# DN KT Materials

## Schedule Outline:

26th/ Dec.

    Flight to DN

    Kickoff Meeting

    Team Building activity

27th/ Dec.

    KT for Mass HTML Production workflow:

Introducing the overall process of font-end project.

https://theblog.adobe.com/what-does-a-ux-designer-actually-do/

Introducing input materials (Photoshop, Wireframe, instruction file)

Testing (text content, layout, behavior across multiple screen resolutions and devices)

    Hands-on: Dev Environment setup (VS Code IDE and its extensions, git, yarn, build tools, ...)

https://medium.com/the-node-js-collection/modern-javascript-explained-for-dinosaurs-f695e9747b70

28th/ Dec.

    Hands-on:  Working with input materials

    Q&A

    Making Exercise (picking from old 「Sompo」project)

29th/ Dec.

    Making Exercise (cont.)

    Review Exercise output

    Q&A

   Flight back to SG

# A: Setup Development Env.

### 1) install apps

#### 1.1) Adobe Apps

- Adobe Photoshop

- Adobe Experience Design

(DN team, please, install and setup these 2 app on Monday (26/12/2017). 7 days trial period will enough for this KT & Hands-on)

#### 1.2) VS Code

- Visual Studio Code ( <https://code.visualstudio.com/download> )

#### 1.3) Node.js(Latest LTS Version: 8.x.x)

<https://nodejs.org/en/download/>

To setup proxy,replace <username> and <password> with your account and run these 2 commands in your terminal.

*npm config set proxy http://<username>:<password>@proxy-yourcompany.com:8080*

*npm config set https-proxy http://<username>:<password>@proxy-yourcompany.com:8080*

*(todo: alternative config file)*

#### 1.4) git

<https://git-scm.com/download/win>

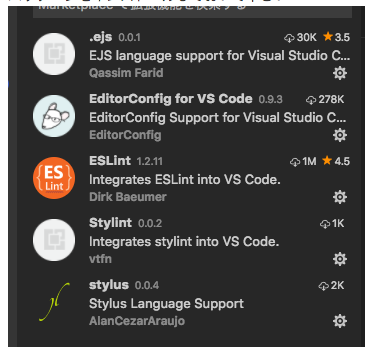
*(todo: proxy config cmd and alternative config file)*

### 2) Visual Studio Code extensions

#### 2.1) open VS Code

#### 2.2)

#### 2.3) search&install plugins below

 - EditorConfig for VS Code  
 - ESLint  
 - .ejs  
 - language-stylus  
 - Stylint

### 3) Photoshop setting

3.1 ) unit setting

* Photoshop’s toolbar 編集（E）-> 環境設定 -> 単位・定規
* 「定規：pixel, 文字: point」に設定

4): Node.js, gulp & yarn setting

- $ npm i -g gulp yarn

- $ gulp -v

- $ yarn --version

# B: Works Inputs Materials

Copy following files to your machine.

### 1) Wireframes

(todo: pick Wireframe files from 4DD shared google drive)

### 2) Components document (Styleguide)

### 3) Base source codes

1. Extract sompo.zip
2. $ cd path/to/sompo/dir  
   $ yarn  
   $ yarn start

4) Instruction files

(todo: pick instruction files from 4DD shared google drive)

# C: Project Files Structure

* EJS(HTML template language)
* Stylus(CSS meta language)
* JS

|  |  |
| --- | --- |
| /README.md |  |
| /.editorconfig | Settings for indent style etc. |
| .eslintrc | Settings for linting JavaScript. |
| /.eslintignore | Settings for linting JavaScript. |
| /.gitignore | Gitignore setting |
| /.sylintignore | Settings for linting Stylus. |
| /.stylintrc | Settings for linting Stylus. |
| /gulpfile.babel.js | Settings for gulp. |
| /package.json | Settings for npm/yarn. When you run `$ yarn`, packages writtern in this file will be installed. |
| /yarn.lock | Settings for yarn |
| /public | Compiled files that are based on /src dir's files. |
| /src | Codes before build. ejs, stylus, images, scripts. DN team should edit only EJS files in /src/pages. |
| /src/assets | images & js |
| /src/atoms | atomレベルのcomponentのejs & stylus |
| /src/config | configs for styleguide etc. |
| /src/molecules | moleculeレベルのcomponentのejs & stylus |
| /src/organisms | organismレベルのcomponentのejs & stylus |
| /src/pages | 実際につくるページのejs. ここでつくったejsは/publicにhtmlとしてコンパイルされる。 |
| /src/styleguide | files for generating styleguides |
| /src/stylus | components以外のstylusのコード。(reset.css, mixin, variables etc.) |
| /src/templates | templatesレベルのcomponentのejs & stylus |
| /src/utility | atmic designの整理に収まらないhelper, utilityクラスのstylus |

D: Exercises

Steps:

* Create file <screen\_name>.ejs in /src/pages/.
* Follow instruction in instruction file.

|  |  |
| --- | --- |
| Screen name | Instruction file |
| xxx | xxx |

(todo: pick some screen and coresponding intruction file from sompo prj)

# E: Checking markup errors

### やり方の概要

1. ブラウザで右クリックして、 「ページのソースを表示」を選ぶ(chromeの場合)
2. <https://validator.w3.org/#validate_by_input> にコードを貼り付ける。”Check”ボタンを押す。
3. warningsとerrorsを確認