

はじめに

このセッションでは CSS の Hands-on を行います。Hands-on とは体験学習を意味する教育用語です。この Hands-on を通してより多くの体験が得られることを望んでいます。

1st Step: CSSでコーティング(学習時間：2時間)

下記ディレクトリに step1 ~ step3 のサンプルがあります。

<https://github.com/albatrosary/css-handson-21cafe>

2nd Step: 開発環境構築 (学習時間：1時間)

1. Node のインストール
2. Ruby のインストール
3. Git のインストール
4. Sass/Compass のインストール
5. Yeoman のインストール

3rd Step: Sass / Compass でコーティング (学習時間：3時間)

1. Sass とは
2. Compass とは
3. 1st Step を Sass / Compass で実装する

ブラウザは

Google Chrome Canary(<http://www.google.co.jp/intl/ja/chrome/browser/canary.html>)

を使用することを推奨します。このブラウザは Chrome の最新機能を備えていて、開発者や開発段階の機能に興味があるユーザ向けに設計されており、動作が停止することがあります。

サンプルは

<git@github.com:albatrosary/css-handson-21cafe.git>

にあります。各自リポジトリを fork / clone してください。

HTML5 と CSS3

HTML と CSS の歴史

年表

1986 SGML(Standard Generalized Markup Language)

1989 HTML(Hyper Text Markup Language)

1996 CSS1

1997 HTML4.0

1998 XML(eXtensible Markup Language) CSS2

2000 XHTML(eXtensible Hyper Text Markup Language)

2004 CSS2.1

2010 XHTML2.0 打ち切り

2011 CSS3

2012 HTML5.0

HTMLの規格はこの何年か変化が無かったがようやく技術革新がありました。新しいHTMLによって今までとかまったく異なる世界になることでしょう。特に、いままではAdobe Flash や JavaFX、Microsoft Silverlight 等でしか出来なかつたようなことをHTMLとCSSそしてJavaScriptを使うことで可能となります。また、こうすることによりプラグインを不要としています。

HTML5について

HTMLの5回目にあたる大幅な改訂版であり、現在、W3Cによって2014年までの正式勧告を目指して策定が行われています。広義のHTML5として、Application Cache、Web Storage、WebSocket、Geolocation API、XMLHttpRequest Level2などを含む場合があります。また、ビデオやオーディオの再生、ドラック&ドロップなどのAPIも追加されています。

CSS3について

CSS3はCSS2.1の次世代の規格としてW3Cで策定が進められている規格です。現状ではベンダープレフィックスを利用し策定中の機能を利用することができるます。

CSS3のベンダープレフィックス

CSS3では、策定中の機能をブラウザが実験的に先行実装したものについてベンダープレフィックスを付けるという決まりがあります。

ベンダープレフィックス	ブラウザ
-webkit-	Chrome
	Safari
	iOSの標準ブラウザ
	Androidの標準ブラウザ
-moz-	Firefox
-o-	Opera
-ms-	Internet Explorer

例) border-radiusプロパティは、ボックスの4つのコーナーの角丸をまとめて指定する際に使用します。具体的には

```
-webkit-border-radius: 4px;
-moz-border-radius: 4px;
-ms-border-radius: 4px;
-o-border-radius: 4px;
border-radius: 4px;
```

1st Step: CSS でのコーティング

<https://github.com/albatrosary/css-hands-on-21cafe>

を参照してください。

2nd Step: 開発環境構築

ここでは、Web開発に必要となるツールのインストールを行います。様々なツールがありますが、今回は yeoman(<http://yeoman.io/>) という開発ツールを使用します。エディターですが、各々お気に入りのエディターを使用してください。参考までに Sublime Text などはよく使われています。

yeoman は

- Yo
Yo は新しいアプリケーションのひな形を作成します。既に必要なビルド情報が Grunt コンフィグレーションに定義されています。
- Grunt
Grunt は Yeoman や grunt-contrib により作成したプロジェクトのビルド、プレビューそしてテストを行います。
- Bower
Bower はプロジェクトが依存するライブラリを管理するために使用します。Bower を使うことでライブラリを自動的にダウンロードしインストールすることが出来ます。

の3つのツールで構成されている「Web App Development Through Tooling」です。

事前準備として

- Node.js
- Git

必要であれば

- Ruby
- Compass

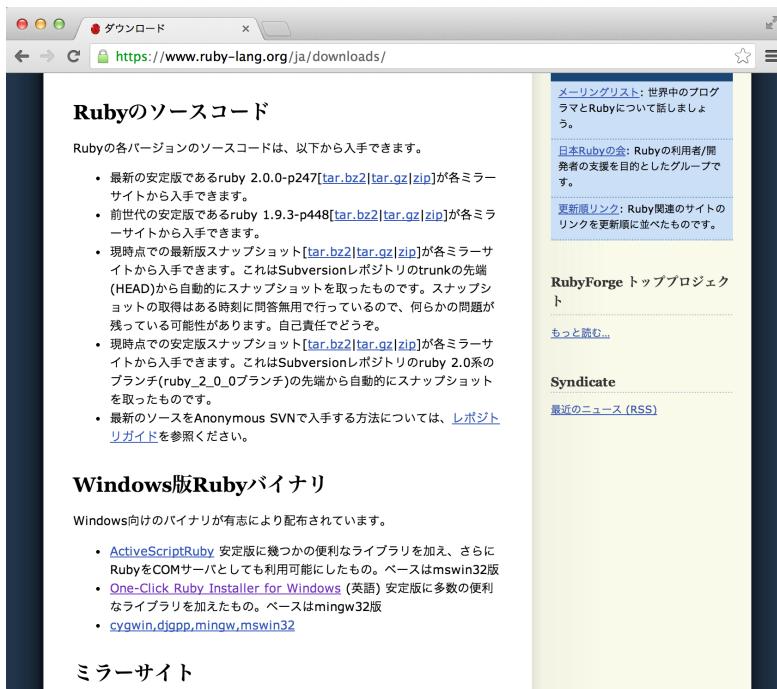
をインストールしておきます。今回は最終目的である Sass/Compass のハンズオンを Yeoman を通じて行いますのでインストールしてください。

Node のインストール

Node のサイト (http://nodejs.jp/nodejs.org_ja/docs/v0.10/) から node をダウンロードしインストールします。メッセージに従いインストールすれば完了です。



Ruby のインストール



windowsの場合、環境変数にパスを登録するかオプションを聞かれますので登録するよう

設定してください。

Git のインストール

git のインストールは github のサイト

<https://help.github.com/articles/set-up-git>

に詳しく記載がありますのでセットアップを行って下さい。まず

<http://git-scm.com/downloads>

から対応する OS の git モジュールをダウンロードしインストールを行います。次に git に自信の名前とメールアドレスを登録します。

```
$ git config --global user.name "Your Name Here"
```

```
$ git config --global user.email "your_email@example.com"
```

github への登録

ローカルの git とgithub を連携させます。下記手順位より公開鍵暗号化方式を使って連携を行います。

```
$ cd ~/.ssh
```

```
$ ssh-keygen -t rsa -C "your\_email@example.com"
```

```
$ pbcopy < ~/.ssh/id_rsa.pub
```

```
$ cat id_rsa.pub
```

cat で開いた公開鍵を github へ登録します。鍵の登録方法は

<https://help.github.com/articles/generating-ssh-keys> の「Step 3: Add your SSH key to GitHub」を参照してください。手順は以下の通りです。

1. Go to your [Account Settings](#)
2. Click "[SSH Keys](#)" in the left sidebar
3. Click "Add SSH key"
4. Paste your key into the "Key" field
5. Click "Add key"
6. Confirm the action by entering your GitHub password

稼動テストとして適当なリポジトリを github に作成し clone, add, commit, push などを使ってみてください。

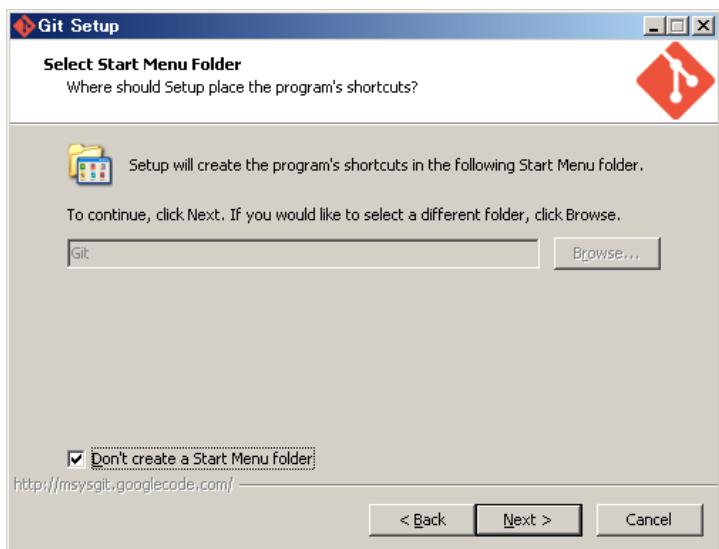
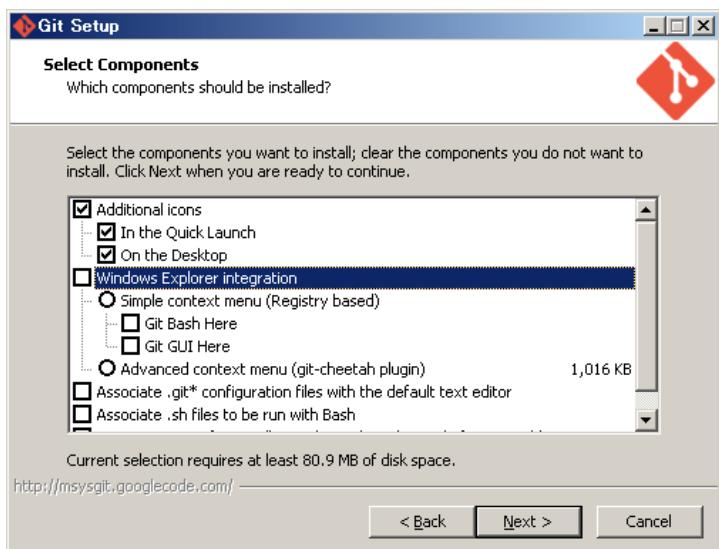
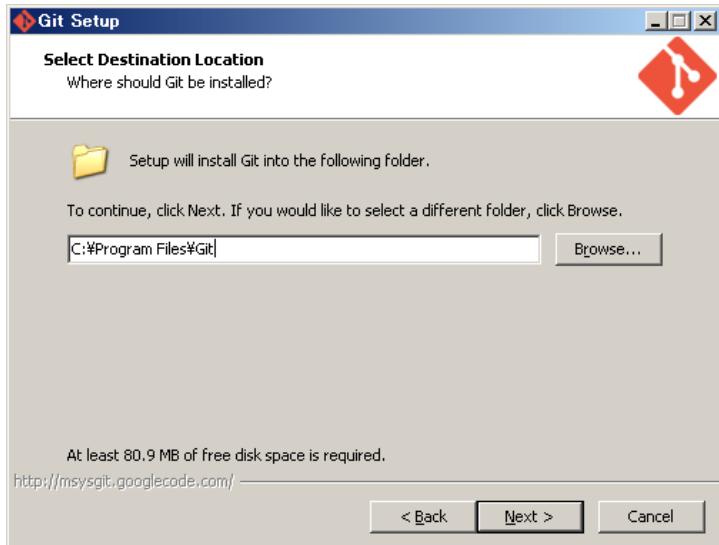
windowsでのインストール

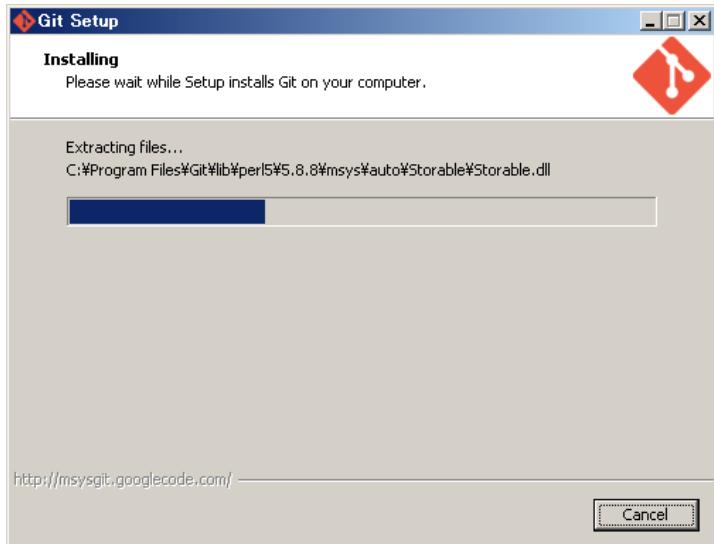
macよりも手順が少し多いです。文字コードの関係もありますのでそれなりの設定が必要です。

msysgitのインストール

msysgitは<http://msysgit.github.io/>からダウンロードします。大きく「Download」と記載してますので直に分かると思います。ダウンロード後インストーラーを実行します。途中設定等を聞いてきますので画面イメージを添付します。









これで msysgit のインストールは終了です。msysgit によりインストールされた git bash の設定を行います。これは ls コマンドで日本語を表示させるための手順です。

```
$ vi ~/.bashrc
alias ls = 'ls --show-control-chars'
$ git config --global core.quotepath off
```

また、windows の改行コードは「CRLF」で Unix 系は「LF」です。同一のファイルに 2 種類の改行コードが混在すると問題になりますので「LF」で統一します。改行コードが「LF」以外の場合はコミット出来ないように下記設定を行います。

```
$ git config --global core.safecrlf true
$ git config --global core.autocrlf input
```

Sass/Compass のインストール

Sass / Compass は下記コマンドラインでインストールを行って下さい。事前準備として git (windows の場合は git bash) をインストールしてください。

```
$ gem update --system
$ gem install sass
$ gem install compass
```

Yeoman のインストール

Yeoman をインストールするには <http://yeoman.io/> を見ても分かる通り簡単で

```
$ npm install -g yo
```

を実行することで Grunt と Bower もインストールされます。

Yeoman の機能は豊富ですので grunt は <http://gruntjs.com> を bower は <http://bower.io/> を参照した方がより良いです。

その他の知識

使用するテンプレートにより異なりますが、モダンWebアプリケーションを開発するにあたり知識として必要となる要素のインク先をまとめます。

1. CoffeeScript(<http://coffeescript.org/>): JavaScript を Ruby のような形式にした言語
2. Sass(<http://sass-lang.com/>): CSS ファイルを作成するために扱うプログラム言語のようなもの (Ruby が必要)
3. Compass(<http://compass-style.org/>): CSS ファイルオーサリングフレームワーク
4. Angular(<http://angularjs.org/>): JavaScript の MVC フレームワーク
5. Backbone(<http://backbonejs.org/>): JavaScript の MVC フレームワーク
6. PhantomJS(<https://code.google.com/p/phantomjs/>): WebKitをヘッドレス化して JavaScript API を利用できるようにしたツール
7. Mocha(<http://visionmedia.github.io/mocha/>): Node.js上もしくはブラウザ上で動く JavaScript テストフレームワーク
8. Jasmine(<http://pivotal.github.io/jasmine/>): Node.js上もしくはブラウザ上で動く JavaScript テストフレームワーク
9. Twitter BootStrap(<http://twitter.github.io/bootstrap/>): Twitter社が提供する CSS フレームワーク
10. RequireJS(<http://requirejs.org/>): JSの非同期ローディングとモジュール化を行うライブラリ

3rd Step: Sass / Compassでのコーディング