

DJANGO 2.0

Uma breve introdução

The Django logo, featuring the word "django" in a white, lowercase, sans-serif font, set against a dark teal rectangular background.



Começando...

- O que é Django?
- Por que usar Django?
- O que é preciso para usar Django?
- Como configurar o ambiente?

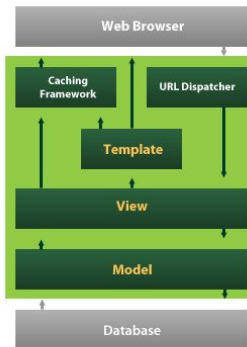
O que é Django?

É um framework para aplicações web gratuito e de código aberto, escrito em Python. Um web framework é um conjunto de componentes que ajuda você a desenvolver sites de forma mais rápida e fácil.

<https://www.djangoproject.com/>

django

5. Templates typically return HTML pages. The Django template language offers HTML authors a simple-to-learn syntax while providing all the power needed for presentation logic.
4. After performing any requested tasks, the view returns an HTTP response object (usually after passing the data through a template) to the web browser. Optionally, the view can save a version of the HTTP response object in the caching system for a specified length of time.



1. The URL dispatcher (`urls.py`) maps the requested URL to a view function and calls it. If caching is enabled, the view function can check to see if a cached version of the page exists and bypass all further steps, returning the cached version, instead. Note that this page-level caching is only one available caching option in Django. You can cache more granularly, as well.
2. The view function (usually in `views.py`) performs the requested action, which typically involves reading or writing to the database. It may include other tasks, as well.
3. The model (usually in `models.py`) defines the data in Python and interacts with it. Although typically contained in a relational database (MySQL, PostgreSQL, SQLite, etc.), other data storage mechanisms are possible as well (XML, text files, LDAP, etc.).



Por que usar Django?

Pros

- A aplicação é desenvolvida de forma acelerada
- Segurança
- *Batteries Included*
- Escalabilidade



Cons

- Velocidade
- Padronização pobre
- Não é a melhor opção para aplicações pequenas e simples

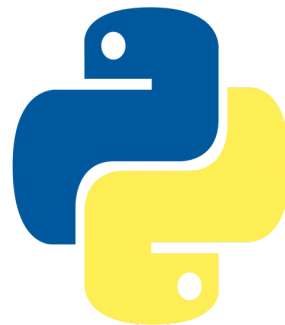
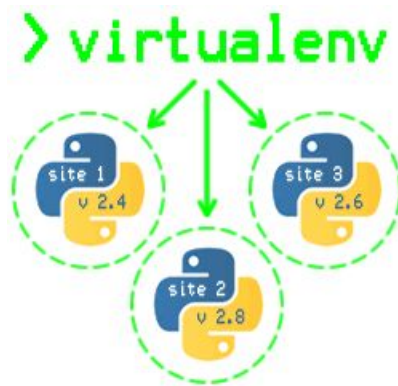


Pré-requisitos para começar em Django

- Python - Biblioteca de interpretação de código
- Pip - Package manager
- Virtual Environment - Ambiente virtual
- Django - Framework
- Editor de texto
- Terminal - Executor



django





Instalando e configurando o ambiente

1º - Instalar a biblioteca Python*: `sudo apt install python3-pip | brew install python3`

2º - Atualizar o pip: `python -m pip install --upgrade pip`

3º - Criar Virtual Environment: `python3 -m venv myvenv`

4º - Instalar versão desejada do Django:

Em `requirements.txt`, adicionar a linha: `Django~=2.0.6`

`pip install -r requirements.txt`

5º - Criar projeto Django: `django-admin startproject <nome_do_projeto>`

https://tutorial.djangogirls.org/pt/django_installation/



Bibliografia e Referências

- Django Girls Tutorial: <https://tutorial.djangogirls.org/pt/>
- Django Tutorial: <https://docs.djangoproject.com/en/2.2/intro/tutorial01/>
- Vídeoaula de introdução ao Django: <https://www.youtube.com/watch?v=F5mRW0jo-U4>
- Documentação Python: <https://www.python.org/doc/>
- Aspectos Django:
<https://www.netguru.com/blog/pros-and-cons-of-django-as-framework-for-python-development>
- Apostila gratuita em português de Django:
<https://pythonacademy.com.br/assets/ebooks/desenvolvimento-web-com-python-e-django/desenvolvimento-web-com-python-e-django.pdf>