

# TP4 – Les gabarits django et les fichiers statiques

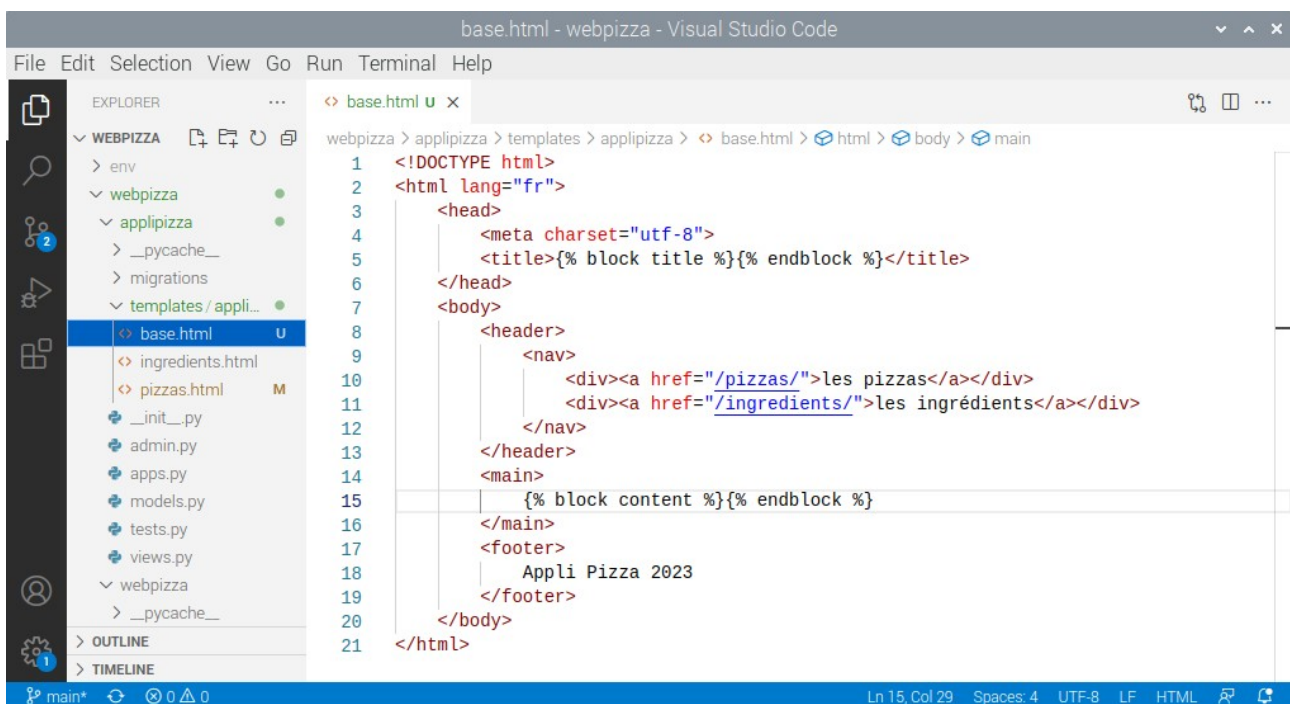
## Contexte et Objectifs

Les premières vues (templates) montrent de la répétition de code html. Par exemple, l'entête html est dupliquée. 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 templates/applipizza un fichier base.html dont le code est le suivant



```
base.html - webpizza - Visual Studio Code
File Edit Selection View Go Run Terminal Help

EXPLORER
webpizza > env
webpizza > webpizza
  > applipizza
    > __pycache__
    > migrations
    > templates / appli...
      > base.html
      > ingredients.html
      > pizzas.html
  > __init__.py
  > admin.py
  > apps.py
  > models.py
  > tests.py
  > views.py
  > webpizza
    > __pycache__
  > OUTLINE
  > TIMELINE

base.html
1 <!DOCTYPE html>
2 <html lang="fr">
3   <head>
4     <meta charset="utf-8">
5     <title>{% block title %}{% endblock %}</title>
6   </head>
7   <body>
8     <header>
9       <nav>
10        <div><a href="/pizzas/">les pizzas</a></div>
11        <div><a href="/ingredients/">les ingrédients</a></div>
12      </nav>
13    </header>
14    <main>
15      {% block content %}{% endblock %}
16    </main>
17    <footer>
18      Appli Pizza 2023
19    </footer>
20  </body>
21 </html>
```

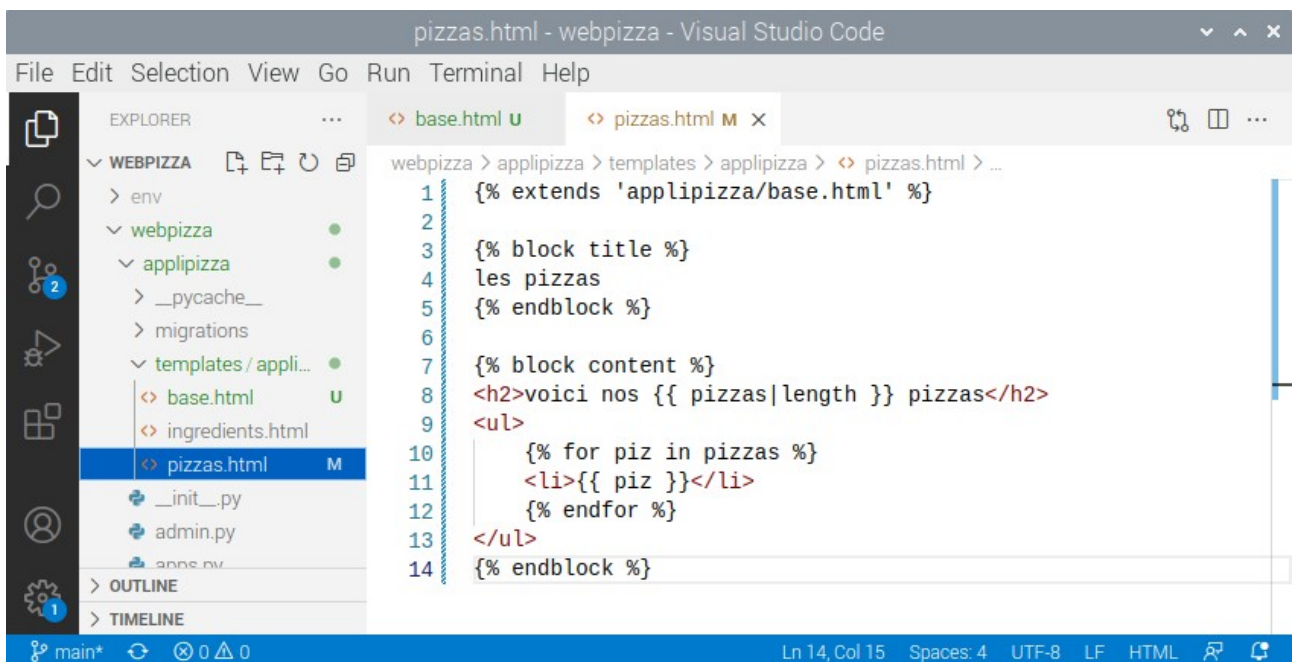
Dans ce fichier base.html, il y a deux parties de même type :

```
{% block title%}{% endblock%}
{% block content%}{% endblock%}
```

Vous pouvez donner les noms que vous voulez à ces blocs, les noms *title* et *content* ne sont pas imposés.

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

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



```
pizzas.html - webpizza - Visual Studio Code
File Edit Selection View Go Run Terminal Help

EXPLORER
WEBPIZZA
  env
  webpizza
    applipizza
      __pycache__
      migrations
      templates/appli...
        base.html
        ingredients.html
        pizzas.html
      __init__.py
      admin.py
      apps.py
  OUTLINE
  TIMELINE

webpizza > applipizza > templates > applipizza > pizzas.html > ...
1  {% extends 'applipizza/base.html' %}
2
3  {% block title %}
4  les pizzas
5  {% endblock %}
6
7  {% block content %}
8  <h2>voici nos {{ pizzas|length }} pizzas</h2>
9  <ul>
10     {% for piz in pizzas %}
11     <li>{{ piz }}</li>
12     {% endfor %}
13 </ul>
14 {% endblock %}
```

- la première ligne est une balise de gabarit qui indique que voitures.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 « content »

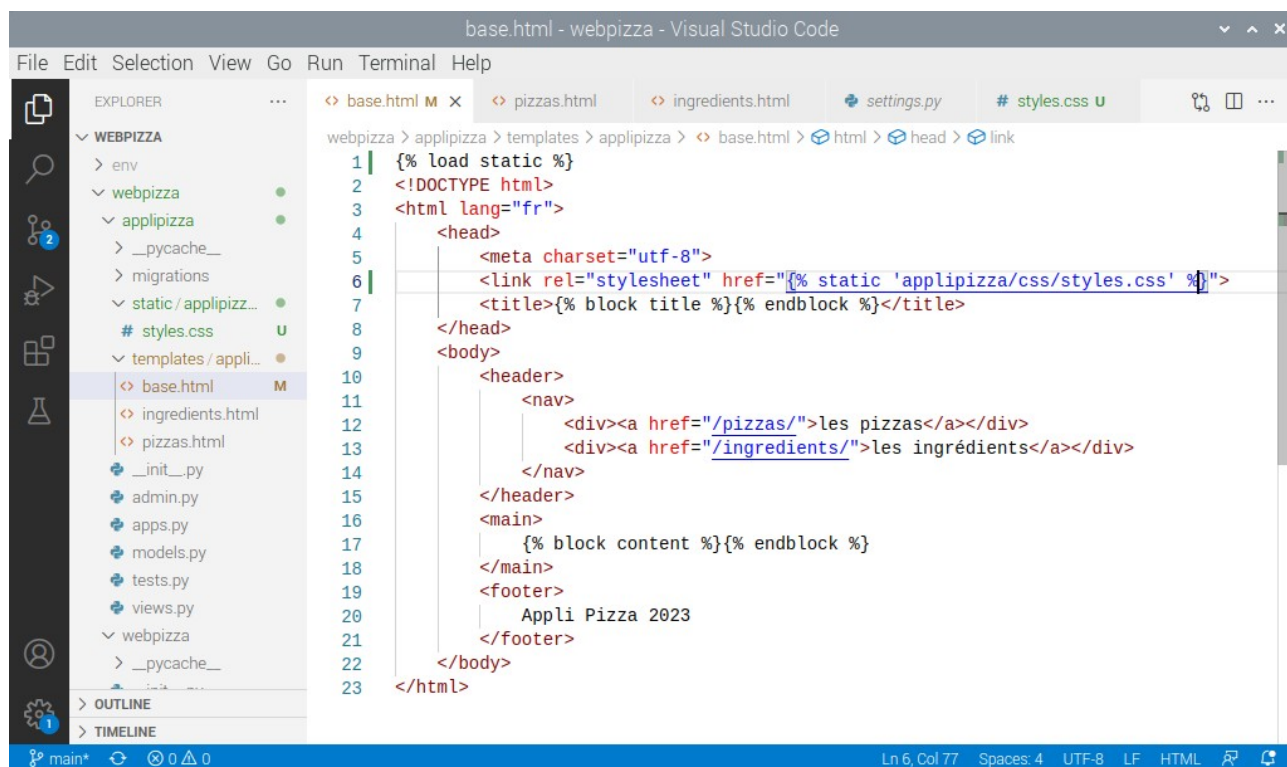
Sauvegardez, lancez le serveur et vous constaterez sauf erreur que le visuel n'a pas changé.

3. Reproduisez le même changement pour ingredients.html

## Un peu de css

Nous allons ajouter quelques fichiers statiques à notre projet : un fichier css, une icône de pizza

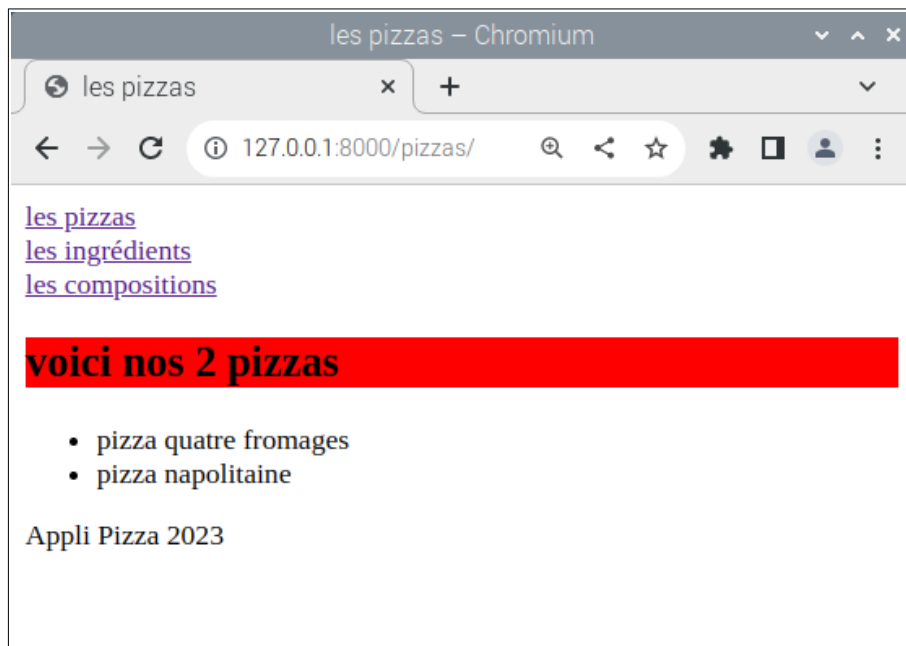
1. Dans la liste `INSTALLED_APPS` du fichier `settings.py`, vérifiez que la ligne `'django.contrib.staticfiles'` est présente.
2. Vérifiez que la ligne `STATIC_URL = 'static/'` y figure aussi.
3. Créez un répertoire `applipizza/static/applipizza/css` et créez un fichier `styles.css` dans ce répertoire. 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



```
base.html - webpizza - Visual Studio Code
File Edit Selection View Go Run Terminal Help

webpizza > applipizza > templates > applipizza > base.html > html > head > link
1 | {% load static %}
2 | <!DOCTYPE html>
3 | <html lang="fr">
4 |   <head>
5 |     <meta charset="utf-8">
6 |     <link rel="stylesheet" href="{% static 'applipizza/css/styles.css' %}">
7 |     <title>{% block title %}{% endblock %}</title>
8 |   </head>
9 |   <body>
10 |     <header>
11 |       <nav>
12 |         <div><a href="/pizzas/">les pizzas</div>
13 |         <div><a href="/ingredients/">les ingrédients</div>
14 |       </nav>
15 |     </header>
16 |     <main>
17 |       {% block content %}{% endblock %}
18 |     </main>
19 |     <footer>
20 |       Appli Pizza 2023
21 |     </footer>
22 |   </body>
23 | </html>
```

6. Vérifiez que tout fonctionne pour `pizzas.html`. Pourquoi les autres vues ne prennent-elles pas le css ? Adaptez les fichiers `ingredients.html` et `compositions.html` pour qu'elles adoptent aussi le gabarit `base.html`, et qu'ainsi le css soit utilisé.



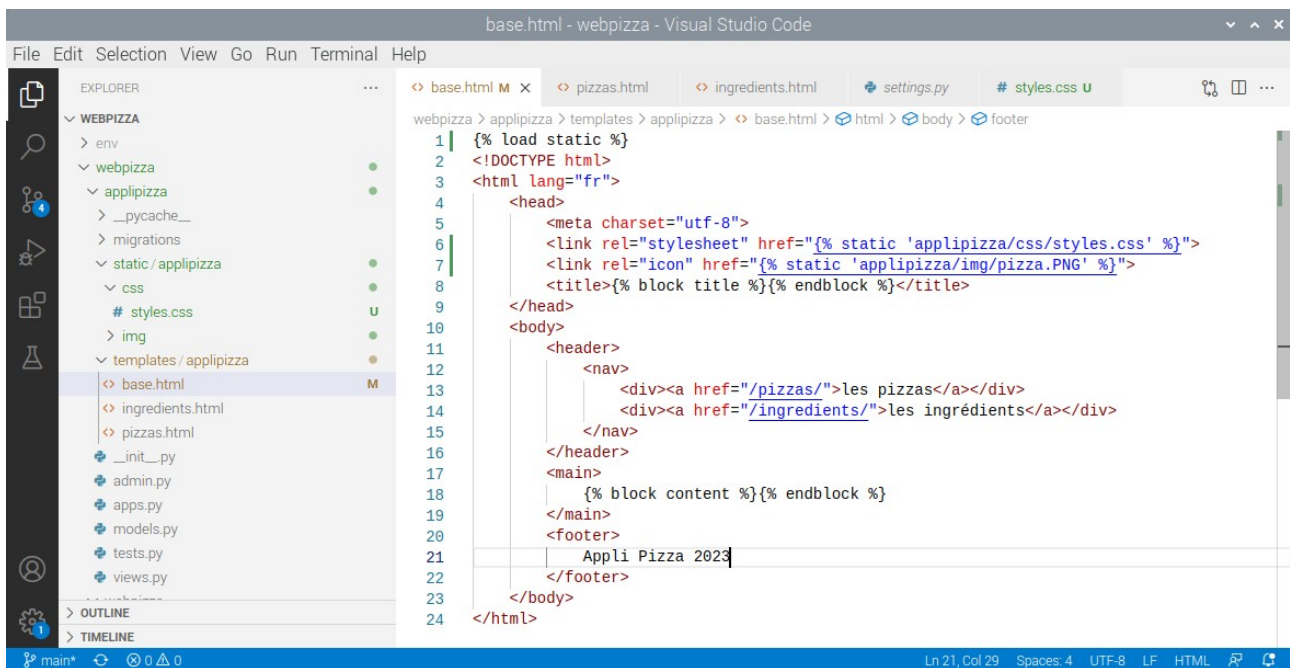
7. Stylisez un peu votre page (par exemple le menu)



## Une icône pour l'application applipizza

1. Sélectionnez une image que vous allez enregistrer au format png dans le répertoire static (vous pouvez prendre une taille 128 x 128 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 :



```
1 | {% load static %}
2 | <!DOCTYPE html>
3 | <html lang="fr">
4 |   <head>
5 |     <meta charset="utf-8">
6 |     <link rel="stylesheet" href="{% static 'applipizza/css/styles.css' %}">
7 |     <link rel="icon" href="{% static 'applipizza/img/pizza.PNG' %}">
8 |     <title>{% block title %}{% endblock %}</title>
9 |   </head>
10 |   <body>
11 |     <header>
12 |       <nav>
13 |         <div><a href="/pizzas/">les pizzas</a></div>
14 |         <div><a href="/ingredients/">les ingrédients</a></div>
15 |       </nav>
16 |     </header>
17 |     <main>
18 |       {% block content %}{% endblock %}
19 |     </main>
20 |     <footer>
21 |       Appli Pizza 2023
22 |     </footer>
23 |   </body>
24 | </html>
```

3. Vérifiez que l'icône apparaît bien dans l'onglet

