

Créer le squelette du site : `django-admin.exe startproject mysite` .

(ne pas oublier le .)

`django-admin.py` est un script qui crée les dossiers et fichiers nécessaires pour vous. Vous devriez maintenant avoir une structure de dossier qui ressemble à celle-ci:

```
djangoirls
├── manage.py
├── mysite
│   ├── settings.py
│   ├── urls.py
│   ├── wsgi.py
│   ├── __init__.py
└── requirements.txt
```

Note : dans votre structure de dossier, vous pourrez voir également le répertoire `venv` que nous avons créé avant.

`manage.py` est un script qui aide à gérer ou maintenir le site. Entre autres, il permet notamment de lancer un serveur web sur notre ordinateur sans rien installer d'autre.

Le fichier `settings.py` contient la configuration de votre site web.

`urls.py` contient une liste de patterns d'urls utilisés par `urlresolver`

Ouvrir `mysite/settings.py` pour changer des paramètres du projet :

```
TIME_ZONE = 'Europe/Paris'
LANGUAGE_CODE = 'fr-fr'
```

Pour créer la base de données :

```
python manage.py migrate
```

Lancez le serveur pour voir si cela fonctionne :

```
Entrer dans le dossier mysite
python manage.py runserver 0.0.0.0:8000
```

Création de la base de données :

```
python manage.py migrate
```

Pour créer une app dans Django :

```
python manage.py startapp <nom de l'app>
```

Pour déclarer l'app dans Django :

Modifier dans le fichier `settings.py`

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'myblog'
]
```

Nous allons avoir besoin de créer les modèles dans notre app.

Il faut donc renseigner ces modèles dans le fichier `models.py`

Ici nous créons un modèle nommé Post
Les variables sont les propriétés du modèle

`__str__` est obligatoire
`publish()` va nous servir à sauvegarder nos données

```
from django.conf import settings
from django.db import models
from django.utils import timezone

class Post(models.Model):
    author = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    title = models.CharField(max_length=200)
    text = models.TextField()
    created_date = models.DateTimeField(default=timezone.now)
    published_date = models.DateTimeField(blank=True, null=True)

    def publish(self):
        self.published_date = timezone.now()
        self.save()

    def __str__(self):
        return self.title
```

Nous publierons notre modele en tapant :
`python manage.py makemigrations myblog`

Pour migrer notre blog :
`python manage.py migrate blog`

Pour pouvoir utiliser les Posts dans l'interface d'admin :
Ajouter les informations suivantes dans `myblog/admin.py`:

```
from django.contrib import admin
from .models import Post

admin.site.register(Post)
```

Nous devons créer le superuser pour pouvoir nous connecter et administrer le site :

`python manage.py createsuperuser`

Relancer le serveur avec `python manage.py runserver`

Ouvrir le navigateur à la page : `localhost:8000/admin`

Allez dans la rubrique Post pour en créer quelques uns.

Nous allons maintenant créer des vues
Dans le fichier `mysite/urls.py`
`from django.contrib import admin`
`from django.urls import path, include`

```
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path('', include('myblog.urls')),  
]
```

Puis créer le fichier `myblog/urls.py`

Et ajouter ceci :

```
from django.urls import path  
from . import views
```

```
urlpatterns = [  
    path('', views.post_list, name='post_list'),  
]
```

Dans le fichier `myblog/views.py` :

```
from django.shortcuts import render
```

```
# Create your views here.
```

```
def post_list(request):  
    return render(request, 'myblog/post_list.html', {})
```

Dans le dossier `myblog` créer un dossier `templates`
puis à l'intérieur, un autre dossier `myblog`

en relançant le serveur, vous devriez avoir une page blanche à l'adresse
<http://localhost:8000>

Cela veut dire que vous avez bien été redirigé vers la page `post_list.html`
Vous pouvez maintenant la modifier avec par exemple :

```
<html>  
<body>  
    <p>coucou</p>  
</body>  
</html>
```

Connectons maintenant notre page avec nos données :

Dans la console saisir :

```
$ python manage.py shell  
Puis :  
from myblog.models import Post  
et  
Post.objects.all()
```

Nous avons différents morceaux en place : le modèle `Post` qui est défini dans le fichier `models.py`, la vue `post_list` dans `views.py` et nous venons de créer notre template. Mais comment allons-nous faire pour faire apparaître nos posts dans notre template HTML ?

Dans `myblog/views.py`, on doit maintenant trouver:

```
from django.shortcuts import render
from django.utils import timezone
from .models import Post
```

```
# Create your views here.
```

```
def post_list(request):
    posts =
    Post.objects.filter(published_date__lte=timezone.now()).order_by('published
_date')
    return render(request, 'myblog/post_list.html', {'posts': posts})
```

maintenant que nous avons transmis les données à notre page, nous pouvons les afficher dans la page web `'myblog/post_list.html'` en la modifiant:

```
<html>
<body>
    <p>coucou</p>
    {% for post in posts %}
    {{ post }}
    {% endfor %}
</body>
</html>
```

Ou pour aller plus loin :

```
<div>
    <h1><a href="/">Mon Blog</a></h1>
</div>

{% for post in posts %}
    <div>
        <p>published: {{ post.published_date }}</p>
        <h2><a href="">{{ post.title }}</a></h2>
        <p>{{ post.text|linebreaksbr }}</p>
    </div>
{% endfor %}
```