Notre Premier Contrôleur

Symfony utilise la structure MVC, on va donc avoir besoin de contrôleurs.

Construction d'un controller

Avec Symfony on crée rarement ce genre de fichier à la main, on va plutôt utiliser le CLI :

```
symfony console make:controller
# Ou l'on peut directement indiquer le nom du controller :
symfony console make:controller NomDuController
```

Commande très facilement à comprendre, on dit à symfony d'utiliser la console pour fabriquer un controller.

Symfony va nous demander le nom que l'on souhaite pour ce controller.
Je vais l'appeler **HomeController**.

Chaque controller doit avoir le mot "Controller" dans son nom, si vous ne le mettez pas ou que vous faites une faute de frappe, Symfony mettra directement le terme "Controller" dans le nom.

Symfony nous indique alors ces deux lignes :

```
created: src/Controller/HomeController.php
created: templates/home/index.html.twig
```

Indiquant qu'il a créé notre controller ainsi qu'un template.
Laissons de côté pour l'instant notre template, on va se pencher vers notre controller.

Structure d'un controller

Voyons lignes par ligne comment est construit notre controller :

- Tout d'abord un namespace.
- Puis quelques use pour importer des classes (pas besoin de require, symfony utilise l'autoloader de composer).
- Puis une classe ayant le même nom que notre fichier et qui hérite d'une classe abstraite qui apporte des méthodes supplémentaires réservées aux controllers.
    - Dans cette classe se trouve une méthode "index". En soit le nom de la méthode importe peu.
    - Au dessus de cette méthode se trouve un attribut "Route".
        - Toute méthode dans un controller qui a cet attribut deviendra une page à part entière.
        - Vous pouvez changer le name mais une fois décidé, ne le changez plus.
        - Le string qui indique le chemin vers cette page peut être changé autant de fois que vous voulez, il sera automatiquement changé là où vous en avez besoin.
    - Actuellement notre page ne fait qu'une seule chose, elle fait appelle à la méthode de rendu qui se trouve dans "AbstractController".

- On lui indique en premier paramètre quel view afficher, (il cherche automatiquement dans le dossier "view").
- En second paramètre on lui donne un tableau associatif qui contient toute les variables qui seront utilisable par la vue. (nos autres variables php ne seront pas accessible dans la vue)

Nous reviendrons sur les possibilités de nos controllers plus tard, maintenant nous allons nous pencher sur notre vue :

Notons que les routes peuvent si on préfère, ne pas être mit au dessus de la fonction controller mais dans le fichier `config/routes.yaml`.

Avant de partir passons à notre vue un tableau et un chiffre aléatoire :

```
[
  'controller_name' => 'HomeController',
  "fruits"=>["banane", "tomate", "cerise"],
  "pays"=>["france"=>"Bonjour le monde !", "angleterre"=>"Hello World !"],
  "chiffre"=>rand(0,10),
  "vide"=>[],
  "xss"=>"<script>alert('Coucou');</script>"
]
```

Et changeons la route par "/" et son nom par "app_accueil".