Séance 04 :

→ DJANGO – Routage et gestion des requêtes – *Contrôleur*

INF37407 – Technologie de l'inforoute

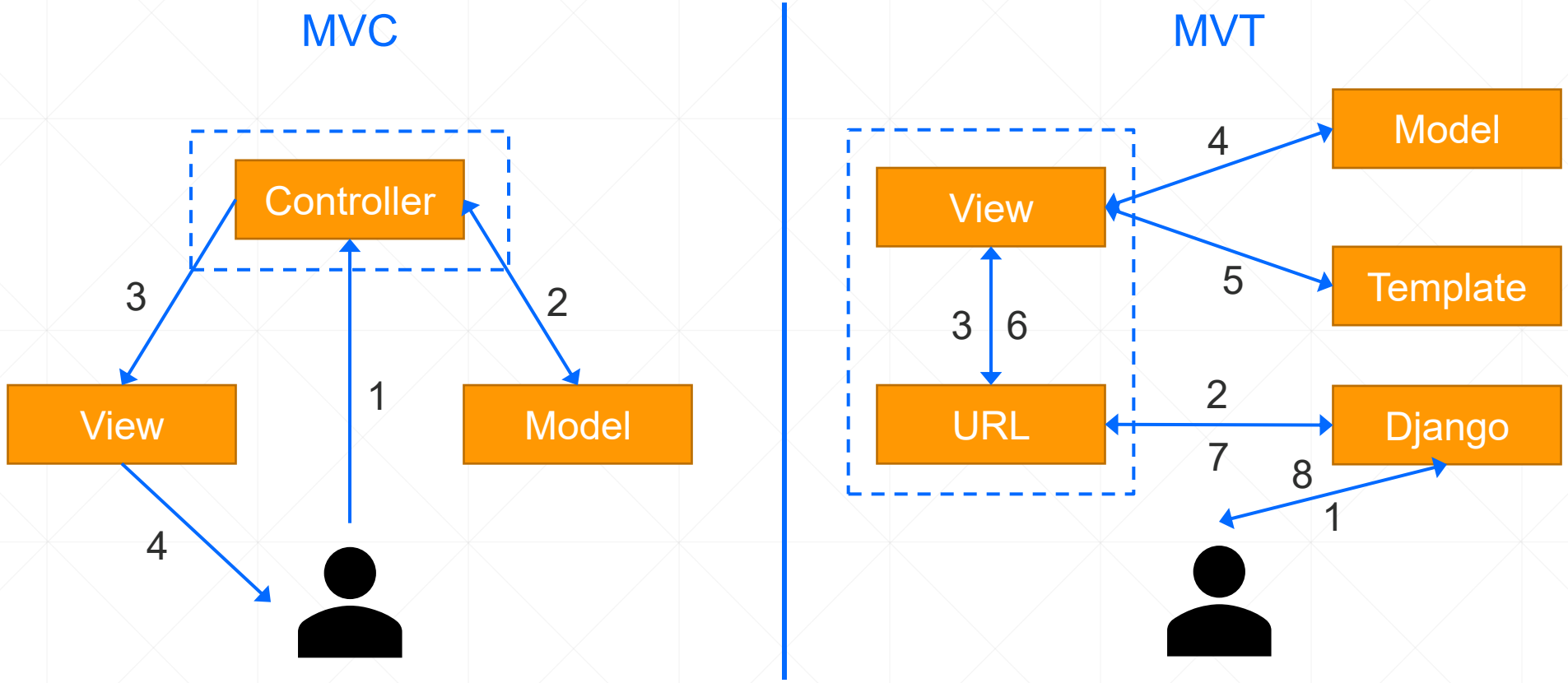
Prof. **Yacine YADDADEN**, Ph. D.

UQAR

Plan

1. Rappel – Architecture MVC & MVT
2. Fonctionnement des *vues* ou *views*
3. Le fichier **`urls.py`**
4. Le fichier **`views.py`**
5. Architecture Django
6. Interface d'administration
7. Mise en pratique :
   a. Création d'un projet avec Django
   b. Création d'une première application
   c. Nos premières pages
8. Questions & Discussion

Rappel – Architecture MVC & MVT



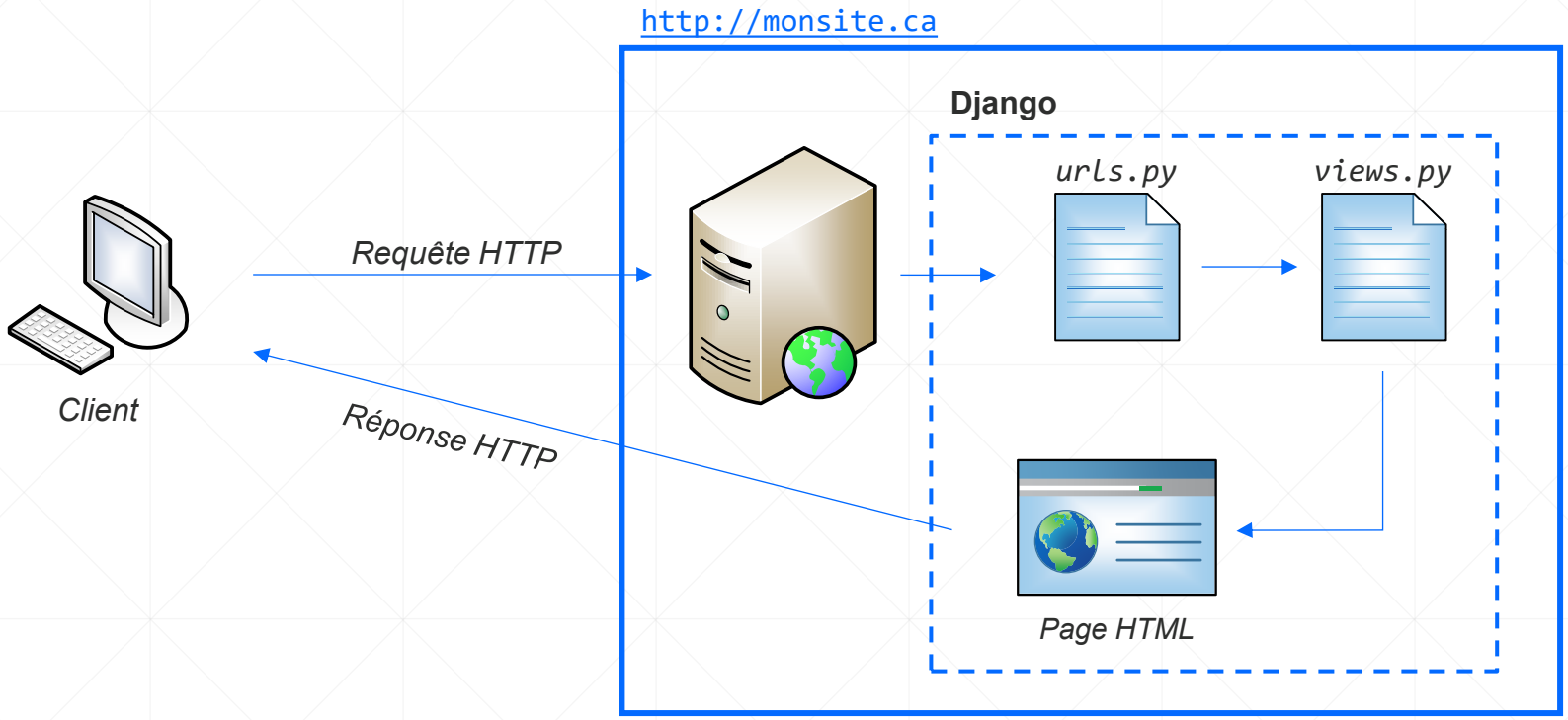
UQAR

Fonctionnement des *vues* ou *views*

Lorsqu'un client demande une page dans un site, il y aura les étapes suivantes :

1. Une requête HTTP est envoyé à l'URL demandé, exemple : `http://monsite.ca` :
   - Contient le nom de la page, exemple : `contact`,
   - Des éventuels paramètres passés à la page,
   - D'autres informations liées au navigateur (version, langue, etc.).
2. Le serveur Web transmettra la requête à **Django**,
3. Ensuite, **Django** cherchera une correspondance dans le fichier **`urls.py`**,
4. Si elle existe, il ira chercher au niveau du fichier **`views.py`** le traitement à effectuer,
5. Une fois effectué, une réponse HTTP est renvoyée au client sous forme HTML.

Schéma global de fonctionnement



UQAR

Le fichier `urls.py`

- **Définition :** « *C'est un fichier Python qui liste l'ensemble des correspondances entre les URLs et les fonctions Python.* »,
- Il y a deux types de ce genre de fichier :
  1. Le premier est général et se trouve dans `project_name/urls.py`, il sert à définir la racine des différentes applications ainsi que de l'interface d'administration,
  2. Le second se trouve dans `project_name/app_name/urls.py`, il est associé à une application et permet de définir les différentes pages.
- L'écriture des différents URL peut utiliser les *expressions régulières*,
- La structure est définie comme suit : **`path(`***`"URL"`***`, `***`fonction_à_appeler`***`)`**

UQAR

Exemple d'*expressions régulières*

Il faut utiliser **`re_path()`** à la place de `path()`.

Expression régulière	Explication
`^`	*En début de l'URL*
`$`	*En fin d'URL*
`\`	*Valeur interprétée (échappement)*
`+`	*Une ou plusieurs occurrences*
`?`	*Aucune ou une seule occurrence*
`{n}`	***n** occurrences*
`{n,m}`	*Entre **n** et **m** occurrences*

Exemple d'*expressions régulières*

Il faut utiliser **`re_path()`** à la place de `path()`.

Expression régulière	Explication
`[]`	*Groupe de caractères*
`.`	*N'importe quel caractère*
`\d+`	*Un ou plusieurs chiffres*
`\D+`	*Un ou plusieurs caractères (pas de chiffres)*
`[a-zA-Z0-9_]`	*Un ou plusieurs mots*
`\w+`	*Un ou plusieurs mots*

Le fichier `views.py`

- **Définition :** « *C'est un fichier Python qui contient les traitements à effectuer via l'exécution de fonctions pour chacune des correspondance URL.* »,
- Chacune des fonctions définie contient un paramètre (objet) **`request`** :
  - Il contient les informations de la *requête HTTP* envoyée par le client,
  - On peut notamment y trouver les informations des formulaires.
- Dans le cas où des paramètres d'URL sont définis dans le fichier **`urls.py`**, il faut également les déclarer et les définir comme paramètres des fonctions correspondantes dans le fichier **`views.py`**.

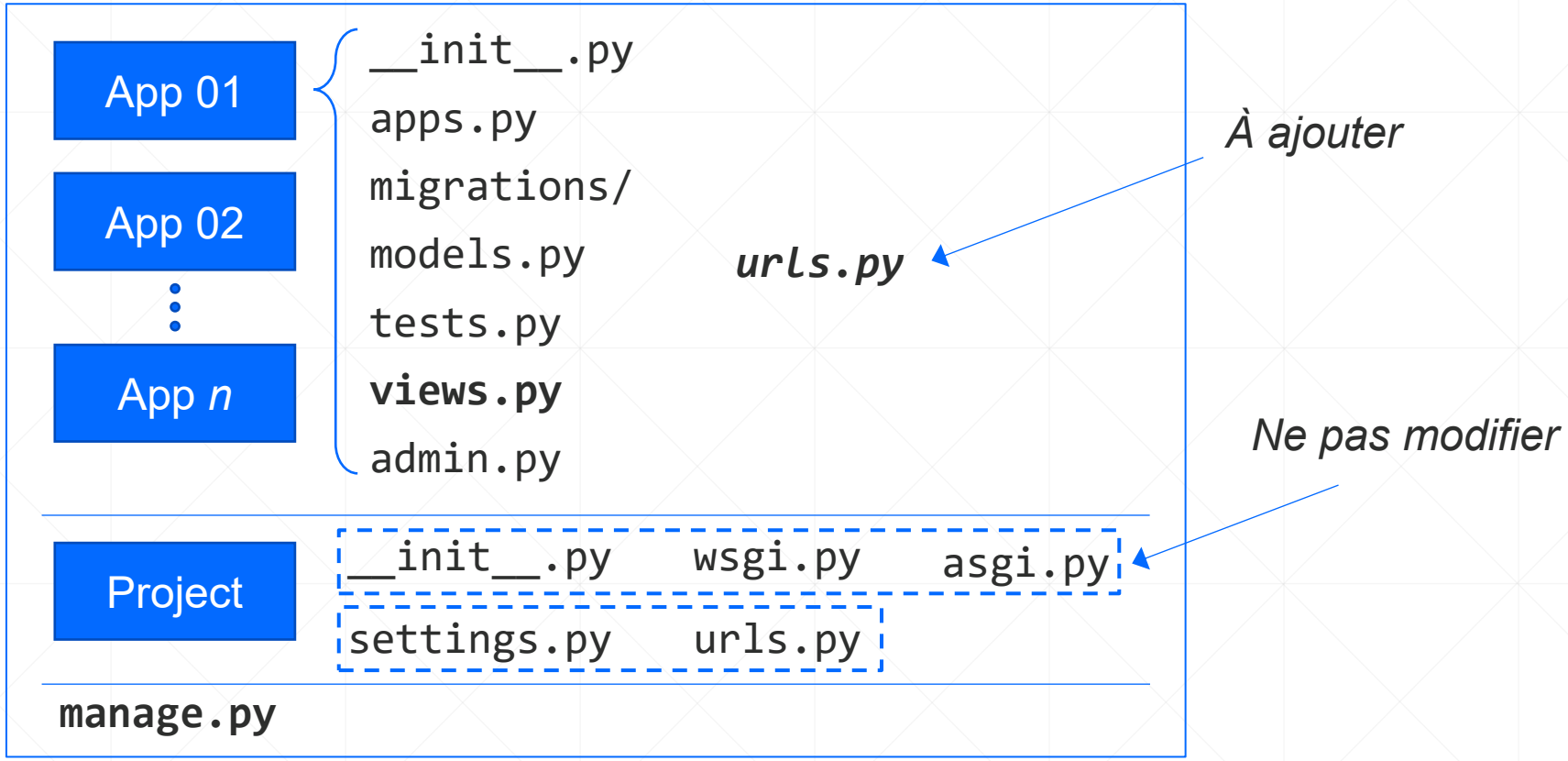
UQAR

Utilisation des paramètres d'URL

Voici les types qu'il est possible d'utiliser :

Type de paramètre	Explication
`str`	*Chaîne de caractères non vide en excluant '/'*
`int`	*Entier positif*
`slug`	*Chaîne de caractères ASCII, exemple :* **`user-interface-name`**
`uuid`	*Format UUID :* **`075194d3-6885-417e-a8a8-6c931e272f00`**
`path`	*Chaîne de caractères non vide en incluant '/'*

Architecture Django

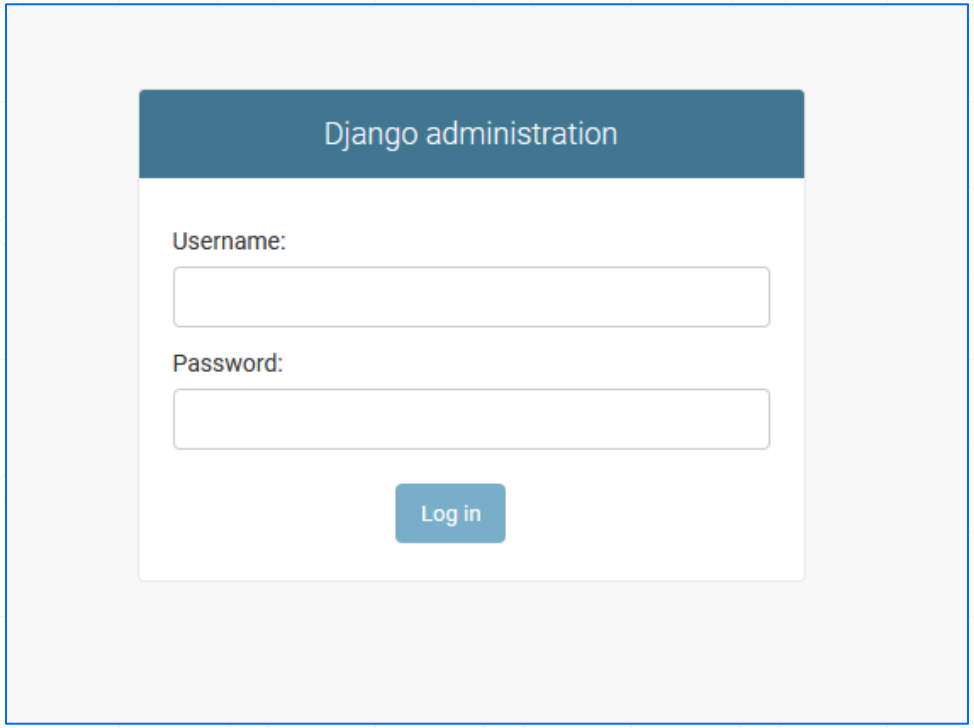


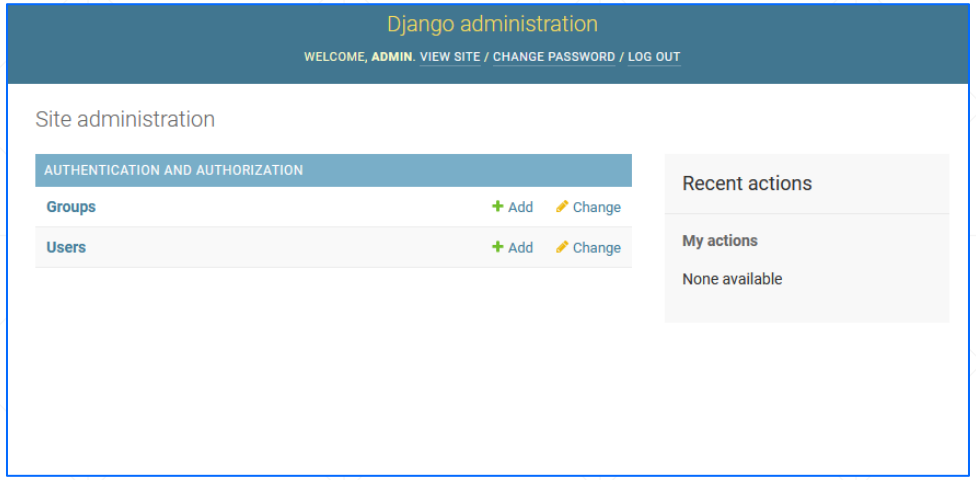
UQAR

Interface d'administration

- **Django** dispose d'une *interface d'administration* permettant en autre de :
  - Appliquer des configurations,
  - Visualiser et manipuler les données.
- Afin d'avoir accès à l'interface d'administration, il faut :
  1. Créer un nouvel utilisateur en lançant : **`python manage.py createsuperuser`**
  2. Entrer : *nom d'utilisateur*, *email* et *mot de passe*,
  3. Relancer le serveur et se rendre sur : `http://127.0.0.1:8000/admin`

Interface d’administration





Mise en pratique

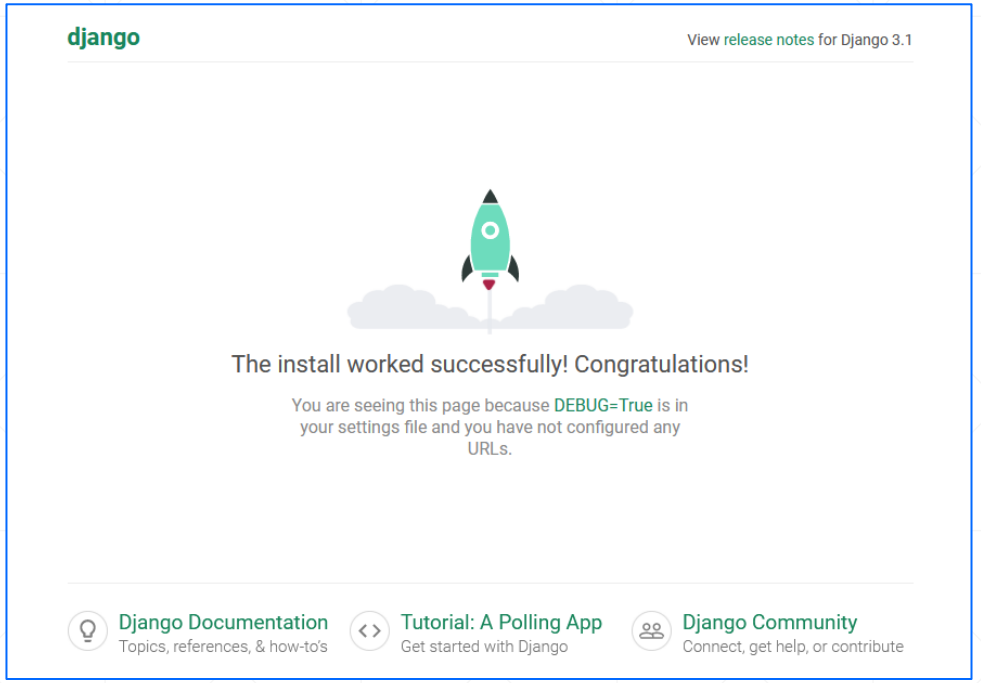
UQAR

Création d'un projet avec Django

1. Au préalable, il faut s'assurer d'être dans le bon environnement virtuel :
   - Commande : **`.\venv_django\Scripts\activate`**
2. Ensuite, Il faut créer le projet :
   - Commande : **`django-admin startproject project_name .`**
3. Afin de vérifier que tout fonctionne, on exécute le projet :
   - Commande : **`python manage.py runserver`**
4. Il y aura des erreurs (migration/base de données) qu'on résout avec :
   - Commande : **`python manage.py migrate`**
5. En relançant le projet, ça devrait fonctionner correctement.

Création d'un projet avec Django

En se rendant à l'adresse http://127.0.0.1:8000/ on obtient :



UQAR

Création d'une première application

1. Afin de créer un nouvelle application associée au projet *project_name* il faut :
   - Commande : **`django-admin startapp`** *app_name*
2. Il faut ensuite déclarer l'application dans le projet :
   - Éditer le fichier : *project_name/**settings.py***,
   - Rajouter la ligne '**app_name.**apps.**AppName**Config' dans INSTALLED_APPS.
3. Dans le dossier *app_name*, il faut créer un fichier **urls.py**,
4. Dans le fichier *project_name/urls.py*, il faut jouter dans **urlpatterns** :

```
path('app_name/', include('app_name.urls')),
```

5. Importer **include** à partir de **django.urls**.

UQAR

Nos premières pages

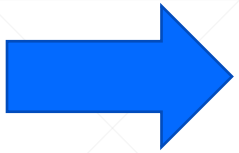
1. Dans le fichier app_name/**views.py** il faut créer les fonctions pour nos pages :

```
from django.http import HttpResponse

def home(request):
    return HttpResponse("<html><body><p>Page d'accueil !</p></body></html>")

def contact(request):
    return HttpResponse("<html><body><p>Page de contact !</p></body></html>")
```

2. Dans le fichier app_name/**urls.py** il faut associer les URLs aux fonctions :



UQAR

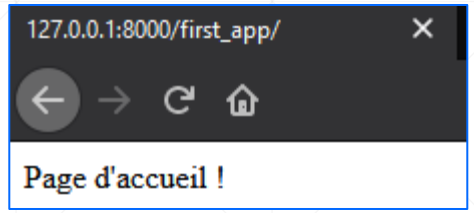
Nos premières pages

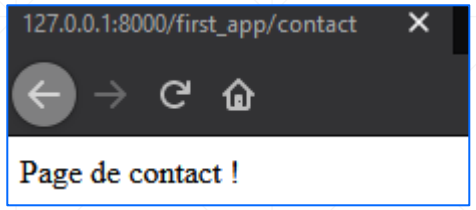
1. Code à utiliser :

```
from django.urls import path
from .views import home, contact

urlpatterns = [
    path('', home, name="home"),
    path('contact', contact, name="contact"),
]
```

2. Résultats sur dans le navigateur :
   - En allant à http://127.0.0.1:8000/app_name/
   - En allant à http://127.0.0.1:8000/app_name/contact





UQAR

Utilisation des paramètres d'URL

Afin de bénéficier d'un paramètre URL, il faut suivre les étapes :

1. Dans le fichier app_name/**urls.py**, il faut ajouter dans **urlpatterns** :

```
path('contact/<str:username>', contact, name="contact"),
```

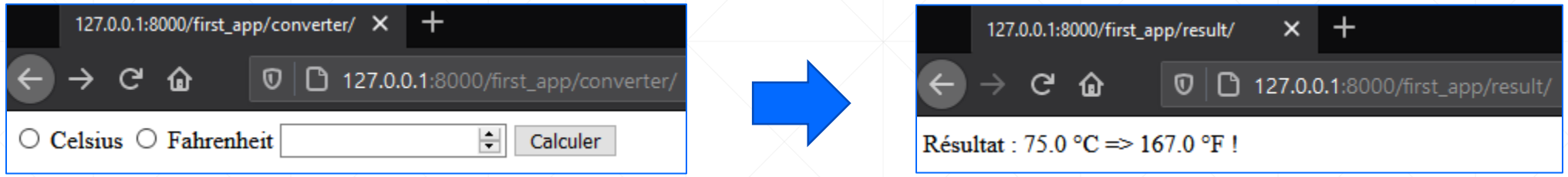
Nom d'utilisateur

2. Dans le fichier app_name/**views.py**, il faut ajouter :

```
def contact(request, username):
    return HttpResponse(
        "<html><body><p>Page de contact - {} !</p></body></html>".format(username)
    )
```

Démonstration

- Création d'une petite application pour la conversion entre :
  - **Degré Celsius °C** et **Degré Fahrenheit °F**
- Elle doit contenir :
  - Deux *boutons radio* pour : **Celsius** et **Fahrenheit**,
  - Un *champs nombre* pour entrer la valeur.
- Le résultat doit être retourné dans une autre page.



Questions & Discussion

UQAR

Bibliographie

1. Bersini, H, Alexis, P. & Degols, G. (2018). Apprendre la programmation web avec Python et Django. Eyrolles.
2. Samson, P. (2020). DJANGO Developpez vos applications web en Python. Édition ENI.
3. Haverbeke, M. (2014). Eloquent javascript: A modern introduction to programming. No Starch Press.
4. Melé, A. (2020). Django 3 By Example: Build powerful and reliable Python web applications from scratch. P
5. Hillar, G. C. (2018). Django RESTful Web Services: The Easiest Way to Build Python RESTful APIs and Web Services with Django. Packt Publishing Ltd.ackt Publishing Ltd.