



# Django





# Wprowadzenie do Django

Framework Django to popularne narzędzie do tworzenia aplikacji webowych w języku Python. Zawiera wiele wbudowanych funkcji ułatwiających tworzenie aplikacji internetowych, takich jak ORM i szablony.







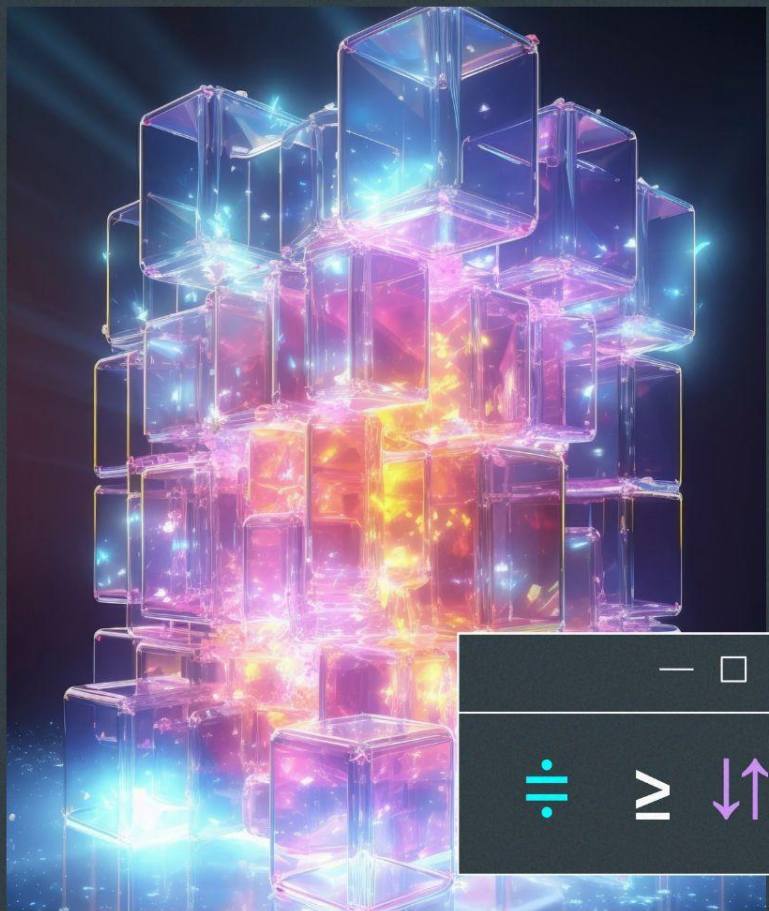
## Początki Django:

- Lawrence Journal-World newspaper
- Adrian Holovaty i Simon Willison
- Nazwa nawiązuje do Django Reinhardta
- Django Software Foundation (2008)

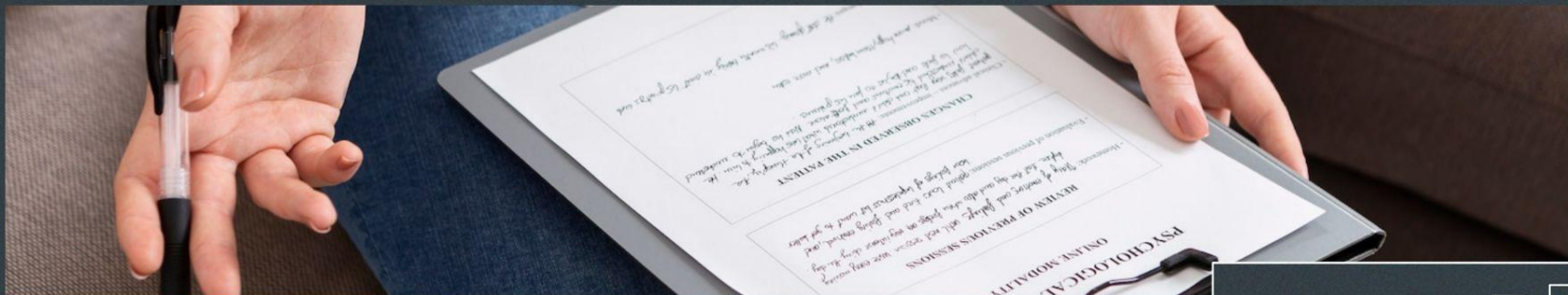


# Co oferuje?

- Elastyczność i skalowalność
- ORM
- Bezpieczeństwo
- Admin panel
- Modułowa struktura
- Wsparcie społeczności







## MVT

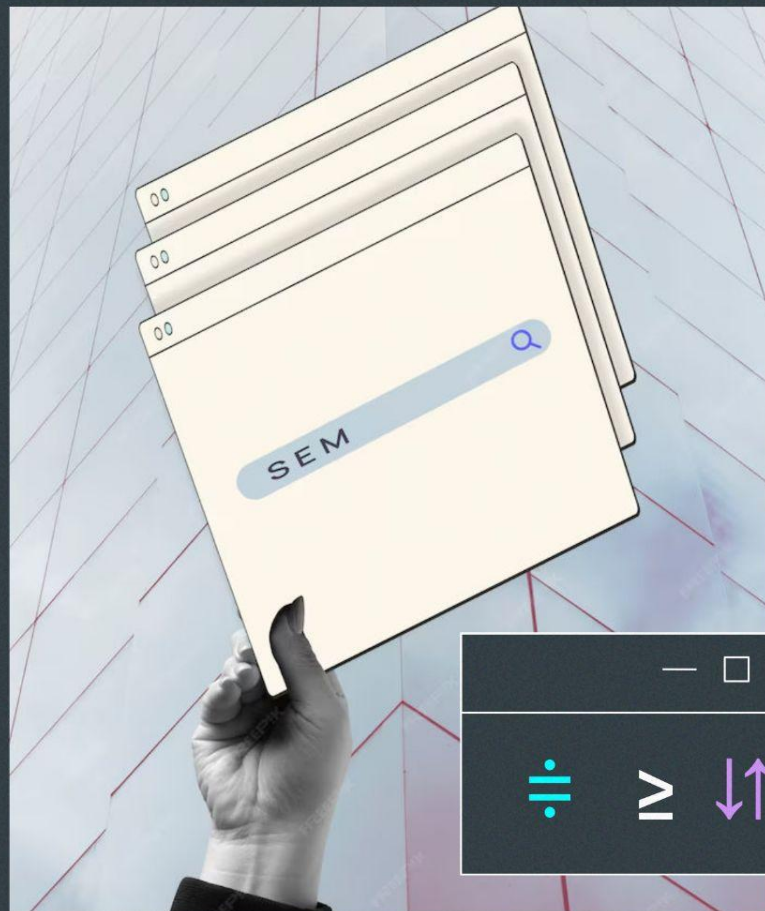
Model MVT - (Model-View-Template) jest wzorcem architektonicznym stosowanym w aplikacjach internetowych, szczególnie w frameworku Django.



# Zalety MVT



- Separacja logiki biznesowej od prezentacji
- Reużywalność komponentów
- Obsługa formularzy
- Zarządzanie sesją użytkownika
- Obsługa błędów





## Wady MVT:

- Pojedyncza odpowiedzialność widoku
- Wprowadzenie logiki do szablonów
- Zmiany w interfejsie użytkownika mogą wymagać zmian w widokach
- Trudność w testowaniu
- Ograniczenia w reużywalności kodu



# Wirtualne Środowisko

```
py -m venv myworld  
myworld\Scripts\activate.bat  
py -m pip install Django  
django-admin startproject storefront  
py manage.py startapp playground
```





# Struktura Plików

## ▼ STOREFRONT

- > .vscode
- > playground
- > storefront
- ≡ db.sqlite3
- 🔌 manage.py

## ▼ storefront

- > \_\_pycache\_\_
- 🔌 \_\_init\_\_.py
- 🔌 asgi.py
- 🔌 settings.py
- 🔌 urls.py
- 🔌 wsgi.py

## ▼ playground

- > \_\_pycache\_\_
- > migrations
- > static
- > templates
- 🔌 \_\_init\_\_.py
- 🔌 admin.py
- 🔌 apps.py
- 🔌 models.py
- 🔌 tests.py
- 🔌 urls.py
- 🔌 views.py



# Views



```
from django.shortcuts import render
from django.http import HttpResponse

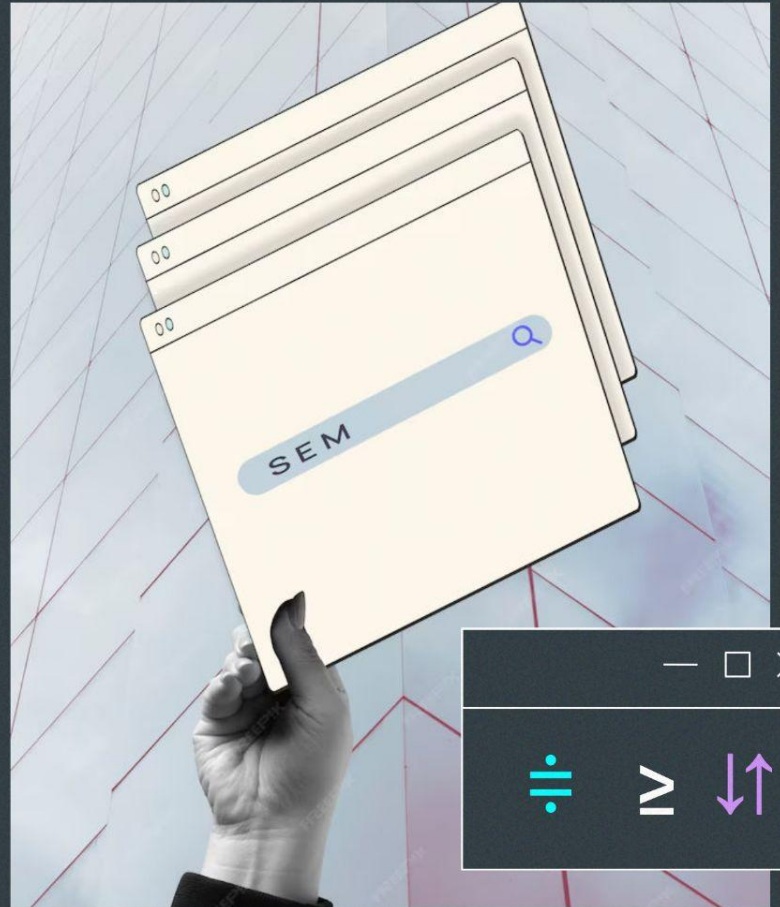
def members(request):
    return HttpResponse("Hello world!")
```



# URLs

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

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



# Template

```
<!DOCTYPE html>
<html>
<body>

<h1>Hello World!</h1>
<p>Welcome to my first Django project!</p>

</body>
</html>
```







# Models

```
from django.db import models

class Member(models.Model):
    firstname = models.CharField(max_length=255)
    lastname = models.CharField(max_length=255)
```

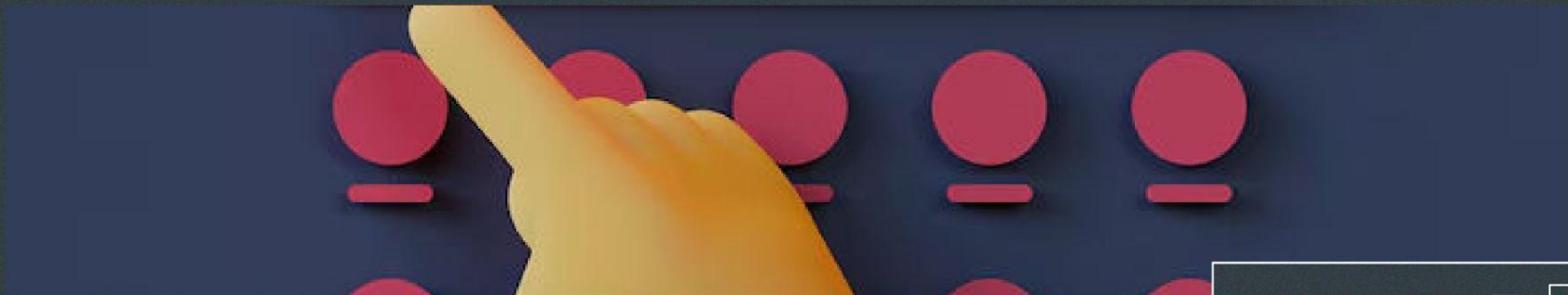


# Django Template Language (DTL)



- Język szablonów używany w Django
- Składnia podobna do Pythona
- Wbudowane filtry
- Warunki i pętle
- Wartości kontekstu
- Dziedziczenie szablonów
- Łatwa integracja z Pythonem
- Bezpieczeństwo





## Zmienne:

```
<h1>Witaj, {{ user.username }}!</h1>
```

## Filtry:

```
{{ value|lower }}
```



Tagi:



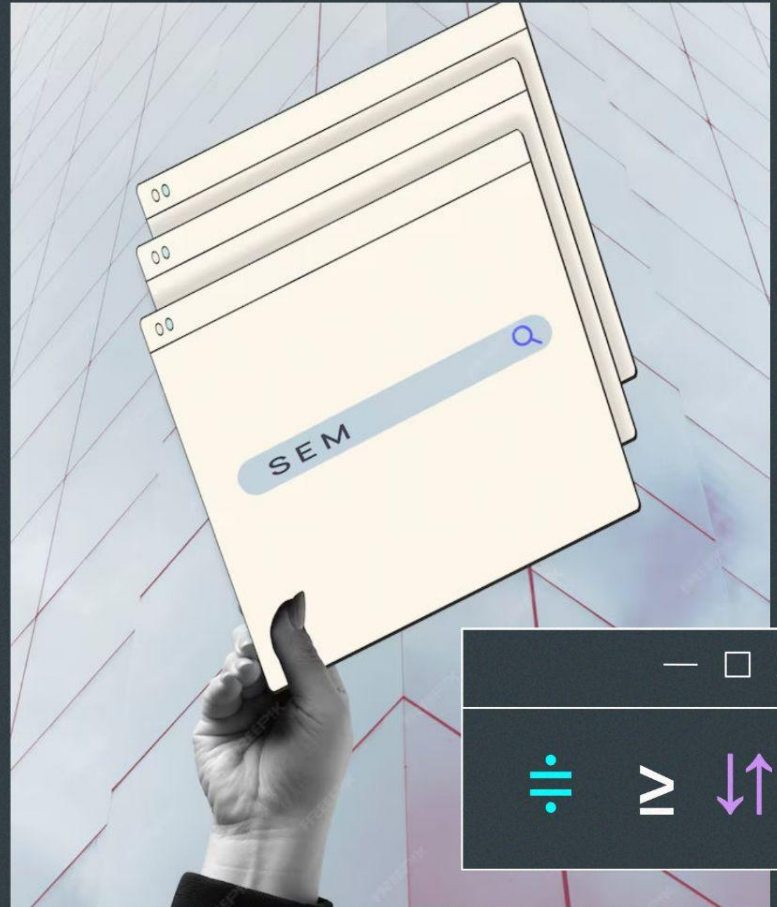
```
<ul>
```

```
{% for item in items %}
```

```
    <li>{{ item }}</li>
```

```
{% endfor %}
```

```
</ul>
```





# Dziedziczenie szablonów i szablony warunkowe

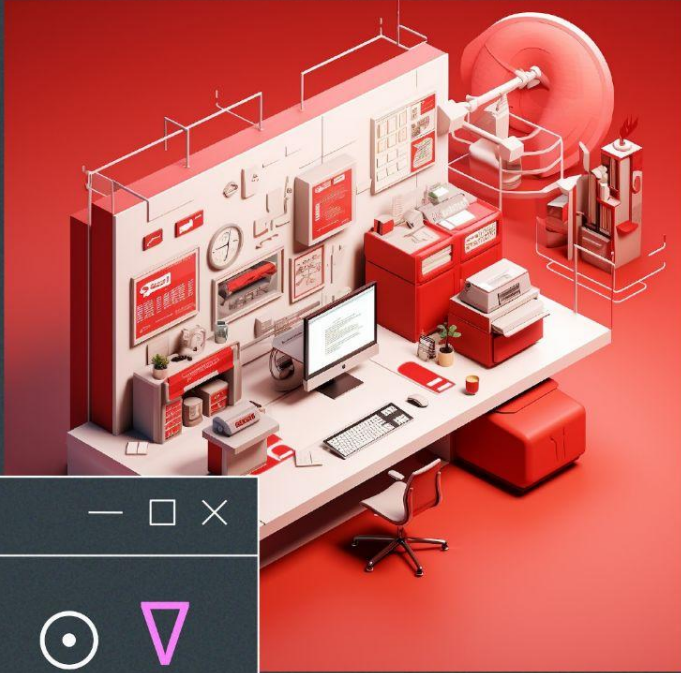


```
{% extends "base.html" %}
```

```
{% if user.is_authenticated %}  
    Witaj, {{ user.username }}!  
{% else %}  
    Witaj, gościu!  
{% endif %}
```



# Django REST framework:



- Narzędzie do tworzenia interfejsów API
- Bogata funkcjonalność
- Serializacja danych
- Widoki oparte na klasach
- Autentykacja i uprawnienia
- Obsługa formatów danych
- Dokumentacja
- Wsparcie dla wewnętrznych funkcji Django



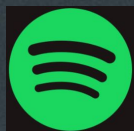
Gdzie znajdziemy Django?



YouTube



Instagram



Spotify



## Źródła:

<https://www.w3schools.com/django/index.php>

[https://en.wikipedia.org/wiki/Django\\_\(web\\_framework\)](https://en.wikipedia.org/wiki/Django_(web_framework))

[https://pl.wikipedia.org/wiki/Django\\_Reinhardt](https://pl.wikipedia.org/wiki/Django_Reinhardt)

<https://geek.justjoin.it/flask-vs-django-dlaczego-nie-warto-wybierac-flaska-w-2021-roku/>





Dziękujemy!



Szymon Wątroba, Dorian Sraga

