



# Création de sites web avancée

## Introduction et installation

### Programmation en agence

Introduction aux méthodes professionnelles

L'objectif de cet atelier est d'appréhender l'usage des technologies et langages utilisés en agences pour concevoir des sites web sur mesure avec Wordpress. Ce sera l'occasion de vous familiariser avec différents langages et concepts de programmation *front-end*.

Tout en fabriquant un portfolio, nous verrons quels outils permettent d'en simplifier la mise en place, la fabrication et l'évolution.

### Mise en situation

Imaginez que vous êtes dans une agence depuis quelques jours et que l'on vous donne un projet à réaliser. Que vous soyez développeur-euse stagiaire ou chevronné-e, la première étape sera de vous familiariser avec les méthodes et outils de travail de votre nouvelle équipe. Dans un premier temps, le but n'est pas de maîtriser la techno mais de s'y adapter, de réussir à l'utiliser pour atteindre votre objectif :

#### *La livraison d'un site web "fonctionnel"*

En jouant le rôle de votre maître de stage, je commencerais par vous faire une introduction à l'écosystème que constitueront vos outils. Nous mettrons ensuite en place notre environnement de développement en installant toutes nos dépendances de travail. Après un tour d'horizon des langages et concepts que vous utiliserez, nous fabriquerons un premier composant pour vous familiariser avec cet environnement. Vous aurez ensuite à charge de produire un portfolio répondant au cahier des charges.

*Chaque équipe a son vocabulaire propre et ses habitudes, il est important d'oser demander des précisions quand un terme ne semble pas clair, un même mot peut vouloir dire plusieurs choses d'une agence à l'autre. C'est le cas ici aussi, si je vais trop vite ou que j'emploie un mot sans en préciser le sens, n'hésitez pas à me poser des questions.*

## Votre mission

Ce que vous aurez à faire durant cet atelier :

- Créer un espace de travail dans l'équipe MMI de Cloud9
- Suivre la procédure d'installation de notre environnement de dev
- Découvrir le *starter theme* **dummy-wordpress** et son écosystème
- Prendre connaissance du *brief* de projet
- Créer les différents composants nécessaires à la production de votre portfolio

## Vos outils

Pour vous aider, appuyez vous sur toutes les ressources à votre disposition :

- La documentation fournie
- La documentation des outils abordées
- La communauté des dev
- Vos collègues
- Votre maître de stage

## Préambule

### Invitation à Cloud9

Une équipe lensmmi à été crée sur Cloud9, pour la rejoindre vous devez y être invité-e. Envoyez-moi une demande par email, vous recevrez alors une invitation. Suivez les quelques étapes de l'inscription, vous pourrez alors créer votre espace de travail.

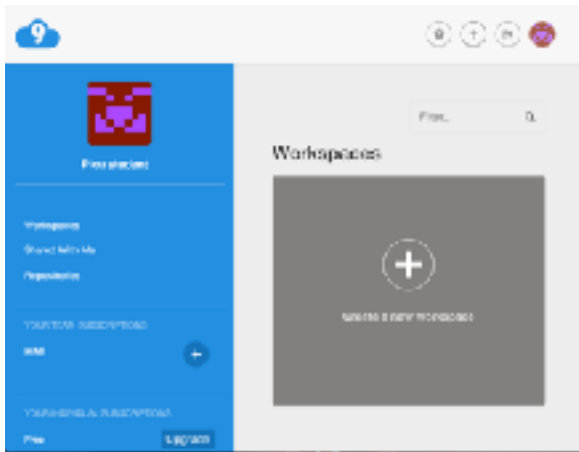
### Web developer tools

Quelque soit votre environnement de travail vos premiers outil sont votre navigateur et son inspecteur. Pensez à toujours avoir l'inspecteur ouvert (*web developer tools*). Pour s'éviter le problème de cache, il existe une option pour le désactiver lorsque l'inspecteur est ouvert.

# Installation

## Utilisation de Cloud9

Cloud9 est un environnement de développement dans le *cloud*. Il va nous permettre d'avoir une machine virtuelle pré-configurée.



Suite à votre inscription à l'équipe MMI, créez le *workspace* qui vous servira tout au long de l'atelier.

## Création d'un espace de travail

Indiquez votre nom pour créer un *workspace* public basé sur une template Wordpress dans la *Team MMI* :

workspace name:

description:

Team:












[Hosted workspace](#) [Clone workspace](#) [Remote SSH workspace](#) [Salesforce](#)

☐ Private  
This is a workspace for your eyes only

☒ Public  
This will create a workspace for everybody to see

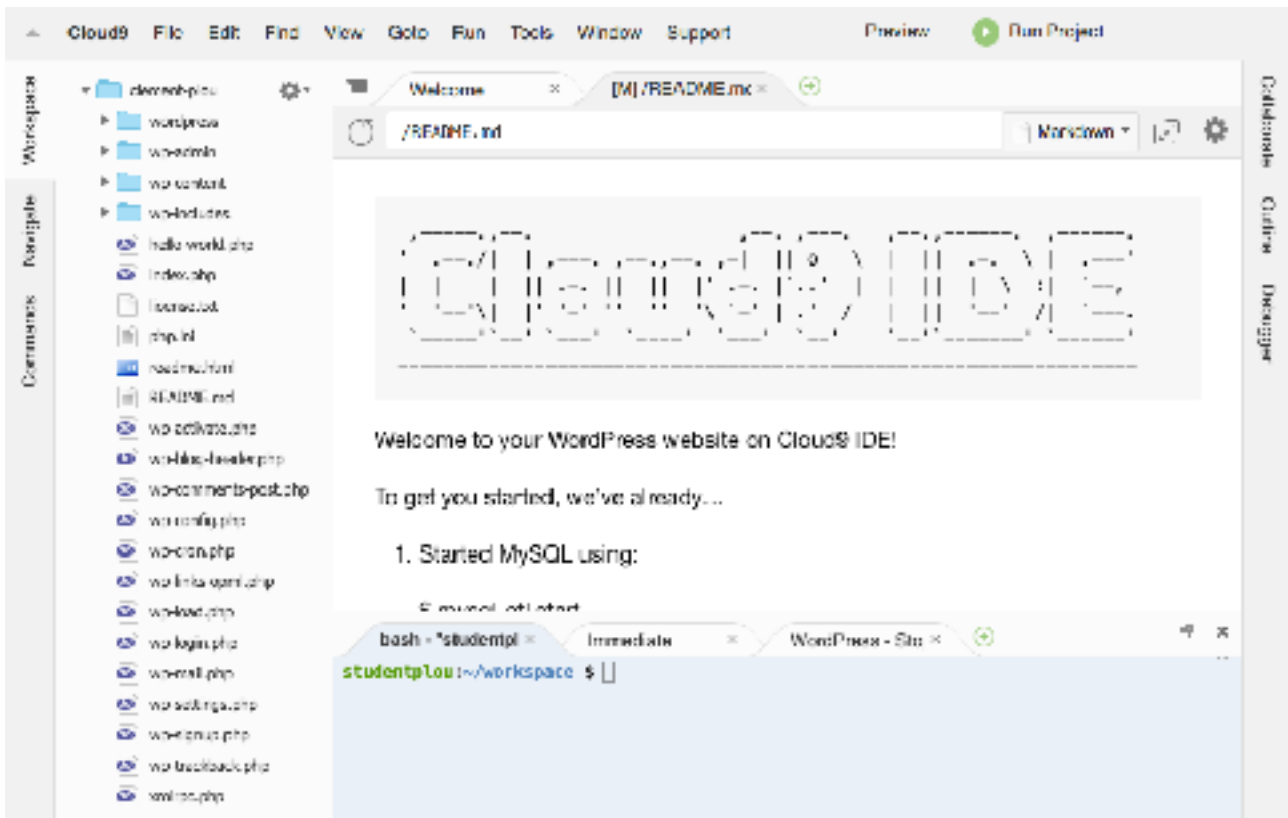
Clone from Git or Mercurial URL (optional)

Choose a template

 HTML5	 Node.js	 PHP, Apache &..	 Python	 Django	 Ruby
 C++	 Wordpress	 Rails Tutorial	 Blank	 Harvard's CS50	

## L'Interface

Vous arrivez ensuite dans l'IDE de Cloud9, il est composé de trois sections et peut être entièrement configurée. À gauche le navigateur de fichier, en bas un accès au terminal et au centre le contenu du fichier ouvert. Le bouton *Run Project* permettra de lancer Apache et de prévisualiser votre site.



Installation de notre environnement

Nous allons maintenant procéder à l'installation de nos outils.

## Wordpress

La première étape est de finaliser l'installation de wordpress. Nous allons donc lancer notre serveur et accéder à l'interface. Au clic sur le bouton *Run Project*, un nouvel onglet de terminal va s'ouvrir et indiquer l'adresse à laquelle votre projet est accessible. Procédez ensuite à la configuration du wordpress, vous aurez alors accès aux *back-end* et *front-end* de votre projet.

Pour la suite, nous allons en profiter pour faire quelques réglages :

- Dans *Réglages / Général* changer l'url du site en passant de *https://* à *http://*
- Dans *Utilisateurs / Votre Profil* désactiver la barre d'outils

## wp-cli

<http://wp-cli.org/#installing>

Le premier outils que nous allons installer se nomme *wp-cli*, c'est un outils en ligne de commande qui va nous permettre d'accélérer certaines tâches. Toutes les commandes qui vont suivre seront à exécuter dans le terminal (premier onglet).

```
$ curl -O https://raw.githubusercontent.com/wp-cli/builds/gh-pages/phar/wp-cli.phar
$ chmod +x wp-cli.phar
$ sudo mv wp-cli.phar /usr/local/bin/wp
```

## Dummy-wordpress

<https://github.com/dummy-team/dummy-wordpress>

Ce *starter theme* sera la fondation de notre projet. C'est un thème pré-configuré, nous verrons en détail comment il fonctionne et nous aidera par la suite.

Utilisons *wp-cli* pour l'installer et l'activer :

```
$ wp theme install https://github.com/dummy-team/dummy-wordpress/archive/master.zip
$ wp theme activate dummy-wordpress
```

Si vous rafraîchissez votre site, il vous indiquera que *Timber* n'est pas activé ou installé, c'est une des dépendances du *dummy-wordpress*. Nous utiliserons *Composer* pour gérer cette dépendance.

## Timber

<https://www.upstatement.com/timber/>

*Composer* est un gestionnaire de dépendance *php*, il va nous permettre de récupérer des bibliothèques de code externes et de gérer ses mises à jour. Votre environnement est livré avec, il suffit de lancer l'installation de nos dépendances, déclarées dans le fichier *composer.json* à la racine de notre thème.

```
$ cd ~/wp-content/themes/dummy-wordpress/  
$ composer install
```

La page d'accueil de votre site devrait maintenant afficher du contenu non stylé. *Dummy-wordpress* est livré avec un *framework* CSS, une version adapté de <https://github.com/dummy-team/dummy>. Les styles et scripts de notre thème utilisent *Sass* et *Babel*, il nous faut donc les compiler. Pour cela nous utiliserons *Gulp*.

## Gulp

<http://gulpjs.com/>

C'est un *task runner*, il va nous permettre d'automatiser certaines tâches. Ici ce sera principalement la compilation des fichiers sources *\*.scss* et *\*.js*. En bonus il intègre *browser-sync* qui nous permettra de rafraîchir notre navigateur à la volée.

*Gulp* est un outil écrit en javascript, il tourne sur le moteur *Nodejs*. Nous utiliserons *npm*, un gestionnaire de paquet *javascript*.

Toujours à la racine de notre thème nous allons installer *Gulp* globalement puis toutes nos dépendances locales (en utilisant *yarn*, plus rapide que *npm*).

```
$ npm install -g gulp-cli yarn  
$ yarn
```

## Browser-sync

Pour permettre à *browser-sync* de fonctionner dans cet environnement deux actions sont nécessaires, modifier sa configuration dans le fichier *gulpfile.js* et ajouter un script à toutes nos pages.

```
# gulpfile.js

browserSync.init({
  port: 8082,
  open: false
})
```

Nous pouvons maintenant exécuter *Gulp* dans notre terminal, il se chargera de compiler automatiquement nos fichiers et de lancer *browser-sync*.

```
$ gulp
```

Une fois lancé, il vous indiquera le code à placer dans toutes vos pages (exemple à ne pas copier/coller). Ce code est à insérer dans le fichier *footer.twig*.

```
[BS] Copy the following snippet into your website,
      just before the closing </body> tag
<script id="__bs_script__">
```