Exercice : Calendar

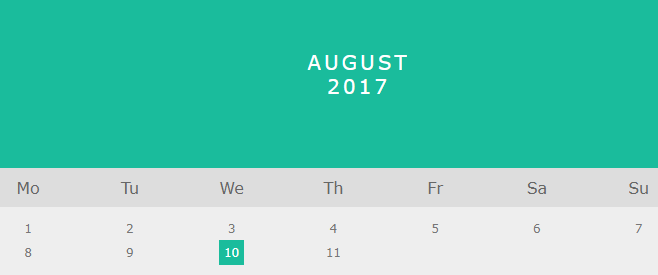
# Le projet

## Introduction

Le projet a pour sujet la génération dynamique d’un calendrier. Etape par étape vous ferez évoluer la solution afin d’obtenir une solution souple et efficace pour générer et afficher votre calendrier.

En guise d’introduction, il vous est demandé de :

* Récupérer l’archive « calendarBegin.zip » sous Moodle
* Intégrer les sources, aussi bien le **html** que le **css** dans PhpStrom



*Légende : résultat à obtenir*

## Analyse

Cette page **html** est totalement statique. Comment pourrions-nous la rendre dynamique à l’aide de **php** ? Imaginez un calendrier qui se mettrait à jour en fonction de la date du système ?

* [Analyse]
  + Etudier le code **html** et son **css.**
  + Identifier les zones qui pourraient être dynamiquement générées par du **php** (variables, boucles, structure conditionnelle, tableau…).
  + Identifier les zones qui ne seront pas dynamiques et donc, resteront de l’affichage statique, peu importe le calendrier à afficher.

Livrable

A la fin de cette étape, chacun a pu identifier les éléments graphiques que nous allons manipuler.

Dans PhpStorm, les codes **html** et **css** ont été placés dans un répertoire « doc ». Nous ne travaillerons sur des copies de ces fichiers.

\*\*\*fin de l’introduction\*\*\*

## Conception



Il est venu le temps de réfléchir en tant que développeur… et quitter un peu le monde de l’analyse.

* [Conception]
  + Dessiner les grandes lignes d’un diagramme de flux représentant la logique à implémenter en **php** pour chacun des zones qui seront générées dynamiquement.

Aide : [Documentation microsoft](https://support.office.com/fr-fr/article/cr%C3%A9er-un-diagramme-de-flux-simple-e207d975-4a51-4bfa-a356-eeec314bd276)

* + Dessiner les grandes lignes d’un diagramme de séquence correspondant à la logique globale qui devra être implémentée (du lancement du script **php** à l’affichage du calendrier en **html**).

Aide : [Documentation microsoft](https://msdn.microsoft.com/fr-fr/library/dd409377.aspx)

* + Le développeur à imaginer un début d’architecture ainsi que les signatures des fonctions en lien avec les diagrammes précédents.

Livrable et critères d’acceptations

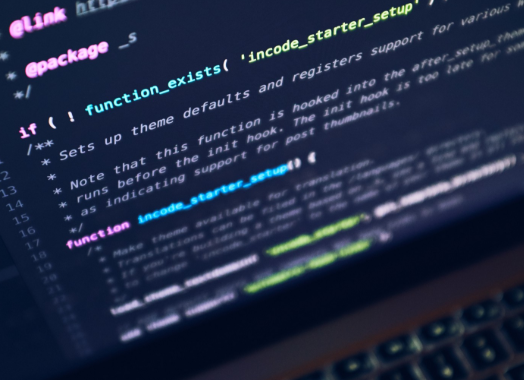
Chaque développeur a dessiné et intégré à son projet un dossier « docs » contenant :

* Un digramme de flux représentant chacune des étapes de création du calendrier.
* Un diagramme de séquence représentant l’interaction entre les fonctions pour créer le calendrier.
* Un premier fichier **php** contenant les régions et les signatures des fonctions à utiliser.

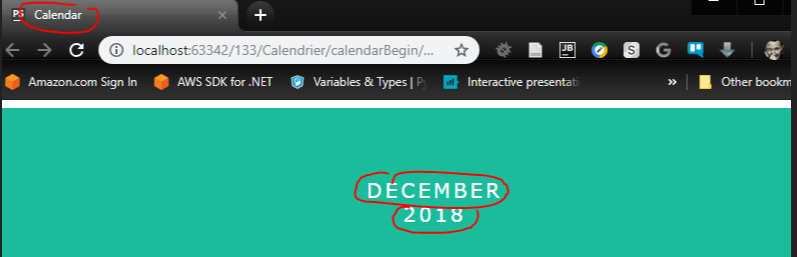
\*\*\*fin de la conception\*\*\*

## Réalisation

**Etape 01**



Commençons le travail de réalisation par les tâches les plus simples…



Aide :

* Récupérer l’archive compressée « calendarStep01.zip » sur Moodle. Intégrez les fichiers dans votre IDE.
* Etudiez-les et identifiez les différentes régions qui ont été configurées.
* Remarquez le positionnement du code html en fin de fichier.
* Un premier mécanisme a été implémenté afin de mettre à jour dynamiquement le titre de la page. Inspirez-vous-en !
* [Réalisation – Step 01]
  + Mettez à jour dynamiquement le nom du mois
  + Mettez à jour dynamiquement l’année

Aide : [Les dates en php](http://php.net/manual/en/function.date.php)

Livrable et critères d’acceptations

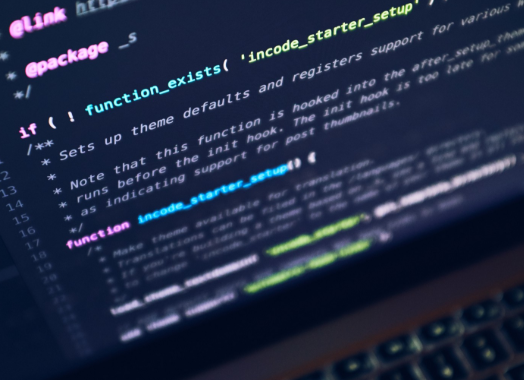
Chaque développeur dispose d’un code respectant les points suivants :

* Le nom du mois courant est affiché dynamiquement.
* L’année du calendrier est affichée dynamiquement.
* Le calendrier n’est pas déstabilisé par l’ajout des nouveaux composants.

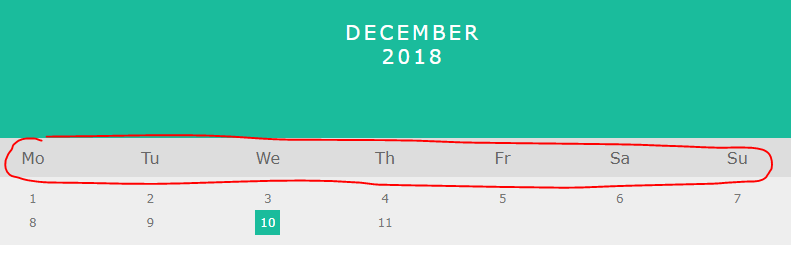
\*\*\*fin de l’étape 01\*\*\*

## Réalisation

**Etape 02**



Pour l’instant nous n’avons mis à jour dynamiquement que des variables simples… attaquons-nous à des formats de variables plus complexes.



Aide :

* Etudier le code html permettant d’afficher les jours de la semaine. Comment pourrions-nous le générer automatiquement ?
* Utiliser un tableau ?
* Boucler sur le tableau et générer du code html ?
* [Réalisation – Step 02]
  + Faites-en sorte que les jours de la semaine proviennent de **php**.
  + Faites-en sorte l’IHM entourée ici en rouge soit calculée au lancement du script **php**.

Livrable et critères d’acceptations

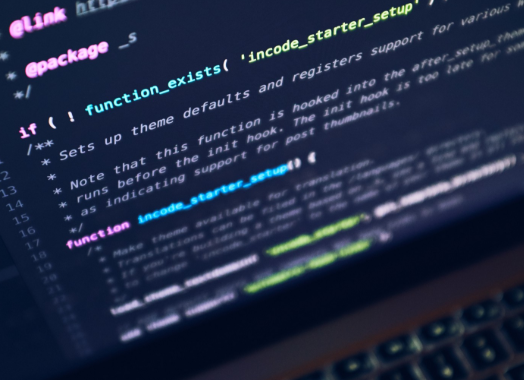
Chaque développeur dispose d’un code respectant les points suivants :

* L’intitulé des colonnes « jour » est généré automatiquement.
* Le calendrier n’est pas déstabilisé par l’ajout des nouveaux composants.

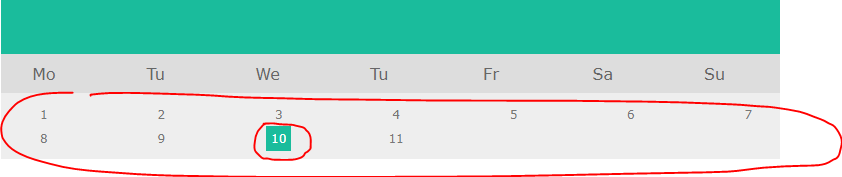
\*\*\*fin de l’étape 02\*\*\*

## Réalisation

**Etape 03**



Continuons avec le calendrier….



Aide :

* Etudier le code html permettant d’afficher la date de chaque jour. Comment pourrions-nous le générer automatiquement ?
* Début et fin du mois ?
* Comment afficher le jour courant d’un autre couleur ?
* [Réalisation – Step 03]
  + Faites-en sorte la date de chaque jour proviennent du **php**.
  + Faites-en sorte l’IHM entourée ici en rouge soit calculée au lancement du script **php**.

Livrable et critères d’acceptations

Chaque développeur dispose d’un code respectant les points suivants :

* La date de chaque jour est produit dynamiquement.
* Le calendrier n’est pas déstabilisé par l’ajout des nouveaux composants.

\*\*\*fin de l’étape 03\*\*\*