TP4 – Les gabarits et les fichiers statiques

Introduction, Contexte et Objectifs

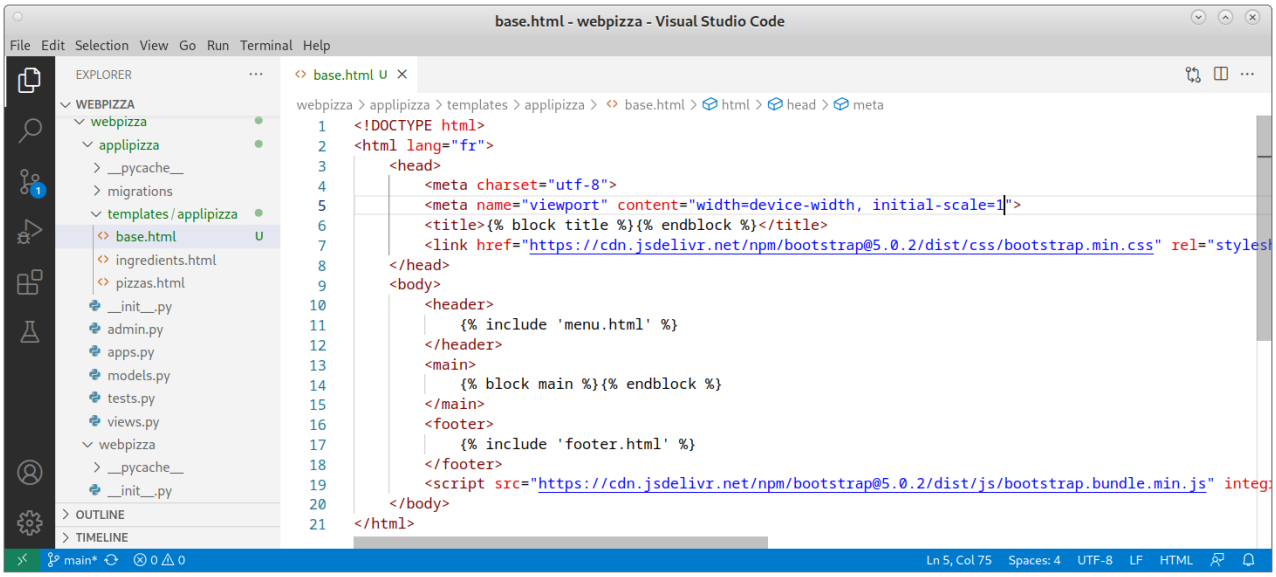
Les premières vues (templates) montrent de la répétition de code html. Par exemple, l'entête html est dupliquée. Ou encore, le menu issu de code bootstrap : tout changement doit être reproduit dans les deux templates pizzas.html et ingredients.html, ce qui n'est pas pratique, et source d'oublis.

L'idée est de fournir un « gabarit » pour que chaque vue produite adopte ce gabarit, en le personnalisant.

Nous allons aussi ajouter quelques fichiers « statiques » comme des fichiers css ou des images.

La vue de base

1. Créez dans le répertoire templates/applipizza un fichier base.html dont le code est le suivant :



Dans ce fichier base.html, il y a quatre parties spéciales : trois *include* et deux *block*

```
{% block title%}{% endblock%}
{% include 'menu.html' %}
{% block main%}{% endblock%}
{% include 'footer.html' %}
```

Les deux **include** vont inclure des parties de code servant à terme à tous les templates, et qu'on va donc factoriser en 2 fichiers pour les inclure à chaque fois.

Ces **include** vont être stockés dans des fichiers html, localisés dans le répertoire des templates.

Les deux **block** seront juste des blocs de code précisés dans les différents templates (pizzas.html, ingredients.html).

Ces **block** ne feront donc pas l'objet de fichiers à part. Ils seront intégrés aux templates.

2. Créez les deux fichiers suivants pour les include :

   a. menu.html

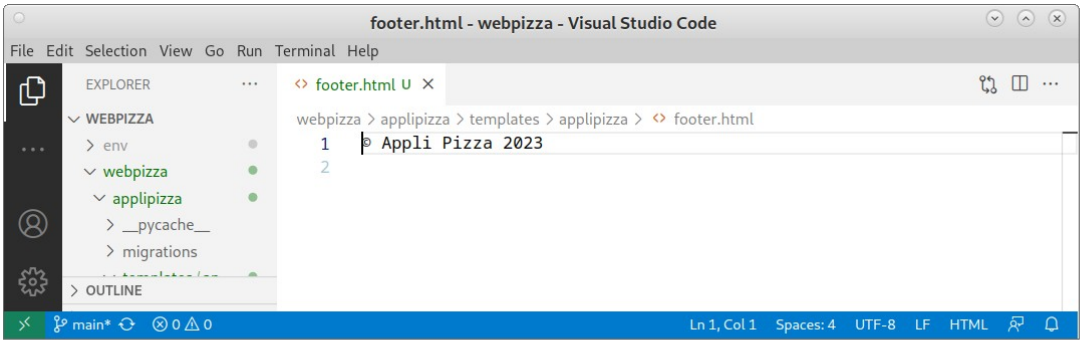
      Ce fichier contient le code html du menu bootstrap. On le perfectionne encore un peu. Voir la page :

      https://getbootstrap.com/docs/5.3/components/navbar/

```
<nav class="navbar navbar-expand-lg navbar-dark bg-primary">
    <div class="container-fluid">
        <a class="navbar-brand" href="#">
            Appli Pizza 2023
        </a>
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarText"
            aria-controls="navbarText" aria-expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarText">
            <ul class="navbar-nav me-auto mb-2 mb-lg-0">
                <li class="nav-item"><a class="nav-link" href="/pizzas/">les pizzas</a></li>
                <li class="nav-item"><a class="nav-link" href="/ingredients/">les ingrédients</a></li>
            </ul>
        </div>
    </div>
</nav>
```

b. footer.html

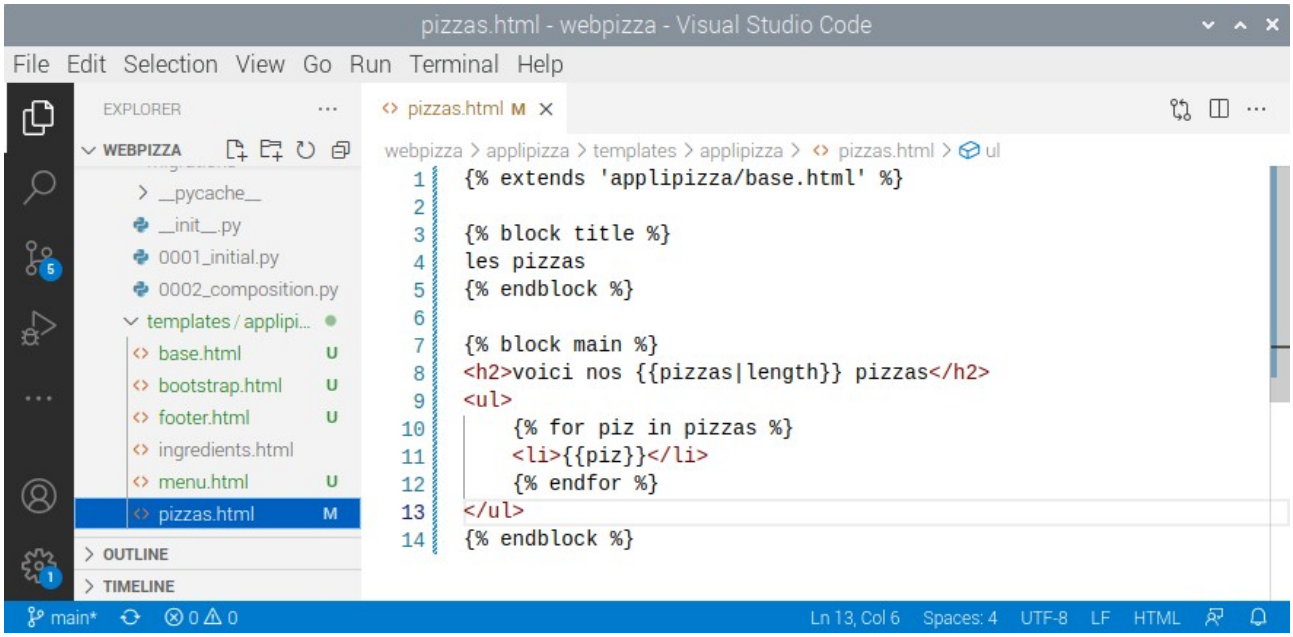
Ce fichier contient le code html du footer



3. Vous pouvez donner les noms que vous voulez aux blocs suivants, les noms *title*, *footer* ne sont pas imposés.

Ces blocs correspondent à des zones que nous allons pouvoir personnaliser.

Modifiez la page pizzas.html pour qu'elle se présente ainsi



- la première ligne est une balise de gabarit qui indique que pizzas.html hérite de base.html

- la deuxième partie est une autre balise de gabarit qui indique quel doit être le contenu du bloc « title »

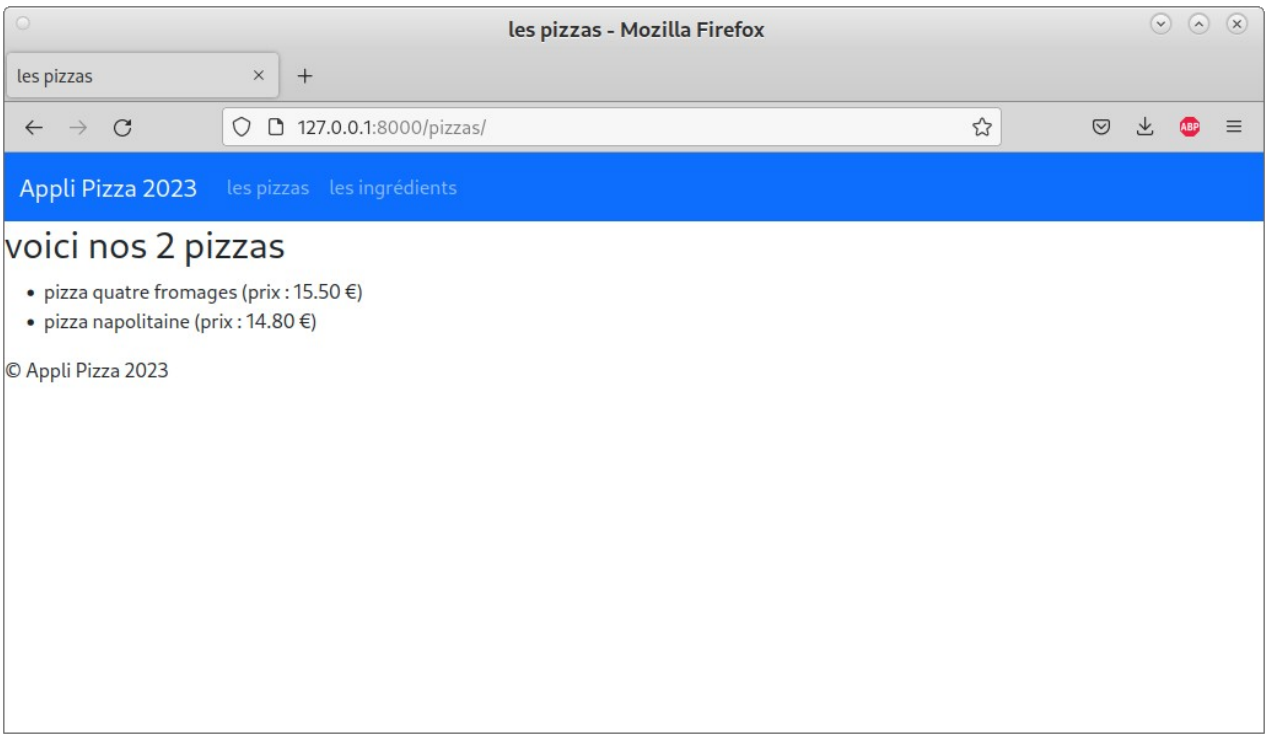
- la troisième indique le contenu du bloc « main »

Sauvegardez, lancez le serveur et vous constaterez sauf erreur que le visuel n'a pas changé.

4. Reproduisez le même changement pour ingredients.html

En résumé :

- pizzas.html et ingredients.html héritent de base.html,
- base.html se charge d'inclure les liens bootstrap, le menu et le footer qui sont dans des fichiers facilement identifiables et modifiables,
- pour compléter base.html, pizzas.html et ingredients.html proposent des block personnalisés, décrits dans chaque template.



Un peu de css personnalisé

Ajoutons un fichier css personnalisé à notre projet : ce sera un fichier « statique », dont le chemin d'accès est spécifique.

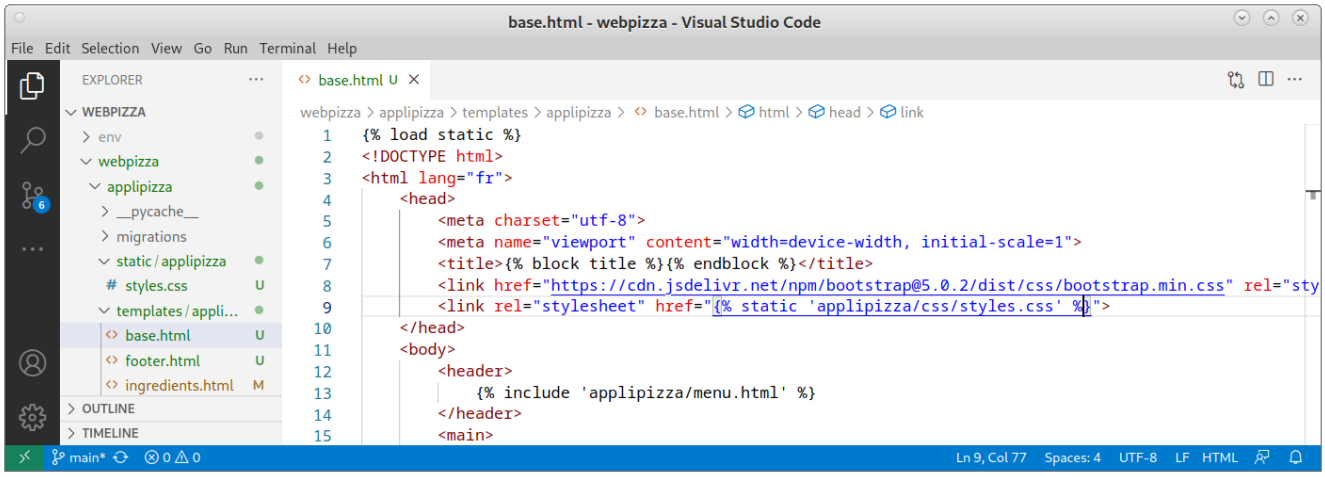
1. Dans la liste INSTALLED_APPS du fichier settings.py, vérifiez que la ligne 'django.contrib.staticfiles' est présente.

2. Vérifiez que la ligne STATIC_URL = 'static/' y figure aussi.

3. Créez un répertoire applipizza/static/applipizza/css et créez un fichier styles.css dans ce répertoire. Ecrivez quelques lignes de css qui vous sauteront aux yeux pour vérifier que votre fichier est bien pris en compte. Par exemple, donnez aux titres h2 un background-color red.

4. Modifiez votre fichier base.html pour incorporer, en première ligne, le code {% load static %}

5. Ajoutez la balise link qui permet d'aller récupérer le css, avec un href comme ci-dessous



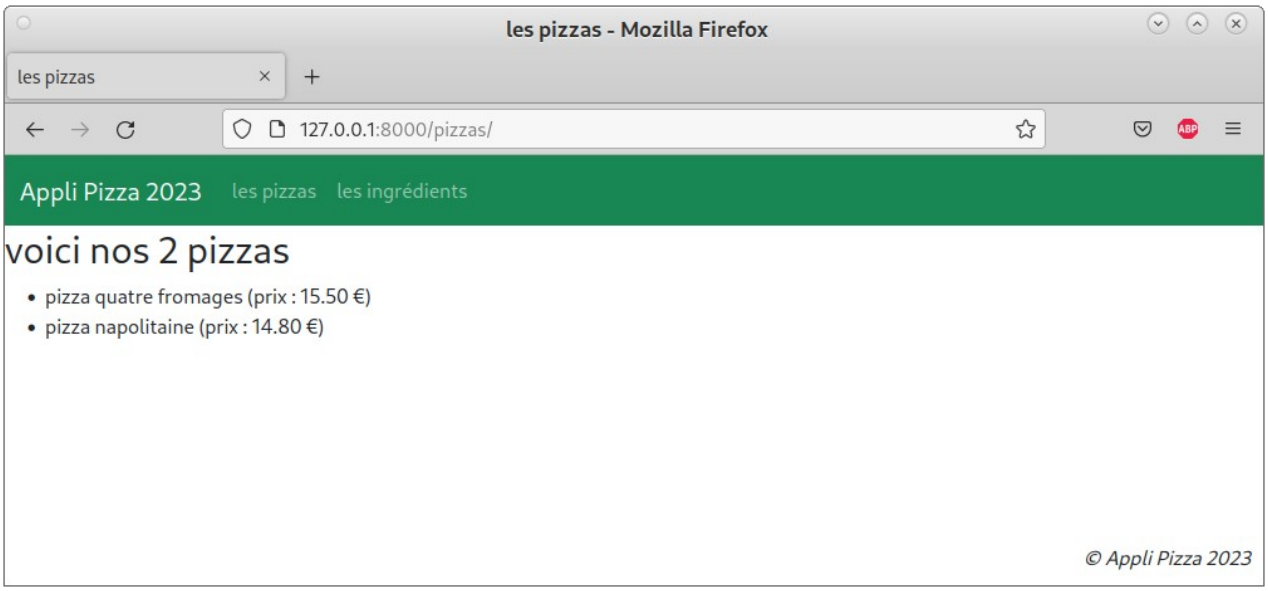
6. Vérifiez que tout fonctionne et que les titres sont bien affichés en rouge, pour les pizzas et les ingrédients. Comme ce n'est pas ce qui se fait de mieux en termes de css, on ne va pas garder ces titres rouges, mais vous allez faire en sorte que le footer soit un « sticky footer ».

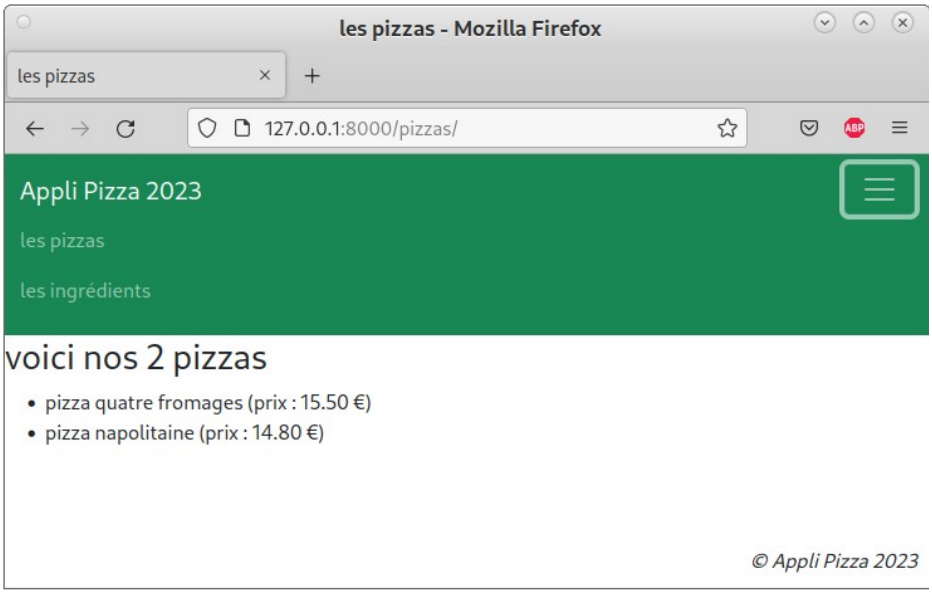
   Autrement dit, le footer colle en bas de la fenêtre. C'est particulièrement visible quand le contenu de la page est si petit que le footer a tendance à apparaître en plein milieu.

Pour cela, vous pouvez utiliser le display : flex, ou d'autres techniques :

https://codeconvey.com/flexbox-sticky-footer/

Arrangez-vous aussi pour savoir changer la couleur de fond de la navbar.

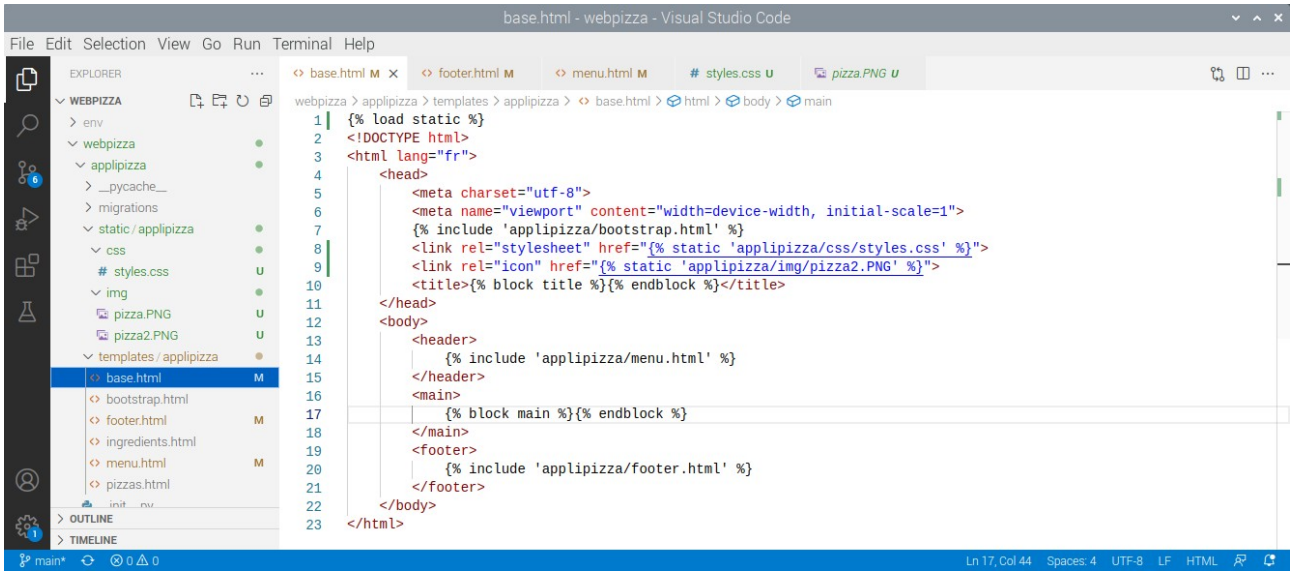




Une icône pour l'application applipizza

1. Sélectionnez une image que vous allez enregistrer au format png (vous pouvez prendre une taille 256 x 256 pixels, à travailler sous Gimp par exemple). Enregistrez-la dans un répertoire static/applipizza/img (à créer, en parallèle de css).

2. Ajoutez, dans base.html, une balise dans le head :



Vérifiez que l'icône apparaît dans l'onglet du navigateur.

3. Faites en sorte que dans la navbar bootstrap, une image apparaisse aussi. Attention, vous aurez à ajouter, dans le template menu.html, la ligne qui permet d'utiliser les fichiers static.