Charte de structure du site Bootstrap

http://formation.bourbier.net/sites/formation.bourbier.net/stagiaires/Site\_Bootstrap/v-cegedim/

Convention de nommage (id, class, variables sass) :

**Division de la page Html (zonning)** :

* nav (id)

logo (class)

ul > li

* + header (id)

h1

p

button (class)

* + - sectionSliders (id)

slider (class)

img

* sectionCards (id)

img

h2

fontAwsone

h3

p

* sectionProfiles (id)

sectionProfilesLight (class)

sectionProfilesImg (class)

h3

sectionProfilesImgGuillements (class)

p

sectionProfilesImgGuillements (class)

sectionProfilesStrong (class)

sectionProfilesImg (class)

h3

sectionProfilesImgGuillements (class)

p

sectionProfilesImgGuillements (class)

* footerTop (id)

h3

p

* + footerBottom (id)

footerBottomBorder (class)

p

footerBottomLogoReseaux (class)

**Division de Sass** :

* Fichiers :
  + desktop.scss > desktop.css
  + mobile.scss > mobile.css

* Variables
  + div-Propriété
* Class, Id
  + divPropriété

GITHUB

hacker’s version

Sur son ordi dans son terminal:

$ git init // *Crées un fichier .git dans son projet.*

$ git add . // *Ajouter les modif dans .git*

$ git commit -m “Le titre de ma modif et mon argumentation du code” // *Donne un titre au fichier a envoyer*

Sur le site GitHub

$hub create // *Créer un repo sur git hub*

Pour les collaborateurs:

$ git remote add origin [git@github.com](mailto:git@github.com):MonCompteGit/LeNomDuProject.git // *Pour se connecter au repo distant*

$ git pull origin master / / *Récupère le projet sur son ordi*

Pour créer une branch

on code jamais dans le Master !!!

* **Always** make sure git status is **clean** before pull, checkout or merge.

$ git checkout -b le-nom-de-ma-branch // *Le -b créer une banche visible dans son terminal.*

$ git push origin le-nom-de-ma-branch // Envoi de mon ordi au repo git

$ git checkout master // Change de branch sur master

$ git checkout master // Mettre à jour le master (le repo)

Pour Merge une branch (fusion des branch)

$ git status

# MAKE SURE THIS IS CLEAN  
$ git checkout le-nom-de-ma-branch // Se mettre sur sa branch   
$ git merge master // Fusion du code sur sa branch

### **IN CASE OF CONFLICT**

Sometimes a Pull Request won't be mergeable.

Why? master changed between the time you created the branch and now.

*$ git status # ⚠️ ⚠️ ⚠️ Make sure it's clean before proceeding  
$ git checkout master  
$ git pull origin master # pull the latest changes  
$ git checkout unmergeable-branch # switch back to your branch  
$ git merge master # merge the new changes from master into your branch  
# 😱 Conflicts will appear. It's normal!  
# 👌 Open sublime and solve conflicts (locate them with cmd + shift + f `<<<<<`)  
# When solved, we need to finish the merge  
$ git add . # add the files in conflict  
$ git commit --no-edit # commit using the default commit message  
$ git push origin unmergeable-branch # push our branch again*