



Responsive Web Design

~ Basic understanding ~

～自己紹介～



Daisuke Yamazaki

レスポンスWebデザイン「超」実践デザイン集中講義



～自己紹介～

Job :

- U-SYS (web professional fellow)
 - デジタルハリウッド (ソーシャルアプリクリエイター専攻)
 - デジタルハリウッド (オンライン : PHP、Webアプリ開発)
 - Web関連のセミナー (全国各地、ヒカ☆ラボ)
- Microsoft MVP (Bing Maps Development)

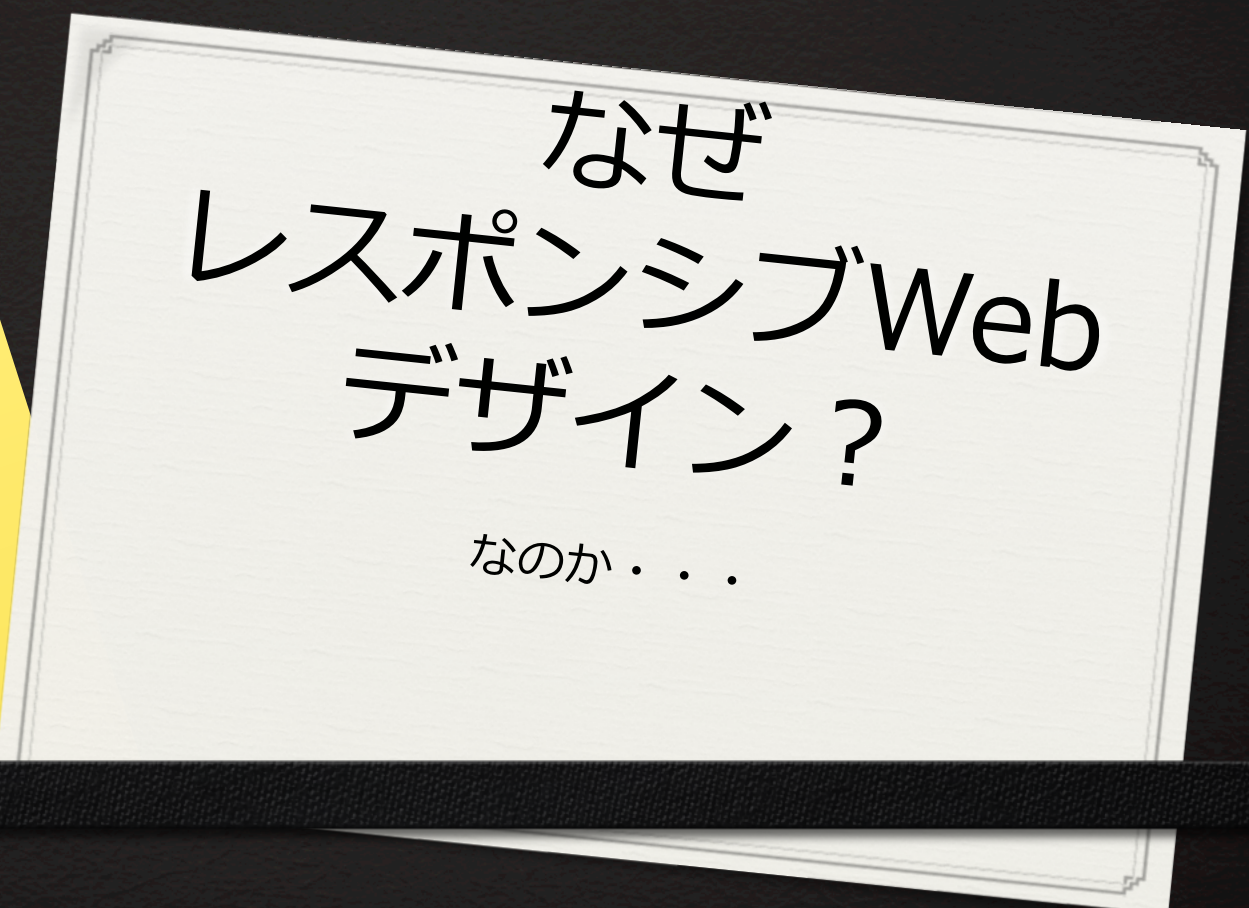


HTML5ビギナーズ
beginners
html5j



執筆&記事 :

日経ソフトウェア、@IT、ITpro、Software Designなど、IT関連のメディアで情報発信中。@IT「HTML5アプリ作ろうぜ」を連載。日経PC21「名作フリーソフトを訪ねて」でも自身の開発したアプリが選出。
HTML5アプリ「impress.js design」ChromeWebストアにて公開中。



なぜ
レスポンス性Web
デザイン？

なのか・・・

従来求められてるWebサイト

0 情報を正しく表示するサイト

0 目的の情報を探しやすいサイト

0 集客を上げられるサイト
SEO、広告、ソーシャルメディアとの連携

0 バリアフリーなサイト
視覚障害者（画面を見れない）、身体障害者（マウス使えない）、
高齢者



今これから求められるWebサイト

「マルチデバイス対応」



- 0 「さまざまなデバイスで正しく表示・利用できるサイトを制作」

先ほどのスライドをに追加して...

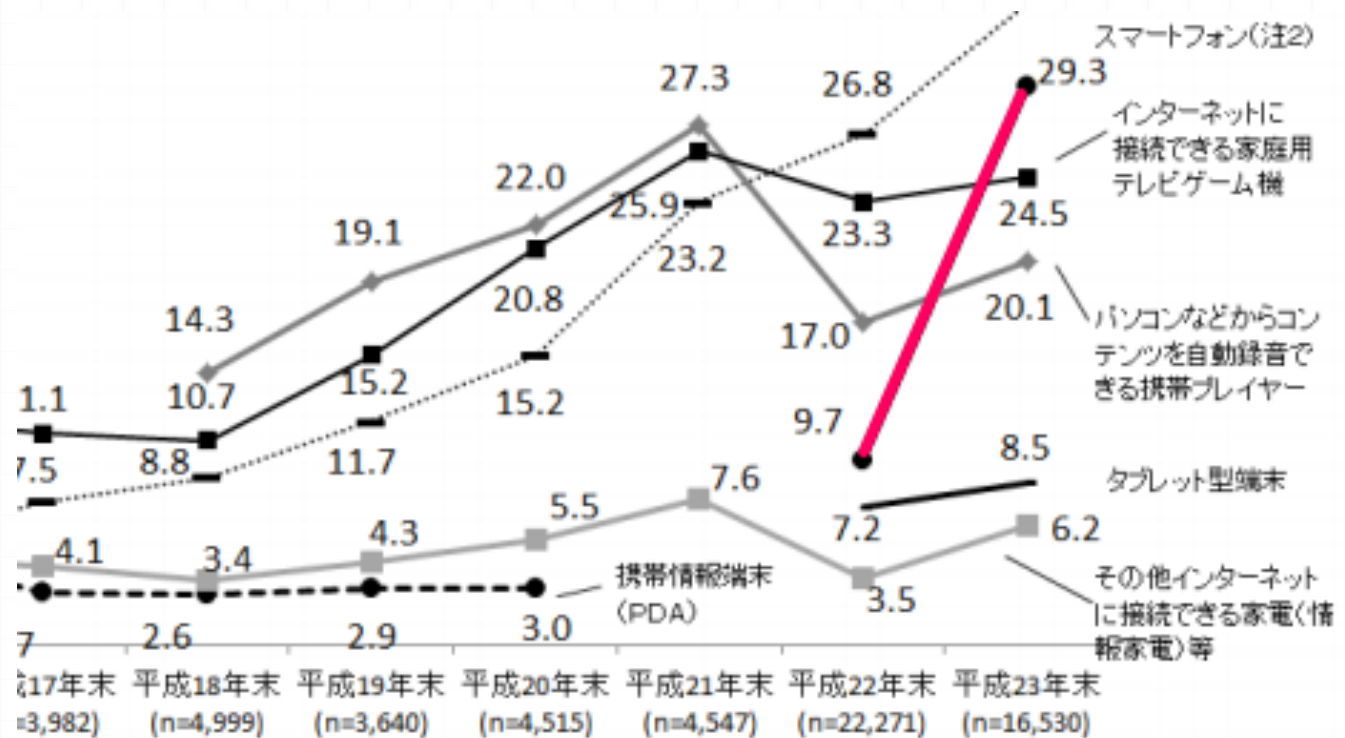
今これから求められるWebサイト

- 0 情報を正しく表示するサイト
- 0 目的の情報を探しやすいサイト
- 0 集客を上げられるサイト
- 0 バリアフリーなサイト
- 0 マルチデバイス対応
PC、携帯電話+スマートフォン、タブレットなど



やること増えたね

モバイル端末からの閲覧増



(8) 主な情報通信機器の普及状況(世帯)

情報通信機器の普及状況を機器別にみると、「携帯電話・PHS」及び「パソコン」の世帯普及率は、それぞれ94.5%、77.4%となっている。また、「携帯電話・PHS」の内数である「スマートフォン」は、29.3%(対前年比19.6ポイント増)と急速に普及が進んでいる。

現在のWebでは不十分なのか？ モバイル向けサイトの必要性

今はパソコンは持っていないなくても・・・



スマートフォン
(携帯電話フィーチャーフォンも)

タブレット
(iPad, Nexus, Kindle…)

をってる人の比率は日々増加してます。

現在のWebでは不十分である理由 モバイル向けサイトの必要性

パソコンではなく、

スマートフォン・タブレット



をWeb閲覧のメインデバイスとして使う時代に来ています。

スマートデバイスを優先でサイトを構築する時代なのかも知れません。

従来のマルチデバイス対応方法

デバイス毎にディレクトリを分ける

URL :

www.domainname.com/**A**/



サイトA (PC向け)

www.domainname.com/**B**/



サイトB (タブレット向け)

www.domainname.com/**C**/



サイトC (スマートフォン)

広告例)

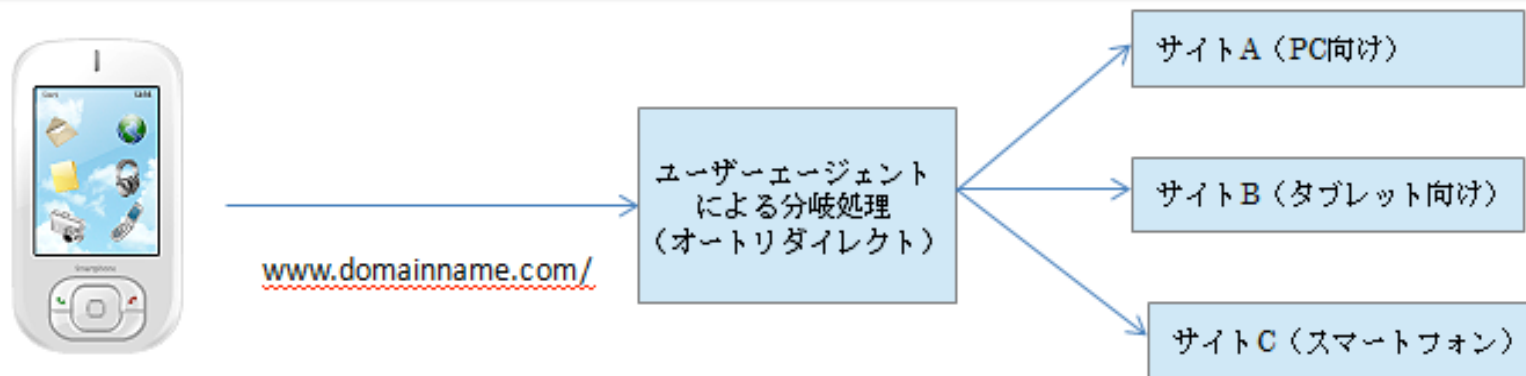
http://www.●●●/...

http://www.●●●/...

http://www.●●●/...

従来のマルチデバイス対応方法

サーバサイド、クライアントスクリプトにて切り分け



```
<?php /* UserAgent Switch */
$ua=$_SERVER['HTTP_USER_AGENT'];
if((strpos($ua,'iPhone')!==false)||(strpos($ua,'iPod')!==false)||
(strpos($ua,'Android')!==false)) {
    header("Location:/sp/index.php");
    exit();
}
?>
```

ユーザーエージェント (自動振り分けの注意点)

- 新たなデバイスが発売された場合
(デバイスのユーザーエージェントを調査し振り分け処理を確認・追加する作業が必要)
- 必要なスクリプトなどの技術を習得している人材が必要

docomo
XPERIA
A SO-04E
使いやすい、の先端へ。
Your Best Choice.



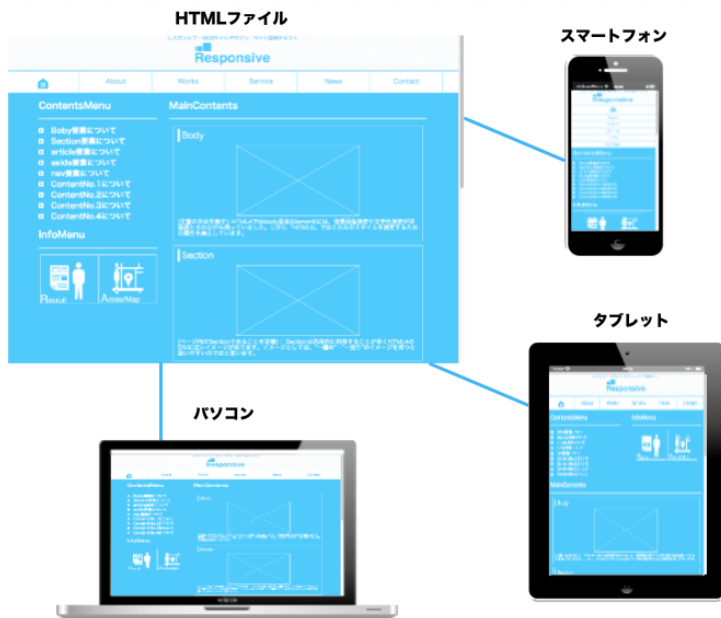
iPhone 5 (S)
進み続けるもの。



iPhone 5 (C)
この色は、あなたです。



ワンソースでマルチデバイス対応 (レスポンスWebデザイン)



- Javascript のようなプログラミング (スクリプト) 知識は必要ない
- CSSの知識で作成できる
- ワンソース (HTMLとCSS) で作成可能

ブラウザのスクリーンサイズ (解像度) を基準にレイアウトを調整することができるのでマルチスクリーンに対応した Webサイトを制作できるのが特徴です。

デザイナーの方には是非覚えてほしい手法です

レスポンシブWebデザイン 誰が？いつ頃から？

レスポンシブWebデザイン提唱者：

ETHAN MARCOTTE氏（イーサン・マルコット氏）

2010年5月25日にブログで紹介したマルチデバイス対応のWebサイト制作の手法「RWD」です。

ARTICLES COLUMNS BLOG TOPICS

A LIST AFANI



Issue No
306

Responsive Web Design

by **ETHAN MARCOTTE** · May 25, 2010

Published in *CSS, Layout & Grids, Mobile/Multidevice, Responsive Design, Interaction Design* · [103 Comments](#)

“The control which designers know in the print medium, and often desire in the

レスポンスWebデザイン どんなことができるの？

レスポンスWebデザイン提唱者：
ETHAN MARCOTTE氏（イーサン・マルコット氏）

記事内容： *Media Query*を使ってレイアウトし、例えば1カラムを2カラムに流動的に変形させる事を指し、閲覧されたそのデバイスの解像度によって柔軟に、シンプルに最適化（フルードグリッド、フルードイメージ）させる。

The image displays three sequential screenshots of a website design in Adobe Dreamweaver CS6, illustrating how the layout adapts to different screen sizes. Each screenshot is titled "Adobe HTML5 特設サイト Dreamweaver CS6 と HTML5 で始めよう" (Adobe HTML5 Special Site Dreamweaver CS6 and HTML5 Let's Start). The first screenshot shows a desktop view with a single column layout. The second screenshot shows a tablet view with a two-column layout. The third screenshot shows a mobile view with a single column layout. The design features a green header and footer, and a main content area with various articles and images. The text in the screenshots is in Japanese, and the overall layout is clean and modern.

レスポンスWebデザイン メリット・デメリット

メリット

- 画面を各デバイスに合わせてソースを複数作る必要がない、作業工数の軽減に繋がる
- ワンソースで管理するのでメンテナンスがしやすくなる
- 1つのURLのみにアクセスを集中できる
(各デバイスごとにTOPページを分けた場合、ページへのアクセスが分散化される)
- 1つのURLだけなので、告知・宣伝等がやりやすくなる
- HTML/CSSなどのクライアント技術のみで実装できるためWebデザイナーが実装できる

レスポンシブWebデザイン メリット・デメリット

デメリット

- 芸術的な見せ方、拘りのあるビジュアルなサイトにはデメリットの可能性あり
- ワンソースで各デバイスに対応するため、設計/デザインの調整に時間がかかる
- ワンソースで対応するため、より高度で複雑な設計となる可能性がある
- 全デバイスに必要なソースコードも読み込みます、必要のない要素も一緒に読み込まれてしまい、ページの読み込み時間やレンダリング速度に影響する可能性も考慮する必要があります。



テクニカル解説

MediaQuery、ブレイクポイント

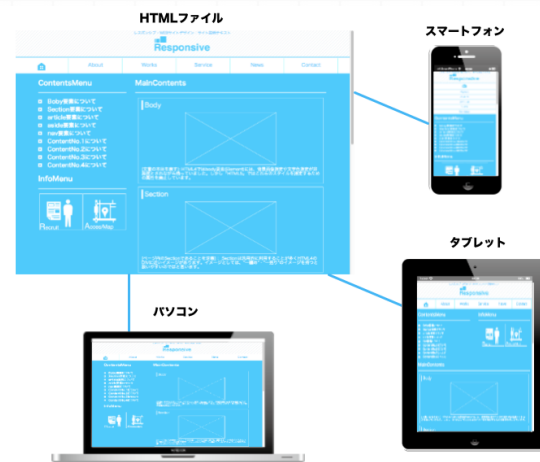
Media Queryとは？

スクリーンサイズ（デバイスサイズ）によりCSSスタイルを複数指定することを可能にします。このCSSを複数指定することを可能にしたのがCSS3の「MediaQueries」です。

特徴：

デバイスごとの細かい配慮／工夫は向いていません。

特定のデバイスにだけ特別な工夫をしようと思った場合に、他のデバイス（CSSスタイル）への影響も出る可能性が考えられます。

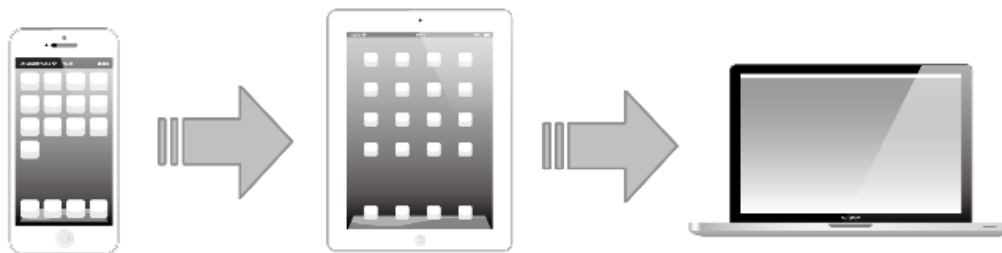


何を基準にして設計・構築する？

スマートフォンサイトから考える

まずスマートフォンを基準に考えることの方が、作る上で簡単はずです。各デバイスの画面表示をスマートフォン用のスタイルを最初に定義し、タブレット、パソコン用の順番でスタイルを決めた方が設計・構築はスムーズかと思います。

(例：スマートフォン480px → タブレット768px → パソコン980px)



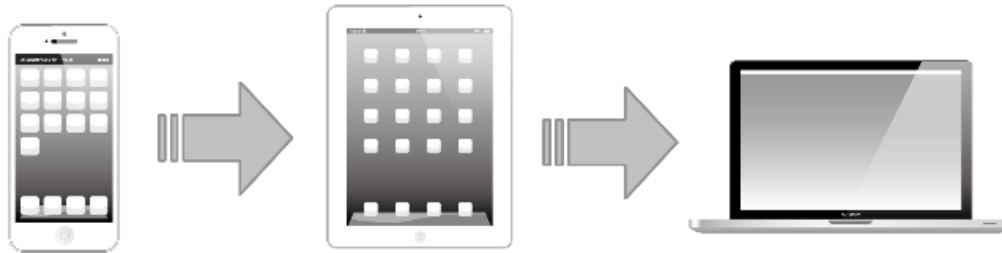
ブレイクポイントとは

ブレイクポイントを決めておく

メディアクエリを使用した方法で画面サイズ別にCSSスタイリングする場合

(例 : スマートフォン320px → タブレット788px → パソコン880px)

と画面サイズの境目を先に決め手します。この決めた画面サイズに合わせてCSSスタイルを切り替える事ができ、この決めた画面サイズによる切り替えのポイントを、「ブレイクポイント」と呼びます。



Media Query 書き方 1

@media ひとつのCSSファイル内で振り分け

```
@media screen and (min-width: 481px) {  
  body {...}  
}  
@media screen and (min-width: 768px) {  
  body {...}  
}
```

Link 複数CSSファイルをlink要素で読み込む

```
<link href="sp.css">  
<link media="screen and (min-width: 481px)" href="tablet.css">  
<link media="screen and (min-width: 768px)" href="desktop.css">
```

@import 複数CSSファイルをimport要素で読み込む

```
@import url("sp.css");  
@import url("tablet.css") only screen and (min-width: 481px);  
@import url("desktop.css") only screen and (min-width: 768px);
```

Media Query 書き方 2

@media が使いやすい

<link…

@import…

複数CSSファイルに分けてブレイクポイント（スマホ、タブレット、パソコン）を管理するとスタイル管理しやすくなりそうですが、

@mediaの方法で単一CSSファイルにした方が、

「 HTTPリクエスト数を節約できる 」
特典があります。

スマートデバイスの3G回線にもよい。



実際の使い方

レスポンシブをIE6~8で使う方法

css3-mediaqueries.js

- 、CSS3の@mediaクエリをサポートしていないブラウザでもサポートしてるかのように見せるスクリプトです。CSSを解析して、ブラウザのサイズを変更したときにもリアルタイムに@mediaクエリを適用します

respond.js

メディアクエリーに対応していないIE8以下のブラウザでもメディアクエリーを利用できるようにするスクリプト。非常に軽量なライブラリーで、古いIEでも簡単にメディアクエリーに対応できます。

ダウンロードURL : <https://github.com/scottjehl/Respond>

レスポンスィブをIE6~8で使う方法

css3-mediaqueries.js

html5shiv.js

```
<!--[if lt IE 9]>  
  
<script src="js/html5shiv.js"></script>  
  
<![endif]-->
```

css3-mediaqueries.js

```
<!--[if lt IE 9]>  
  
<script src="js/css3-mediaqueries.js"></script>  
  
<![endif]-->
```

実際のコード例

```
1 <!DOCTYPE html>
2 <html lang="ja">
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width">
6 <link rel="stylesheet" href="css/reset.css">
7 <link rel="stylesheet" href="css/base01.css">
8 <title>sample6 | WORKS</title>
9
10 <!--[if lt IE 9]>
11 <script src="js/html5shiv.js"></script>
12 <![endif]-->
13
14 <!--[if lt IE 9]>
15 <script src="js/css3-mediaqueries.js"></script>
16 <![endif]-->
17 </head>
18
19 <body>
20 <div id="stage">
21 <header id="header">
22 <h1 id="top"></h1>
```

viewport

Viewport

viewport指定できること

- モバイル端末ブラウザの表示幅を指定
- ピンチによる、拡大縮小の可否の指定
- 拡大縮小の最大・最少倍率の指定

記載方法

- `<meta name="viewport" content="width=1024">`
※ 例：アップルサイト
- `<meta name="viewport" content="device-width">`
※ 端末画面の幅に合わせる

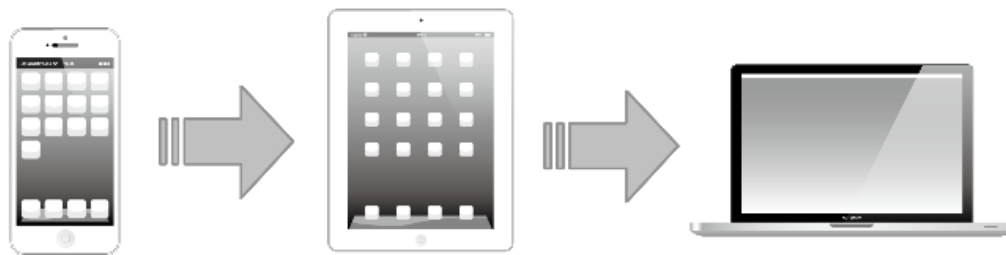
実際のコード例

```
1 <!DOCTYPE html>
2 <html lang="ja">
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width">
6 <link rel="stylesheet" href="css/reset.css">
7 <link rel="stylesheet" href="css/base01.css">
8 <title>sample6 | WORKS</title>
9
10 <!--[if lt IE 9]>
11 <script src="js/html5shiv.js"></script>
12 <![endif]-->
13
14 <!--[if lt IE 9]>
15 <script src="js/css3-mediaqueries.js"></script>
16 <![endif]-->
17 </head>
18
19 <body>
20 <div id="stage">
21 <header id="header">
22 <h1 id="top"></h1>
```

CSS

CSS作成の考え方

CSSを作成する際、
スマートフォンの一番小さい方に合わせて記述する
方法で解説してます。



CSS

タブレット向けのスタイル : 481px ~

```
@media only screen and (min-width: 481px) { //*****[ 1 :Start ]*****  
  
}
```

PC向けのスタイル : 769px~

```
@media only screen and (min-width: 769px) { //*****[ 2 :Start ]*****  
  
}
```

CSS

CSSコードサンプルを見ましょう！

img

Img要素 (フルドイメージ)

width・height属性を指定すると比率がおかしくなります

フルドイメージ

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

IE 7以下では縮小された画像が粗く見える。
この問題はimgSizer.jsというjsで対処できる。次ページへ>>

img

imgSizer.js

```
<!--[if lte IE 7.0]>  
    <script type="text/javascript" src="imgSizer.js"></  
script>  
<![endif]-->
```

注意ポイント

デスクトップ上でのブラウザ実行は × です。

http://....

https://....

のように実際にWebサーバ上に上げて確認する必要が有ります。

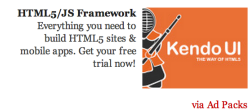
※XMAPPなどPCにWebサーバを作るのはOK！

A white rectangular card with a double-line border is tilted on a black background. A yellow sticky note is attached to the left side of the card. The word "DEMO" is printed in a black serif font on the card.

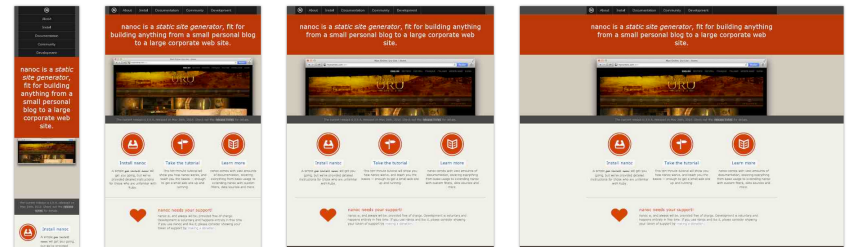
DEMO

参考サイト

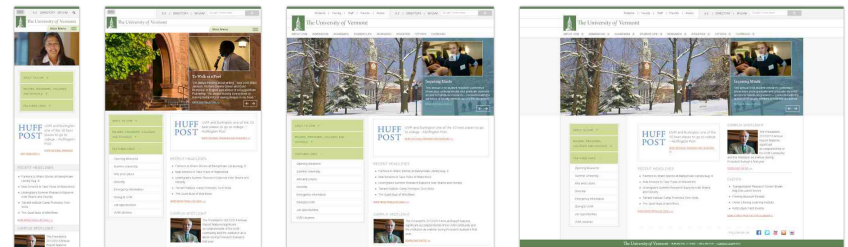
■ Media Queries (<http://mediaqueri.es/>)



nanoc



The University of Vermont



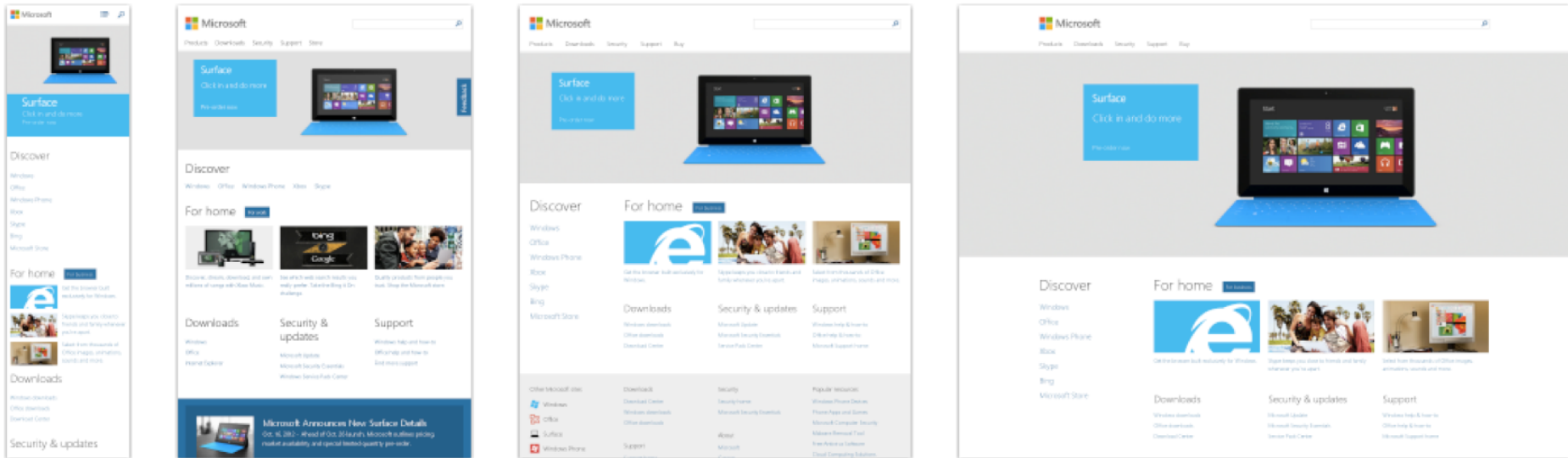
Erik Ford

参考サイト

■ Microsoft (<http://www.microsoft.com/en-us/default.aspx>)

Microsoft

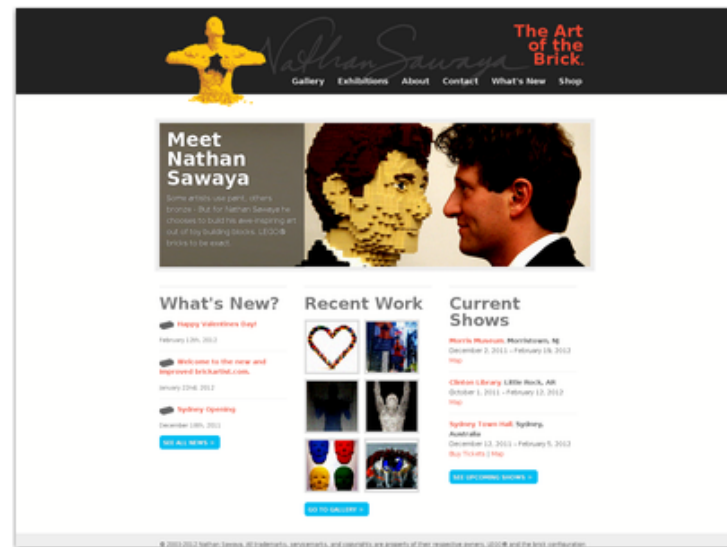
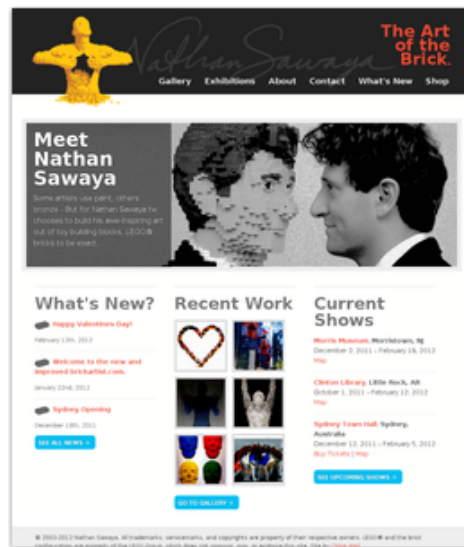
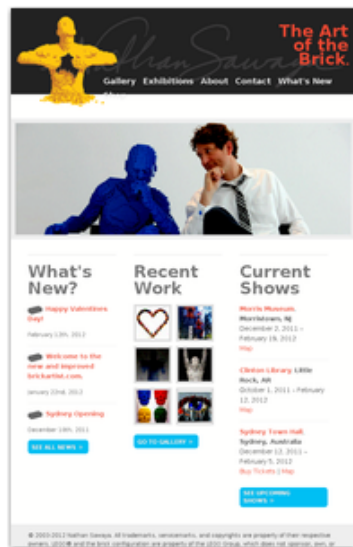
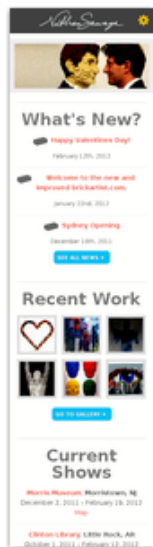
32



参考サイト

■ Nathan Sawaya (http://brickartist.com/)

Nathan Sawaya



キービジュアルの画像がスライド（フェードイン・フェードアウトで切り替わ

参考サイト

- COOP (<http://coworkchicago.com/>)

COOP



右側からパソコン、タブレット、スマートフォンサイズの表示になりますが、スマートフォンのサイズ（左側）が大きくデザインが変わります。

参考サイト

Adobe社

「Dreamweaver CS6 と HTML5 で始めようスマートフォン&タブレット対応」
<http://adobe-html5.jp/>

アドビシステムのHTML5特設サイトもレスポンスWebデザインで作成されています。ブレイクポイントは0~520pxの場合は1カラムレイアウト、520px~920pxの間は2カラムレイアウト、920px~より大きい場合は3カラムレイアウトの表示となります。

ブレイクポイント : 520px, 920px



アンケートにご協力ください！

あなたのフィードバックが、
html5jの活動を支えています！

<http://bit.ly/html5C201311>

アンケートへ答えた方にもれなく記念品プレゼント！
(アンケートの完了画面を1F受付にてご提示ください)

お知らせ:

最終スペシャルセッションの開始時刻が変更になりました(18:30→**18:20**)
10分早まっています。ご注意ください！

