

Tips&Tricks  
La trousse de secours du front-end dev

Bruxelles Formation | CEPEGRA | Olivier Céréssia | 2015  
  
webdesigner.cepegra-labs.be

# Préambule

La vie d’un front-end developper est parfois un peu barbante. Il faut souvent, pour initier un projet, recommencer par des bases qui, d’un projet à l’autre, sont communes.

L’écriture du code peut parfois être brouillonne, peu structurée. Votre façon de nommer vos classes, fichiers, dossiers peut avoir un impact direct sur votre productivité ou sur la productivité des personnes avec qui vous allez travailler.

Cerise sur le gâteau : on nous demande, en plus de savoir tout faire, de tout faire vite !

Le but de ces quelques pages est simple : vous donner des clés, des outils pour améliorer votre productivité, votre façon de travailler, vos habitudes de codeur.

Bref, vous offrir des conseils et des « trucs & astuces » qui pourront vous aider au quotidien.

On va se pencher sur :

* vos habitudes de codage,
* les logiciels et extensions qui vous améliorer votre confort de travail,
* les raccourcis clavier qu’il est bon de connaître dans vos environnements de travail (OS et logiciels),
* la structuration de vos fichiers,
* la nomenclature de vos fichiers,
* l’arborescence de vos projets,
* etc.

Tout cela pour faire de vous des super-front-end dev.

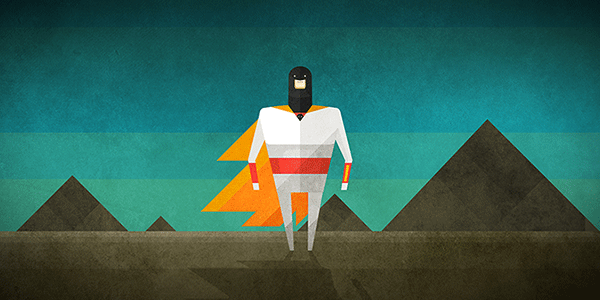


Table des matières

[Préambule 2](#_Toc348513155)

[html5 : la révélation 4](#_Toc348513156)

[ Nouvelle grammaire 4](#_Toc348513157)

[Simplification du !DOCTYPE 4](#_Toc348513158)

[Une syntaxe permissive 4](#_Toc348513159)

[ Nouvelles balises sémantiques 5](#_Toc348513160)

[<header> et <hgroup> 5](#_Toc348513161)

[<footer> 5](#_Toc348513162)

[<nav> 5](#_Toc348513163)

[<aside> 5](#_Toc348513164)

[<section> 5](#_Toc348513165)

[<article> 5](#_Toc348513166)

[<figure> 6](#_Toc348513167)

[ Nouvelles balises multimédia 7](#_Toc348513168)

[<audio> 7](#_Toc348513169)

[<video> 8](#_Toc348513170)

[ Des formulaires améliorés 9](#_Toc348513171)

[De nouveaux types de champs 9](#_Toc348513172)

[Range 10](#_Toc348513173)

[date, datetime, month, week, time 10](#_Toc348513174)

[search 10](#_Toc348513175)

[De nouveaux attributs intelligents 11](#_Toc348513176)

[Placeholder 11](#_Toc348513177)

[Autofocus 11](#_Toc348513178)

[Autocomplete 11](#_Toc348513179)

[Required 12](#_Toc348513180)

[ Des attributs puissants 12](#_Toc348513181)

[Draggable 12](#_Toc348513182)

[Contenteditable 12](#_Toc348513183)

[ Vers un HTML5 « transitionnel » ? 13](#_Toc348513184)

# Organiser son espace de travail

## Arborescence : créer un dossier par projet

Ça semble « bête » à dire, mais il est vital que chaque projet soit isolé dans un dossier « à part » afin que vous puissiez simplement vous y retrouver et les différencier par un nom cohérent et éloquent.

Il est par exemple intéressant d’avoir un dossier général avec le nom du projet puis, à l’intérieur, 2 dossiers qui ont chacun leur utilité :

* « *sources* » pourrait par exemple accueillir tout le matériel nécessaire à la réalisation de votre intégration (psd, pdf du client, images du client, textes, etc.)
* « *spec* » pourrait accueillir toutes les spécifications de votre projet, comme par exemple le cahier des charges, l’architecture, les wireframes, etc.
* « *www* » pourrait être uniquement réservé à l’intégration, c’est là que vous allez vraiment coder votre site.

### Organisez correctement votre dossier « *www* » (intégration)

Un dossier mérite une attention toute particulière : il s’agit du dossier réservé à accueillir le fruit de votre intégration.

Il est important de classifier les différents éléments qui composent votre intégration dans des dossiers différents.

Par exemple, on privilégiera une écriture abrégée pour chaque dossier qui représente le mieux possible ce qui se trouve à l’intérieur du dossier :

* « *img* » pour le dossier qui reprend toutes les images de contenu de votre site,
* « *theme* » pour le dossier qui reprend toutes les images qui servent à la décoration de votre site (surtout pertinent avec les CMS – sinon on peut regrouper les 2 dossiers sous « img »),
* « *js* » pour le dossier qui accueille tous les fichiers de type javascript,
* « *css* » pour le dossier qui accueille tous les fichiers de type Cascading StyleSheet.

### Utiliser les classes et identifiants comme noms d’images

N’hésitez pas à donner à vos images un nom qui se rapporte directement à une zone de votre site. On pourrait par exemple avoir :

###### #footer {background :url(‘../theme/footer.jpg’) repeat-x ;}

De cette façon, il est plus facile d’identifier, lors de l’intégration, à quelle zone correspond quelle image.

## 