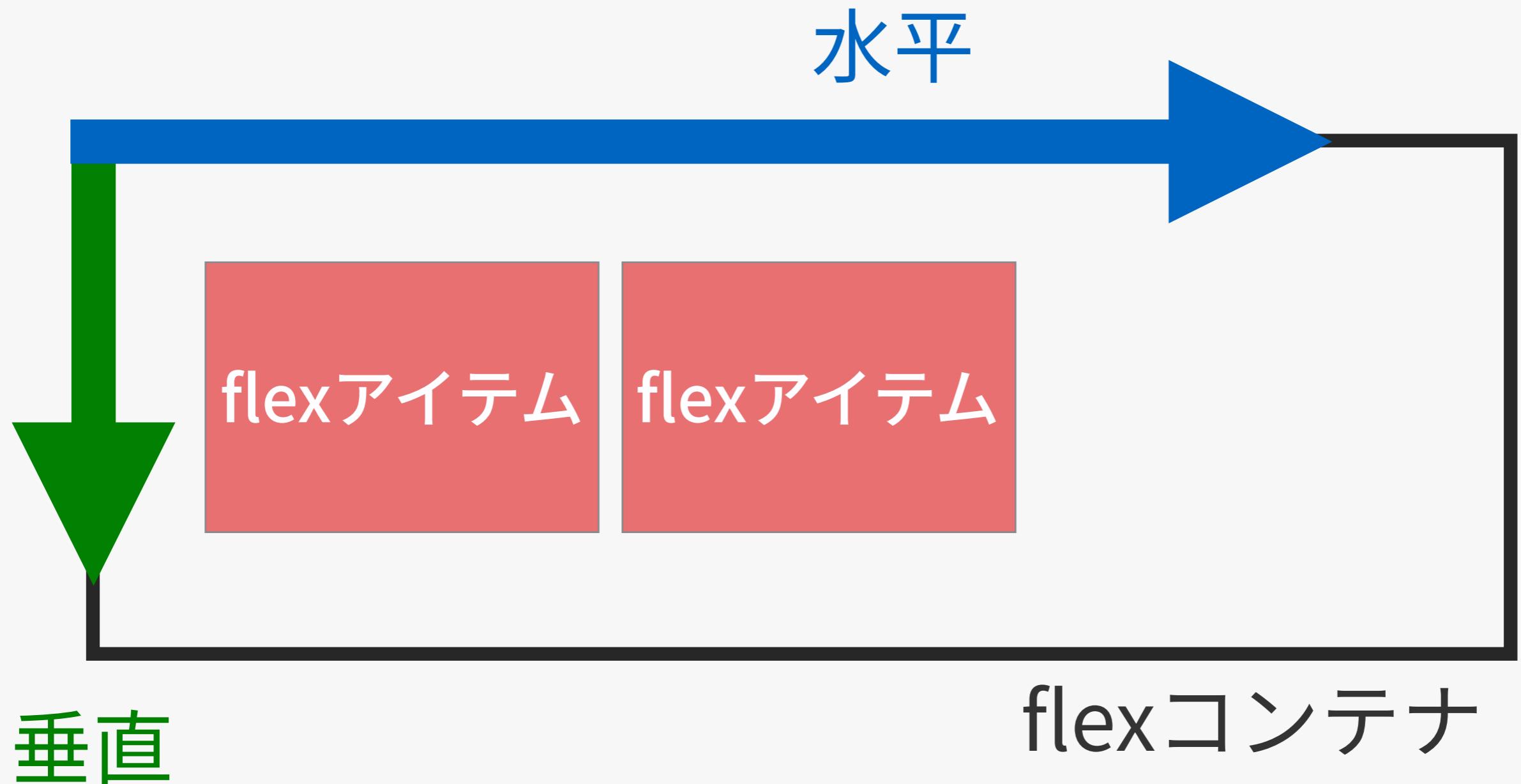
Flexboxでのレイアウト

Flexboxの基本



Flexboxでのレイアウト

Flexboxの基本



Flexboxでのレイアウト

水平方向の並び方→justify-content

flex-start : 左揃え



space-between :
均等配置



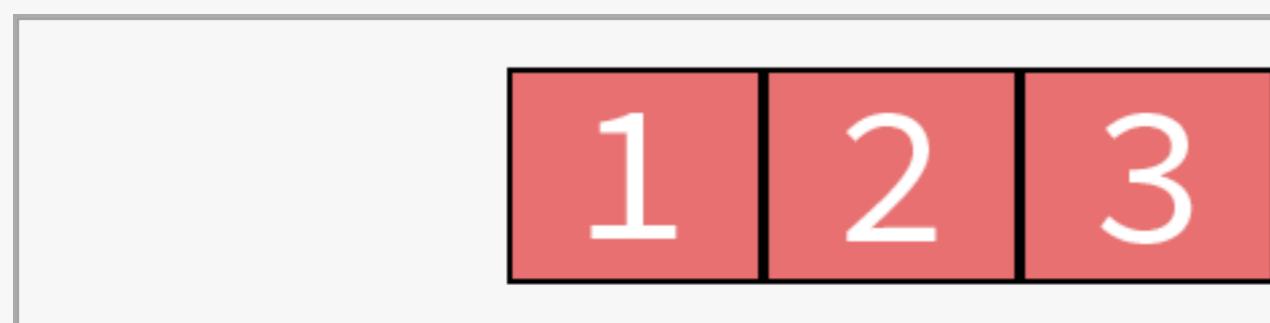
center : 中央揃え



space-around :
均等配置



flex-end : 右揃え



Flexboxでのレイアウト

垂直方向の並び方→align-items

flex-start : 上揃え



stretch : 親と同じ高さ



center : 中央揃え



flex-end : 下揃え



Flexboxでのレイアウト

Live Coding

Flexboxでのレイアウト

3カラムのカードレイアウト

<p>animal1</p>  <p> Lorem ipsum dolor sit amet, consectetur adipisicing elit. Harum, repellendus.</p>	<p>animal2</p>  <p> Cum dolorem id nemo quas recusandae repellendus totam vel voluptates?</p>	<p>animal3</p>  <p> Animi eum labore rem? Atque libero nihil vel vero vitae!</p>	
<p>animal4</p>  <p> Ab amet cum eos esse fugit maiores molestias neque sint.</p>	<p>animal5</p>  <p> Animi ducimus ea, eum laborum nostrum nulla officia quo voluptate.</p>	<p>animal6</p>  <p> Amet deserunt excepturi necessitatibus numquam, quidem repellat repellendus similiique voluptatibus?</p>	
<p>animal7</p>  <p> Assumenda consectetur deleniti earum excepturi laudantium natus provident</p>	<p>animal8</p>  <p> A, ab autem consequuntur ipsa libero modi nihil quos ullam.</p>		

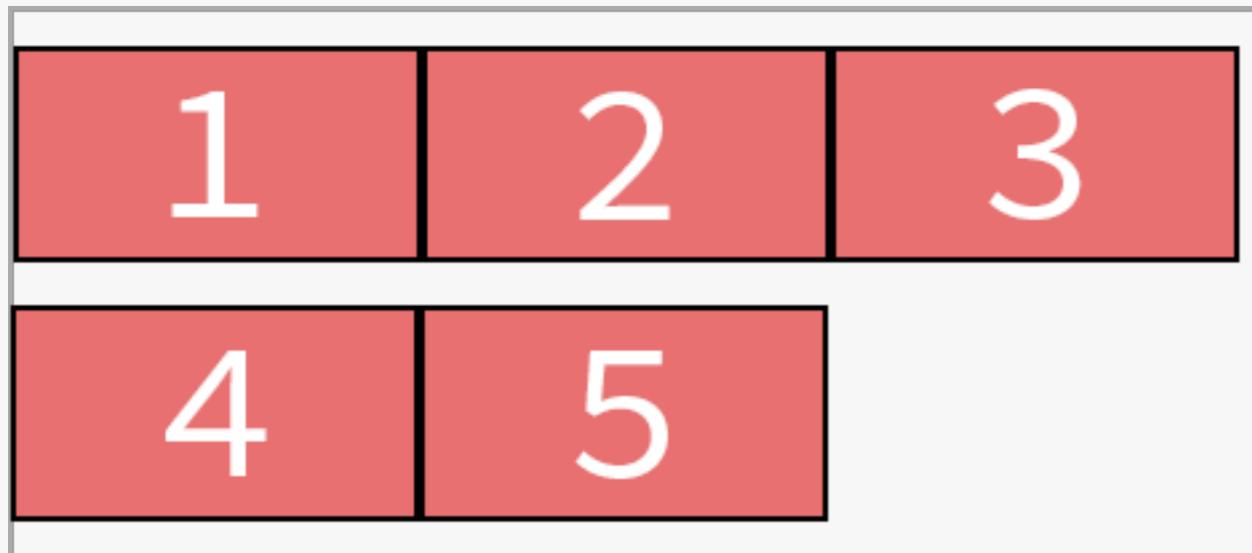
Flexboxでのレイアウト

折り返しの設定をする:flex-wrap

nowrap : 折り返さない



wrap : 折り返す



Flexboxでのレイアウト

Live Coding

Section 2-3

Flexboxで レスポンシブ対応

Flexboxでレスポンシブ対応

レイアウトを変える



animal1

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Harum, repellendus.

animal2

Cum dolorem id nemo quas recusandae repellendus totam vel voluptates?

animal3

Animi eum labore rem? Atque libero nihil vel vero vitae!

animal4

Ab amet cum eos esse fugit maiores molestias neque sint.

animal5

Animi ducimus ea, eum laborum nostrum nulla officia quo voluptate.

animal6

Amet deserunt excepturi necessitatibus numquam, quidem repellat repellendus similiqe

[menu1](#) [menu2](#) [menu3](#)



Zoo [menu1](#) [menu2](#) [menu3](#)

animal1

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Harum, repellendus.

animal2

Cum dolorem id nemo quas recusandae repellendus totam vel voluptates?

animal3

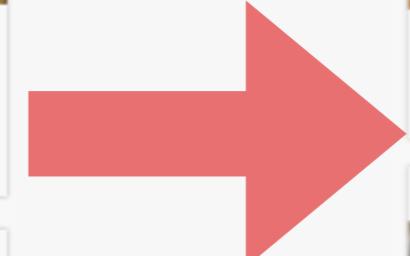
Animi eum labore rem? Atque libero nihil vel vero vitae!

animal4

Ab amet cum eos esse fugit maiores molestias neque sint.

animal5

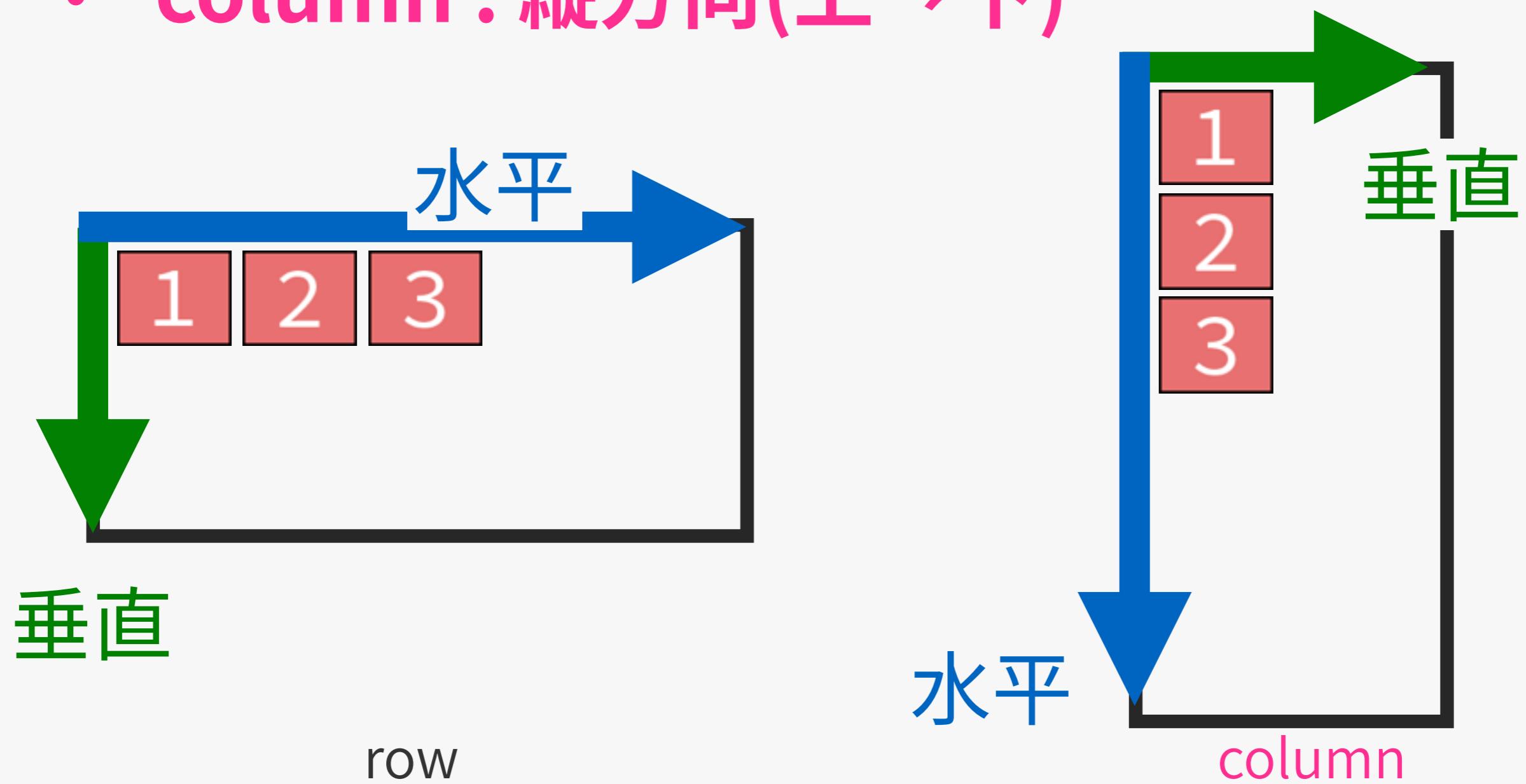
animal6



Flexboxでレスポンシブ対応

並び方向の設定→flex-direction

- row : 横方向(左→右)
- column : 縦方向(上→下)



Flexboxでレスポンシブ対応

Live Coding

Section 2-4

Flexboxで クロスブラウザ対応

Flexboxでクロスブラウザ対応

Flexboxの記述方法の違い

- ・ ベンダープレフィックス
- ・ プロパティ名の違い

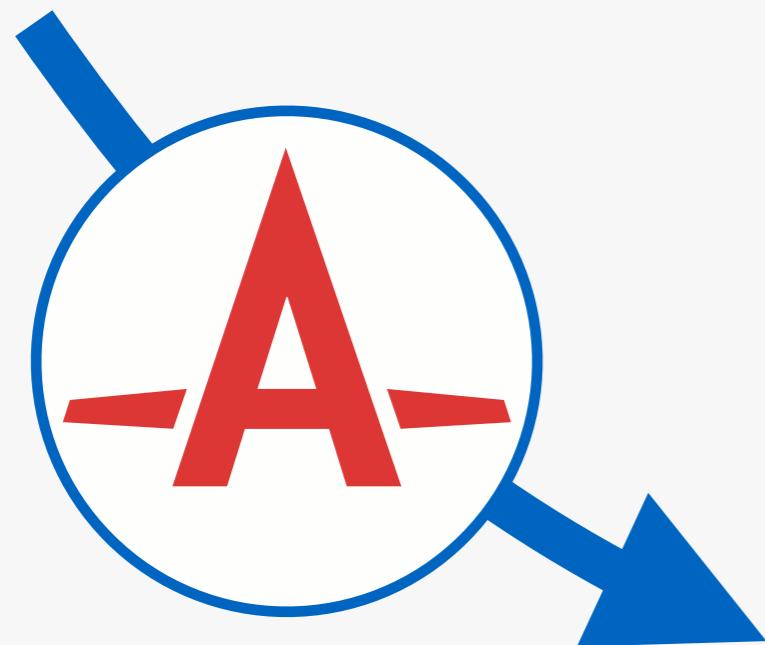
```
display: -webkit-box;  
display: -webkit-flex;  
display: -ms-flexbox;  
display: flex;
```

Flexboxでクロスブラウザ対応

Autoprefixer

自動的にベンダープレフィックスを付与するツール

`display: flex;`



Autoprefixer

`display: -webkit-box;`
`display: -webkit-flex;`
`display: -ms-flexbox;`
`display: flex;`

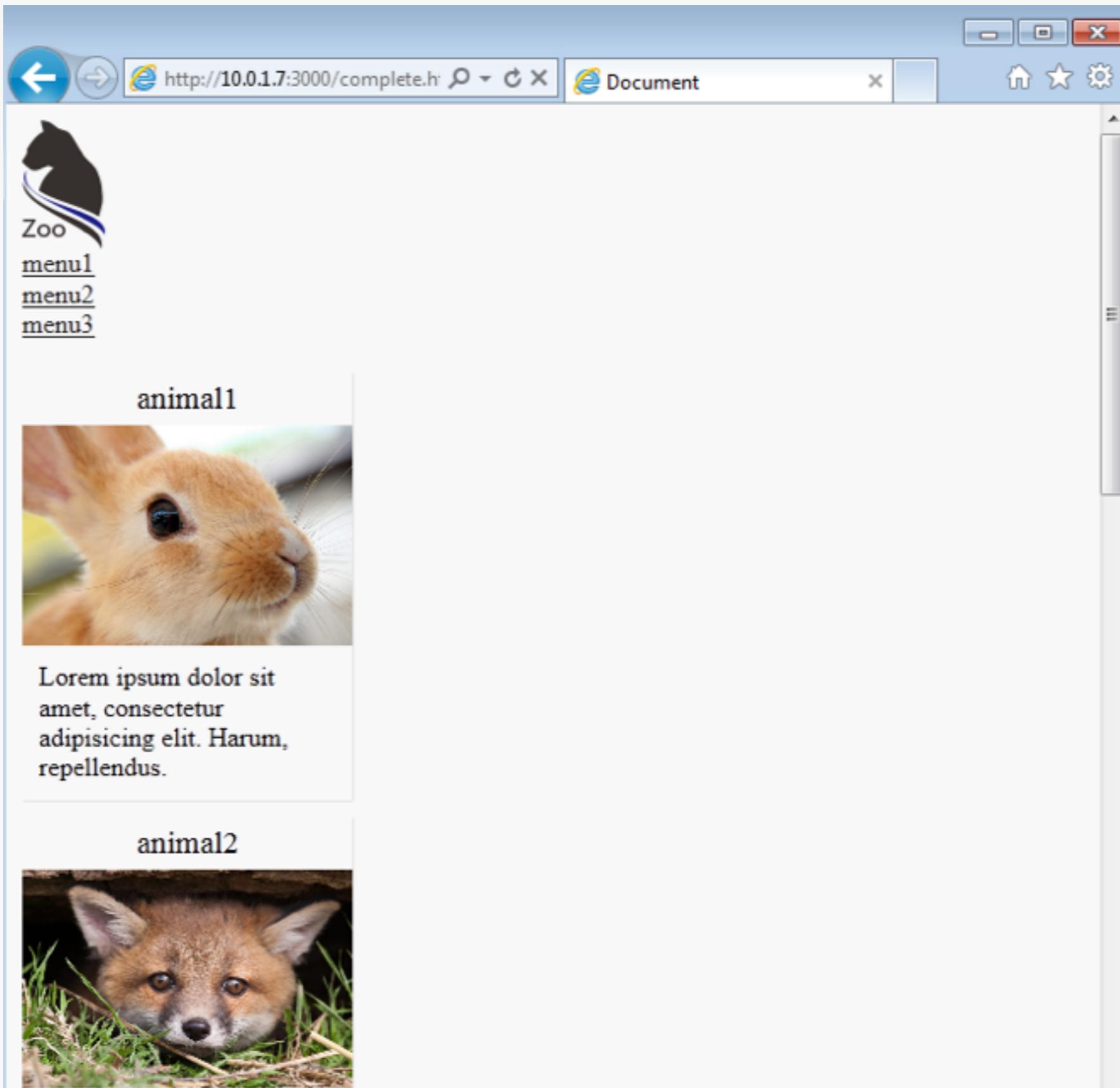
Flexboxでクロスブラウザ対応

Live Coding

Flexboxでクロスブラウザ対応

「IE 9で崩れていますけど？」

Flexboxでクロスブラウザ対応

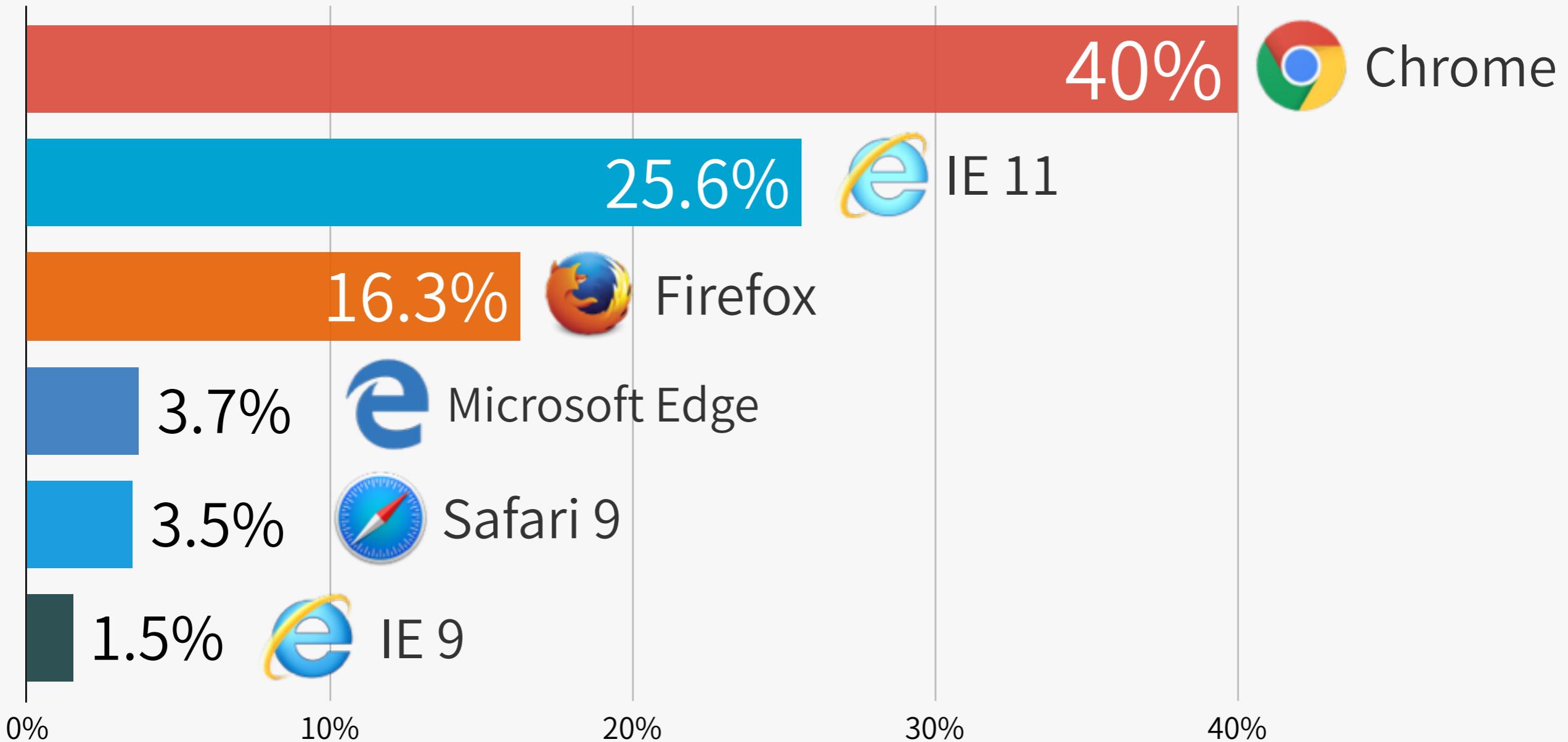


Flexboxでクロスブラウザ対応

古いIEのシェアの縮小

日本におけるデスクトップブラウザ別シェア

(2016/4~2016/6 [Stat Counter](#) 調べ)



Flexboxでクロスブラウザ対応

IE 9でもFlexboxを使いたい

- ・ ポリフィル 「**flexibility.js**」
- ・ ブラウザによっては異なる挙動を
するので注意
- ・ 2016/7/28現在では最新バージョンは2だが、
バグがあるため1を使う

<https://github.com/jonathantneal/flexibility>

Flexboxでクロスブラウザ対応

Flexibility.jsの読み込み(HTML)

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

displayに-js-を設定する

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

Flexboxでクロスブラウザ対応

The screenshot shows a web browser window with a blue header bar. The address bar displays the URL <http://10.0.1.7:3000/complete.html>. The title bar says "Document". The browser interface includes standard buttons for back, forward, search, and refresh, along with icons for home, star, and settings.

The main content area displays a grid of six animal images, each in its own card-like container:

- animal1**: A close-up of a light brown rabbit's face.
- animal2**: A close-up of a red fox's face, looking slightly to the side.
- animal3**: A close-up of a lion's face, showing its mane.
- animal4**: A white sheep grazing in a field.
- animal5**: A large brown Cape buffalo with prominent curved horns.
- animal6**: A close-up of a cheetah cub's face.

Each animal card contains some placeholder text:

- animal1:

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Harum, repellendus.
- animal2:

Cum dolore id nemo quas recusandae repellendus totam vel voluptates?
- animal3:

Animi eum labore rem? Atque libero nihil vel vero vitae!
- animal4: (No text)
- animal5: (No text)
- animal6: (No text)

At the top right of the main content area, there are three menu links: [menu1](#), [menu2](#), and [menu3](#).

Flexboxのまとめ

Flexboxのまとめ

1. Flexboxはボックスレイアウト用CSS3
2. 横並び・縦並びの豊富なオプション
3. Autoprefixerとポリフィルを駆使して
クロスブラウザ対応

webデザイナーが今覚えておくべき Flexboxとレスポンシブイメージ

休憩時間

#ヒカラボ

Section 3

画像表示の新常識 レスポンシブイメージ



#ヒカラボ

レスポンシブイメージ

レスポンシブwebデザイン用の画像

取り扱いを定めた**HTML5.1の技術**

- ・ 閲覧環境に応じた画像の表示
- ・ アートディレクションの取り扱い

Section 3-1

今までの画像表示の問題

今までの画像表示の問題

メインビジュアルの挿入



[menu1](#) [menu2](#) [menu3](#)



animal1



Lorem ipsum dolor sit amet,
consectetur adipisicing elit.
Harum, repellendus.

animal2



Cum dolore id nemo quas
recusandae repellendus totam vel
voluptates?

animal3



Animi eum labore rem? Atque
libero nihil vel vero vitae!

animal4

animal5

animal6

今までの画像表示の問題

デバイスごとの画像出し分け

モバイル
(Retina)



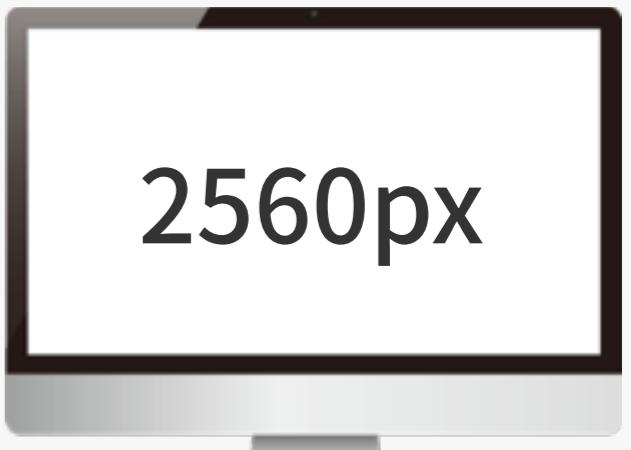
タブレット
(Retina)



ノートパソコン



デスクトップ
パソコン



small.jpg



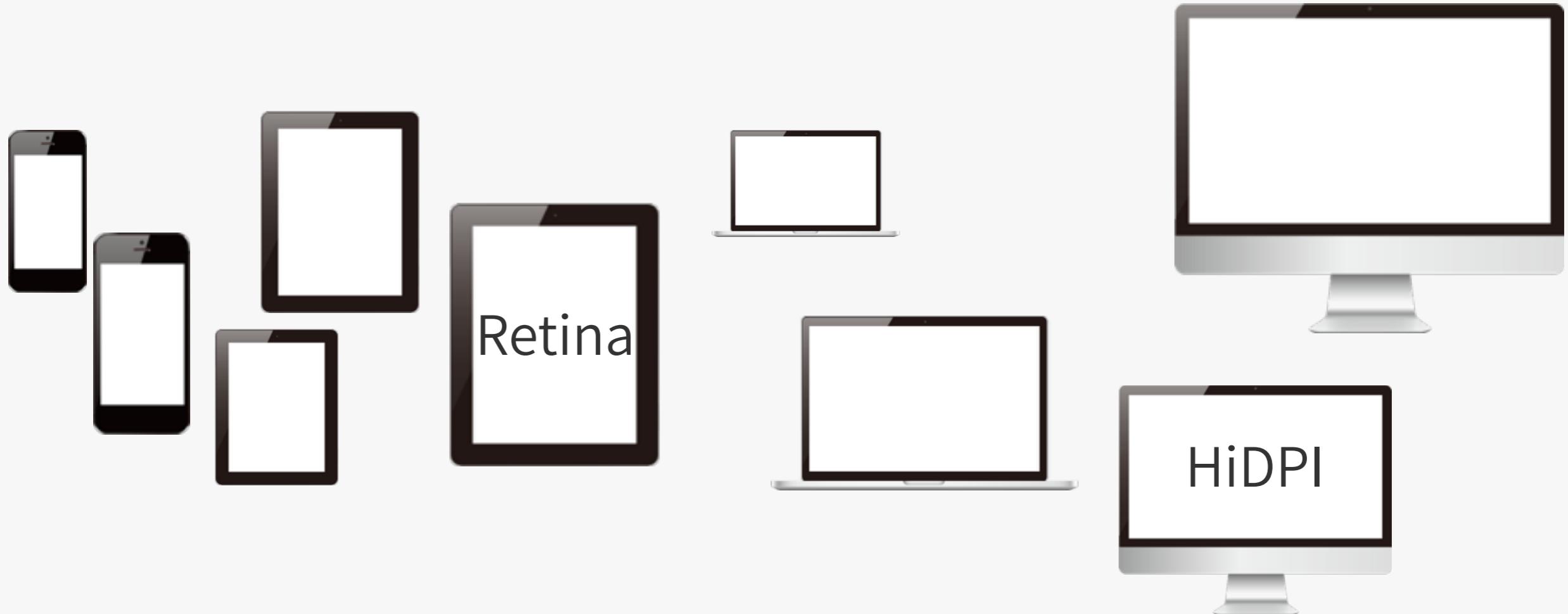
medium.jpg



large.jpg

今までの画像表示の問題

デバイスごとの画像出し分け



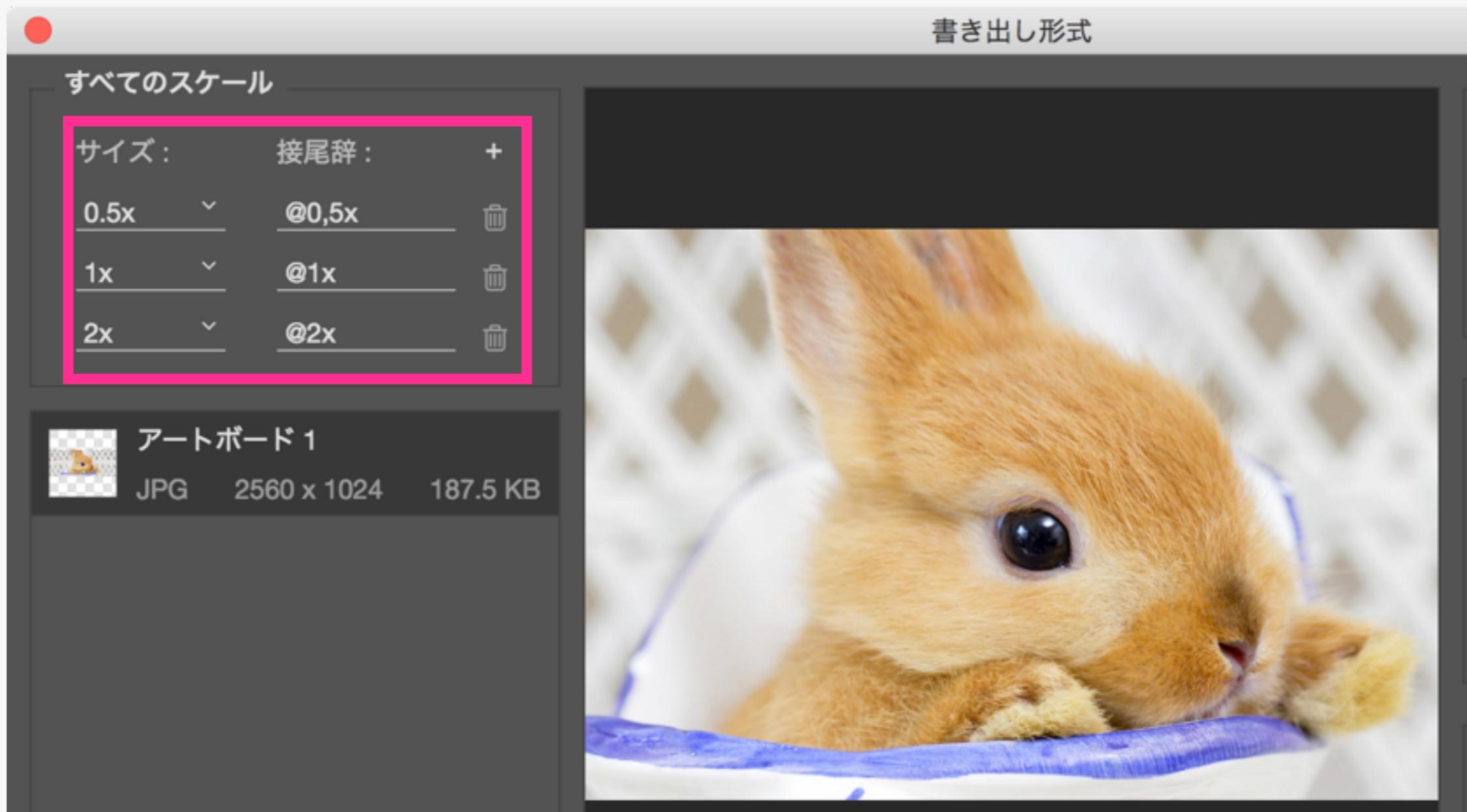
small.jpg

medium.jpg

large.jpg

今までの画像表示の問題

画像ファイルの準備は容易



Photoshop CC 2015の複数スケール書き出し

今までの画像表示の問題

CSSで画像出し分け処理：煩雑

```
#mainVisual {  
    background-image: url(../images/large.jpg);  
}  
  
@media (max-width: 1279px)  
        and (min-resolution: 2dppx) {  
    #mainVisual {  
        background-image: url(../images/medium.jpg);  
    }  
}  
  
@media (max-width: 639px) {  
    #mainVisual {  
        background-image: url(../images/small.jpg);  
    }  
}
```

今までの画像表示の問題

JavaScriptで画像出し分け処理：煩雑

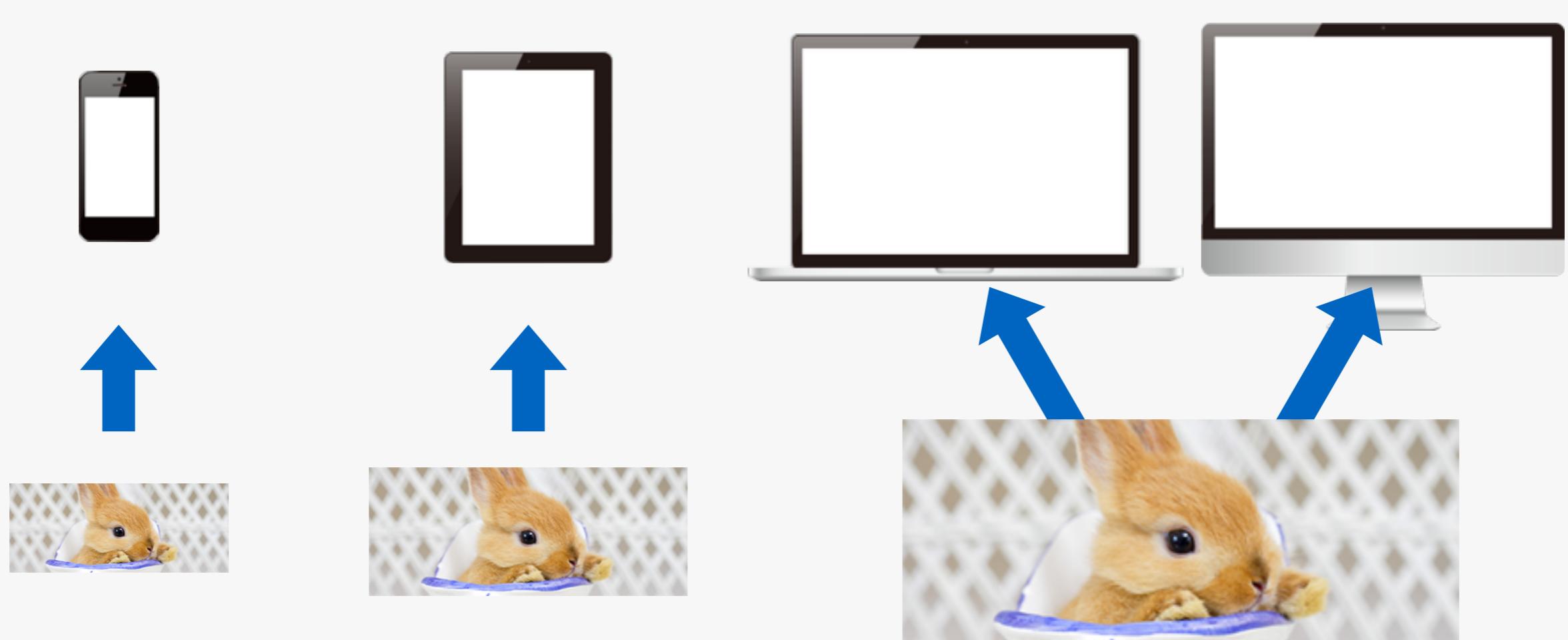
```
var filePath = "";  
  
if (window.innerWidth >= 1280)  
    filePath = "large.jpg";  
else if (window.innerWidth >= 640  
    && window.devicePixelRatio >= 2)  
    filePath = "medium.jpg";  
else  
    filePath = "small.jpg";  
  
$("#mainVisual")  
.html('')
```

今までの画像表示の問題

環境ごとの画像出し分けの問題点

HTMLのimgだけでは実現できず、

CSSやJavaScriptを使う必要があった



Section 3-2

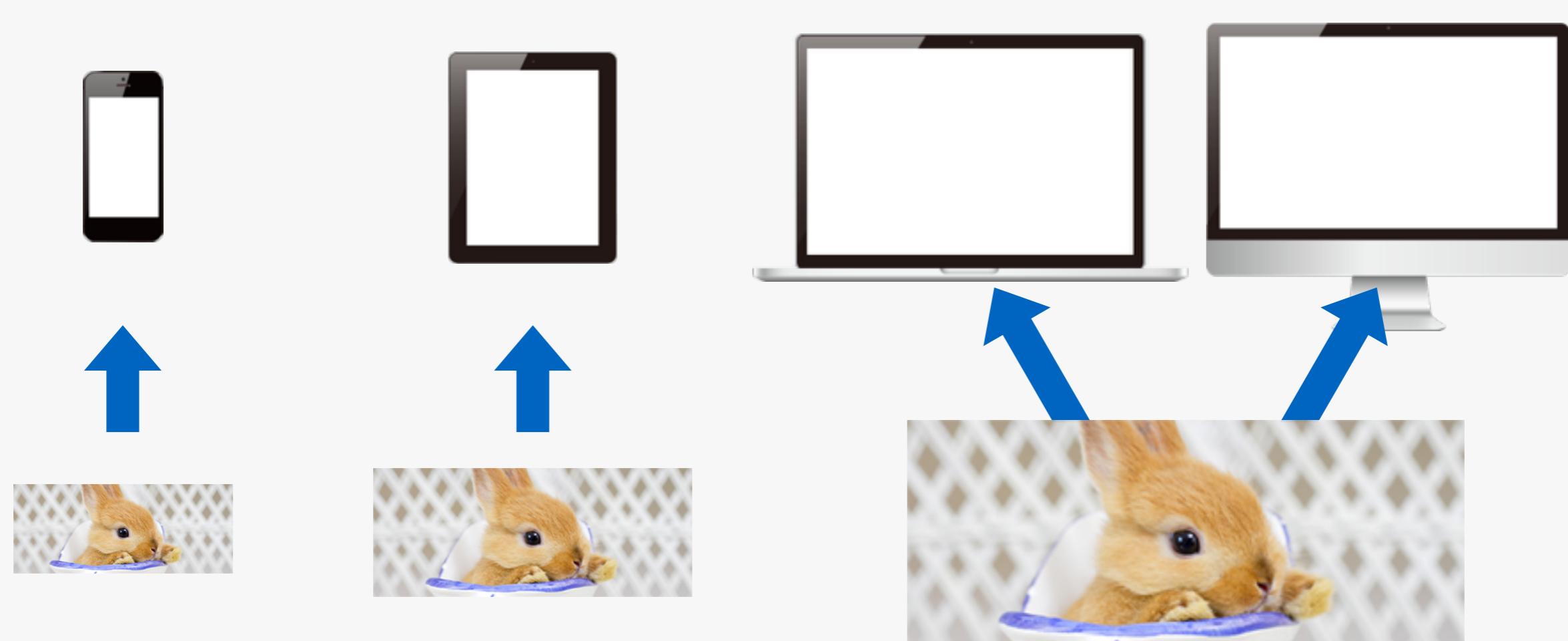
レスポンシブイメージでの 画像出し分け

レスポンシブイメージで画像出し分け

レスポンシブイメージの機能

CSS・JavaScriptは不要、

HTMLのimgタグのみで実現可能



レスポンシブイメージで画像出し分け

HTMLタグに3つの情報を記述する

- ・ 画像ファイルのパス
- ・ 画像ファイルの横幅
- ・ 表示したい画像の横幅

レスポンシブイメージで画像出し分け

srcset属性

- ・ 画像ファイルのパスと横幅を記述する
- ・ src属性のレスポンシブイメージ版
- ・ パスと横幅の間に半角スペース

```
srcset="small.jpg 640w,  
medium.jpg 1280w,  
large.jpg 2560w"
```

small.jpg→640px / medium.jpg→1280px / large.jpg→2560px

レスポンシブイメージで画像出し分け

sizes属性

- ・ 画面上に表示する画像の横幅
- ・ width属性のレスポンシブイメージ版
- ・ メディアクエリーと組み合わせが可能

```
sizes="(max-width: 1280px) 100vw,  
1280px"
```

ディスプレイ横幅1280px以下→画面いっぱい
それ以外では1280px

レスポンシブイメージで画像出し分け

Live Coding

レスポンシブイメージで画像出し分け

注意点

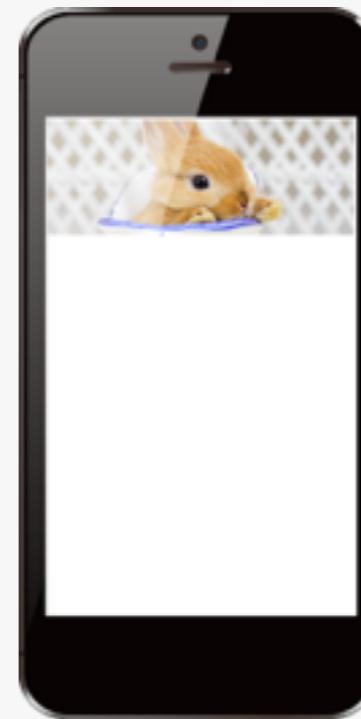
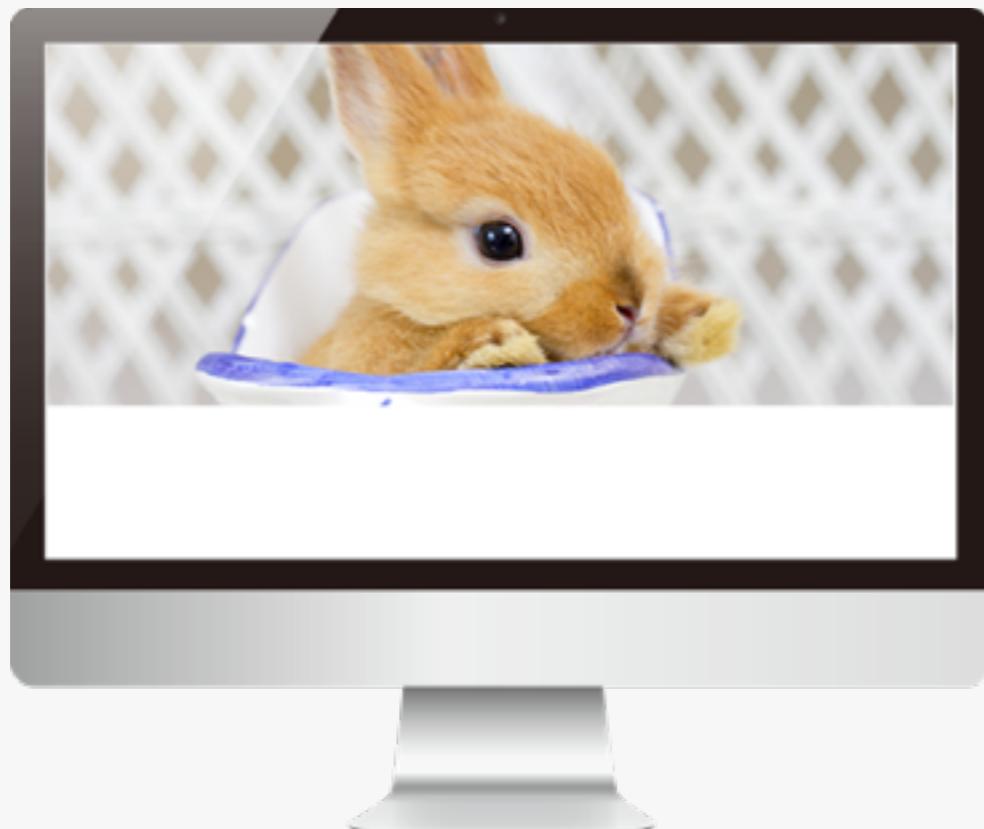
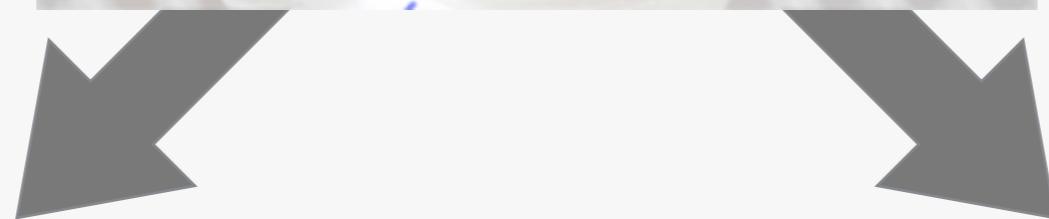
- ・ ブラウザによって挙動が異なる
 - ・ Chrome : キャッシュを使う
 - ・ Microsoft Edge : キャッシュを使う
 - ・ Firefox : キャッシュを使わない
 - ・ Safari : キャッシュを使う

Section 3-3

画像の最適な「見え方」

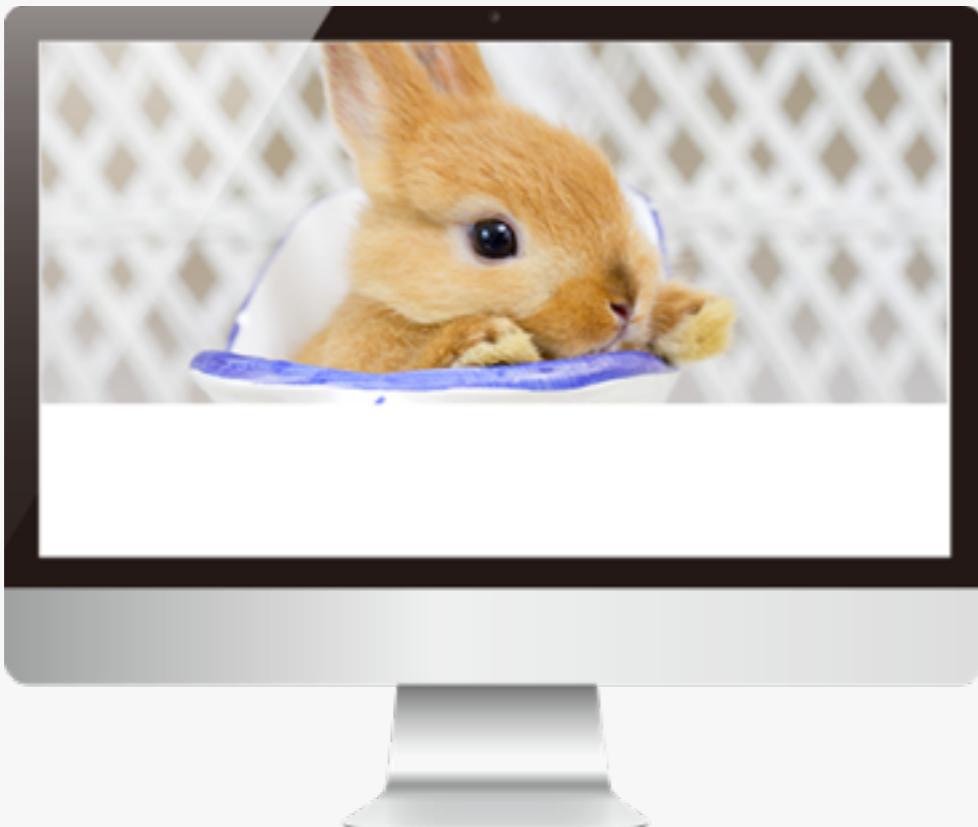
画像の最適な「見え方」

デバイス毎に同じ見た目の画像を使う場合

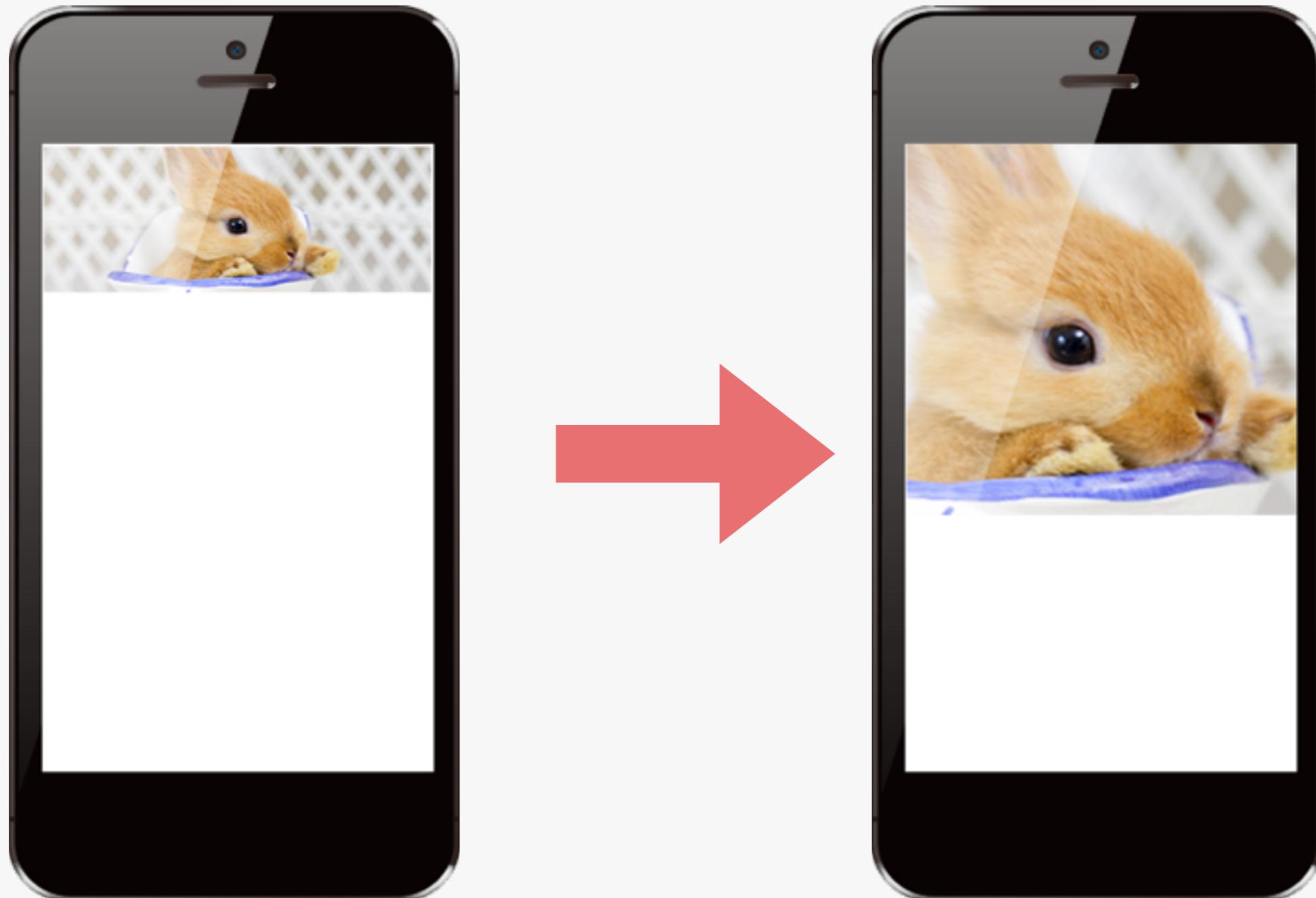


画像の最適な「見え方」

デバイス毎に違う見た目の画像を使う場合



画像の最適な「見え方」



画像の最適な「見え方」

アートディレクション

- ・ デバイス毎に違う画像を出し分ける
- ・ HTML5.1の新要素「picture」を使う



Section 3-4

pictureで実現する アートディレクション

pictureで実現するアートディレクション

picture要素

- ・ アートディレクションを実現する要素
- ・ メディアクエリーに応じて画像を
強制的に切り替える

pictureで実現するアートディレクション

picture要素

1.source要素

- ・ メディアクエリ(**media**)と画像パス(**srcset**)
- ・ 複数指定可能

2.img要素

- ・ 必須指定

```
<picture>
  <source media="(max-width:400px)" srcset="sp.jpg">
  <source media="(max-width:600px)" srcset="tab.jpg">
  
</picture>
```

pictureで実現するアートディレクション

Live Coding

Section 3-5

レスポンシブイメージの クロスブラウザ対応

レスポンシブイメージのクロスブラウザ対応

srcset・sizes属性のブラウザ対応状況

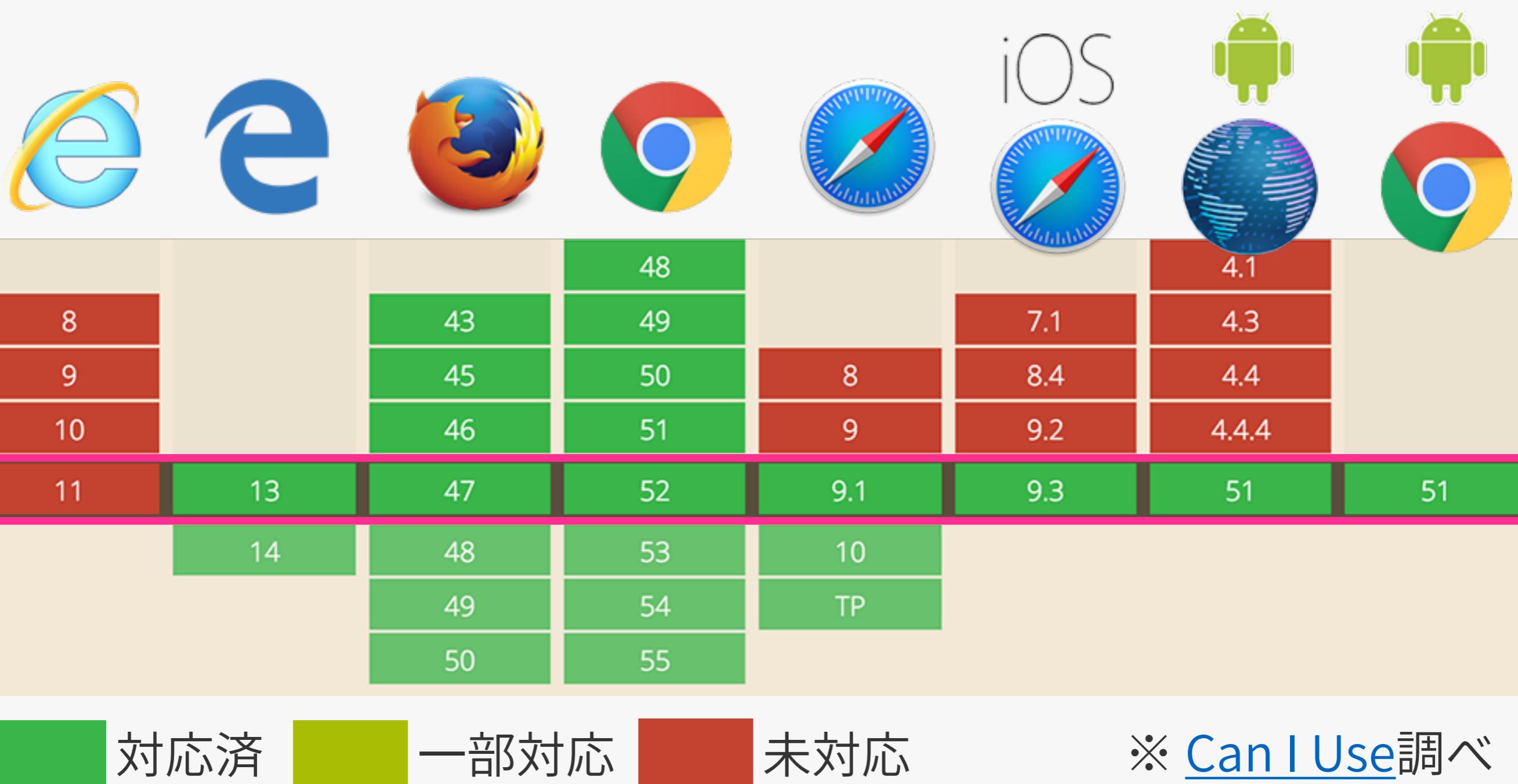
主要ブラウザは全て対応済



レスポンシブイメージのクロスブラウザ対応

picture要素のブラウザ対応状況

主要ブラウザは全て対応済



レスポンシブイメージのクロスブラウザ対応

未対応ブラウザでもsrcsetやpictureを使うための「picturefill.js」

picturefill.jsの読み込み(HTML)

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

<https://github.com/scottjehl/picturefill>

レスポンシブイメージのクロスブラウザ対応

未対応ブラウザでもsrcsetやpictureを使うための「picturefill.js」

IE 9でも対応したい場合、

sourceタグを認識させるために

videoタグが必要

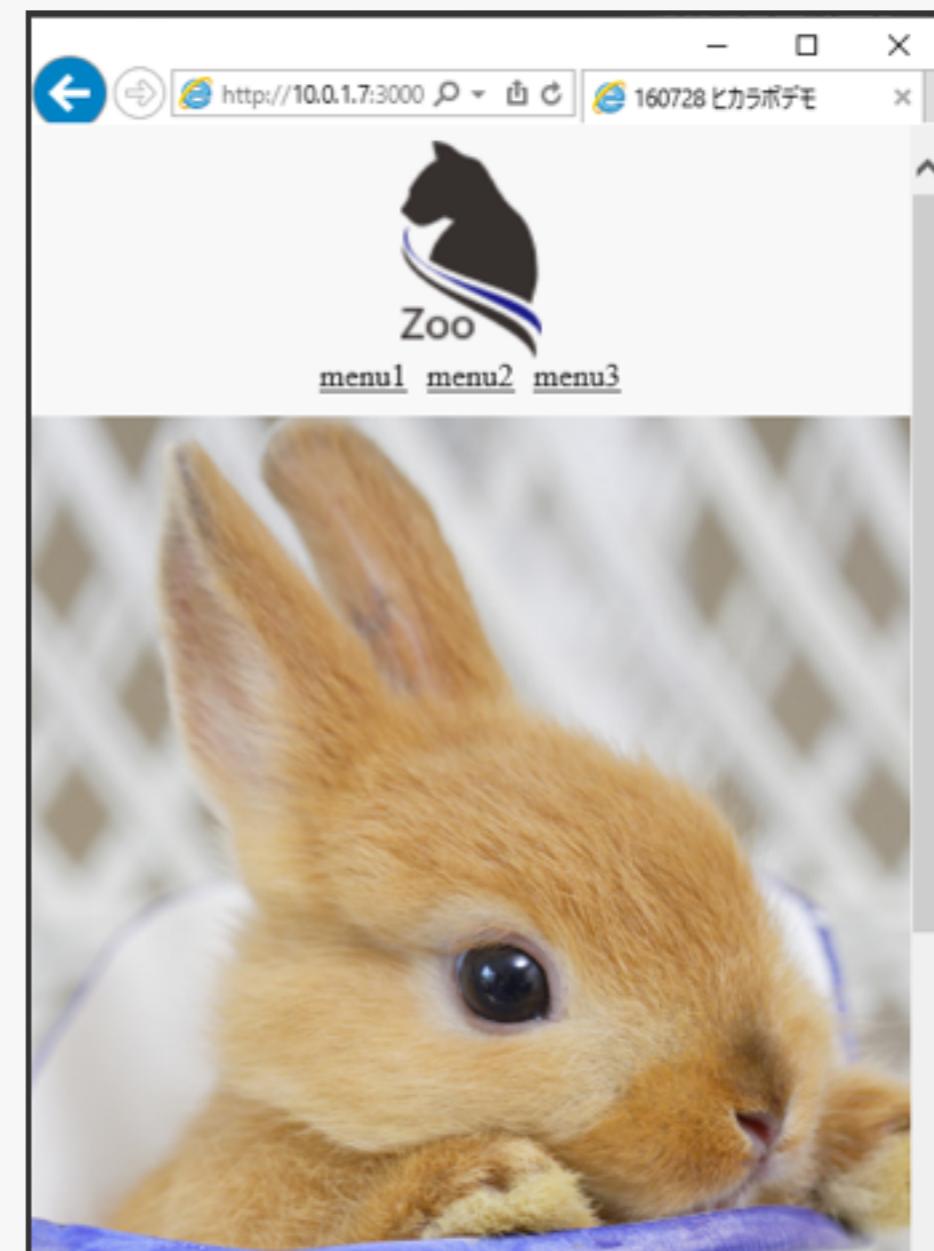
```
<picture>
  <!--[if IE 9]><video style="display: none;"><![endif]-->
  <source>
  <source>
  <!--[if IE 9]></video><![endif]-->
  <img>
</picture>
```

レスポンシブイメージのクロスブラウザ対応

IE 11で動作している様子



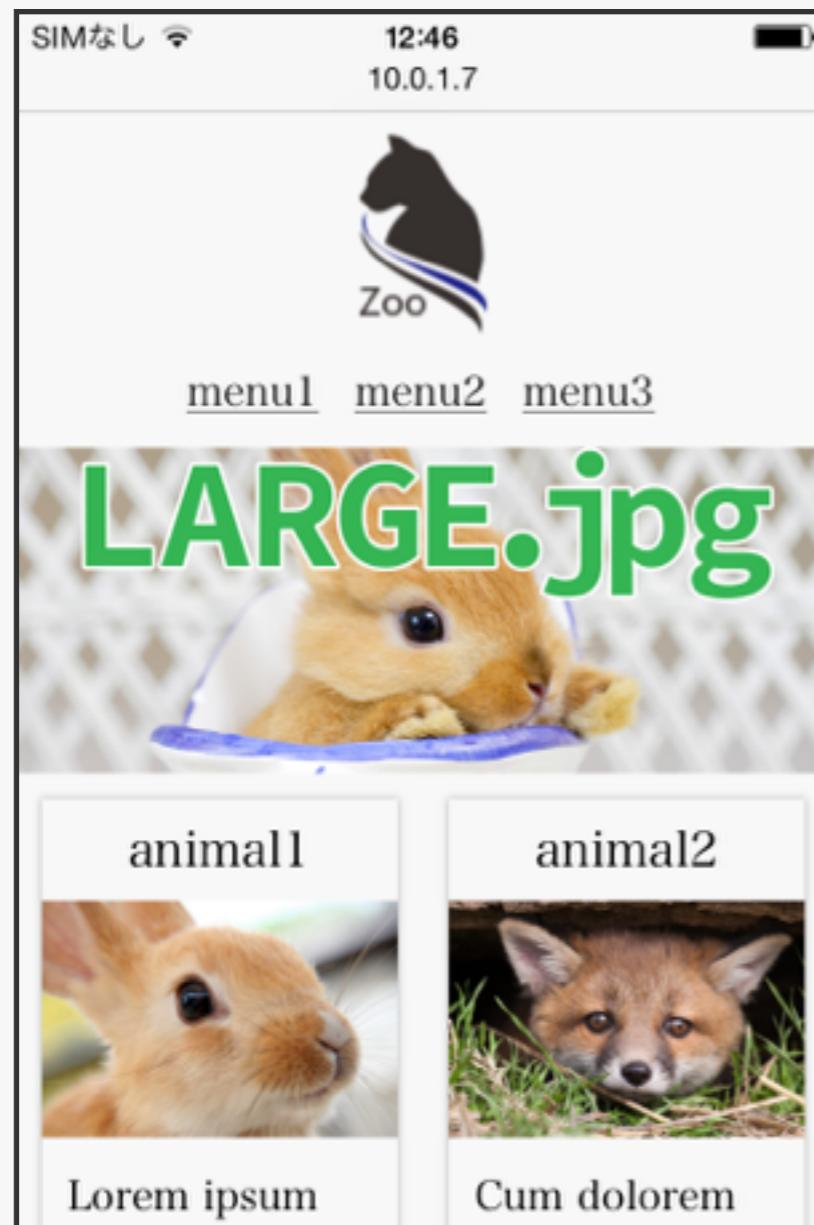
picturefill.jsなし



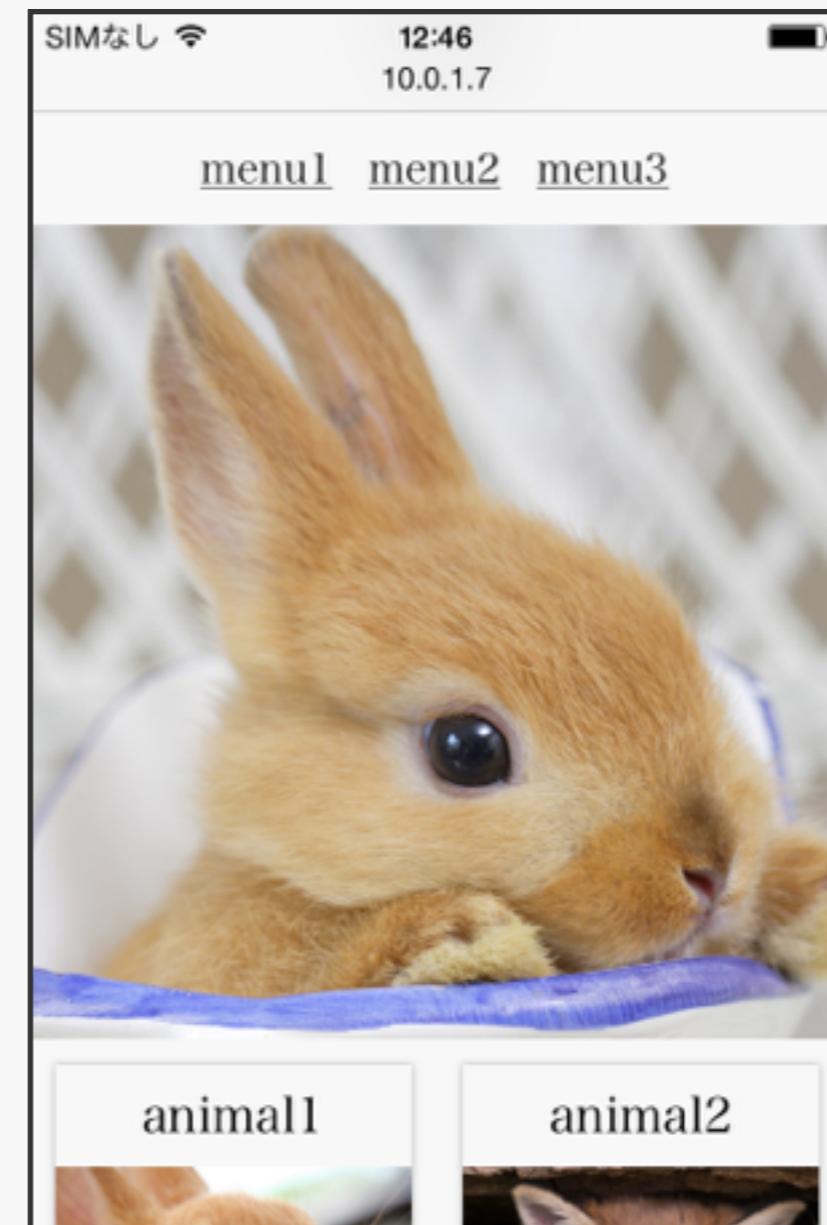
picturefill.jsあり

レスポンシブイメージのクロスブラウザ対応

iOS 7 Safariで動作している様子



picturefill.jsなし



picturefill.jsあり

レスポンシブイメージ まとめ

レスポンシブイメージのまとめ

1. レスポンシブなレイアウトを作るための
画像表示の仕組み
2. 見た目の**同じ**画像出し分け
→**srcset**属性と**sizes**属性
3. 見た目の**違う**画像を出し分け
→**picture**要素

最後に



最後に

- ・ 多様なユーザー環境に対応すべき時代
- ・ ブラウザの進化により使えるようになった便利な技術
 - 1. 「Flexbox」
シンプルな記述で
豊富なボックスレイアウトが可能なCSS3
 - 2. 「レスポンシブイメージ」
HTMLタグのみで
レスポンシブな画像を取り扱うHTML5.1

ご清聴ありがとうございました

ご質問等はTwitterでお気軽に寄せください！



ICS 鹿野 壮
@tonkotsuboy_com