



Flask

web development,
one drop at a time

Testes e #JustPython

Eduardo Mendes



Nome:

Eduardo Mendes

Contatos:

{facebook, github, gist
instagram, linkedin,
telegram, twitter}/dunossauro



Eduardo Mendes

3.034 inscritos

INÍCIO

VÍDEOS

PLAYLISTS

CANAIS

DISCUSSÃO

SOBRE



Envios REPRODUZIR TODOS



Live de Python #62 -
Programação orientada a...

471 visualizações •
Transmitido 3 dias atrás



Live de Python #61 -
Programação orientada a...

839 visualizações •
Transmitido 1 semana atrás



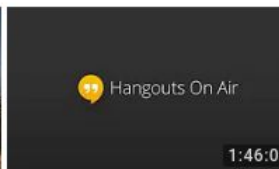
Live de Python #60 -
Exceções

582 visualizações •
Transmitido 2 semanas atrás



Live de Python #59 - Objetos
assíncronos

435 visualizações •
Transmitido 3 semanas atrás



Python funcional - Aula #3 -
Despacho, closures e...

376 visualizações •
Transmitido 4 semanas atrás



Live de Python #58 -
Requests assíncronos -...

494 visualizações •
Transmitido 1 mês atrás

Envios mais famosos REPRODUZIR TODOS



Live de Python #1 - Testes
com Python [Uma introduçã...

3,5 mil visualizações •
Transmitido 1 ano atrás



Live de Python #35 - Redes
neurais usando a biblioteca...

2,7 mil visualizações •
Transmitido 7 meses atrás



Live de Python #20 -
Selenium / Web scraping #1

2,4 mil visualizações •
Transmitido 11 meses atrás



Live de Python #21
Beautifulsoup / Web scrapi...

2,1 mil visualizações •
Transmitido 11 meses atrás



Como eu configuro meu atom
para programar Python?

1,7 mil visualizações •
3 meses atrás



Github foi comprado pela
microsoft? Vamos...

1,5 mil visualizações •
2 meses atrás

Roteiro

- JustPython
- Flask
- Testes
- Ferramentas “extras”
- Estrutura do projeto
- Coisas que você deve saber antes de começar
- CODE!!!



github.com/dunossauro/flask_test
s

Vai estar tudo aqui

#JustPython

JustPython é um manifesto ao simples, uma ode ao Python puro.

justpython.style

Flask



[overview](#) // [docs](#) // [community](#) // [extensions](#) // [donate](#)

Flask is a microframework for Python based on Werkzeug, Jinja 2 and good intentions. And before you ask: It's [BSD licensed](#)!

Flask is Fun

Latest Version: [1.0.2](#)

```
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello World!"
```

Testes

- Testes são a parte mais importante do desenvolvimento
- Você sabe se o que fez funciona sem testar?
- Pq não automatizar seus testes?

Só vamos falar de testes unitários
(só tenho 40 minutos)

Ferramentas extras

A ferramenta mais pythonica que existe pra automatizar tarefas e gerenciar ambientes virtuais de várias versões do python ao mesmo tempo.



<https://tox.readthedocs.io/>

Ferramentas extras

A ferramenta mais pythonica que existe pra automatizar tarefas e gerenciar ambientes virtuais de várias versões do python ao mesmo tempo.

 tox.ini

update test envs



3 months ago

 README.rst

Flask




<https://tox.readthedocs.io/>

 tox.ini	Merge remote-tracking branch 'origin/master' into switch-to-pallets-t...	18 days ago
 werkzeug-import-rewrite.py	Some fixes	3 years ago


 [README.rst](#)

Werkzeug

 tox.ini	update test envs	3 months ago
---	------------------	--------------

 [README.rst](#)

Flask

 tox.ini	temporarily pin pytest to avoid test bug	17 days ago
---	--	-------------

 [README.rst](#)

Jinja2

Ferramentas extras

O Coverage vai nos mostrar o que estamos esquecendo de testar. Coverage é uma métrica um tanto quanto polêmica, você pode cobrir tudo, mas fazer testes ruins.... É, não vai adiantar muito

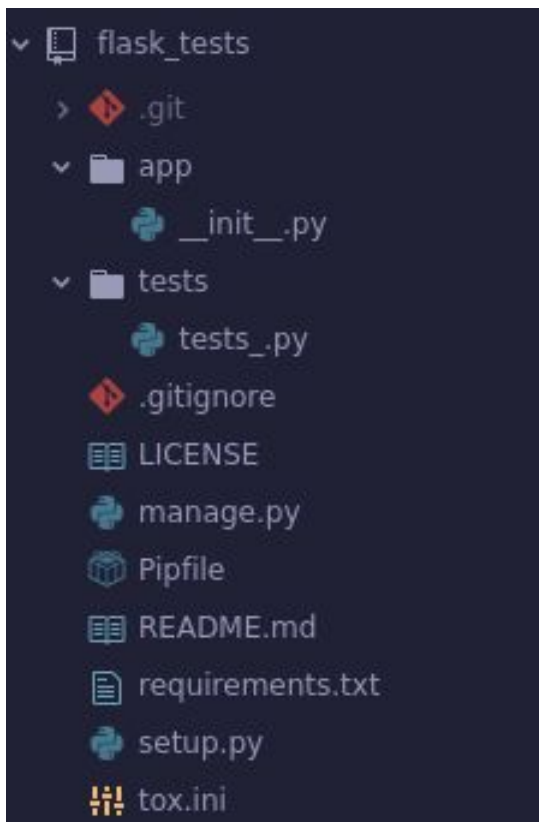
3. Use `coverage report` to report on the results:

```
$ coverage report -m
```

Name	Stmts	Miss	Cover	Missing
my_program.py	20	4	80%	33-35, 39
my_other_module.py	56	6	89%	17-23
TOTAL	76	10	87%	

<https://coverage.readthedocs.io>

Estrutura que EU gosto



- Todo o código da aplicação em um path
- Todos os testes em outro path
- Um manage.py pra gerenciar o que executar
- tox.ini pra ajudar a salvar vidas
- Readme pq ninguém tem bola de cristal
- License pq se você não fizer software livre você tem problemas mentais
- .gitignore pq ninguém quer saber qual editor você está usando, nem commitar pyc

Coisas que você deve saber antes de iniciar

```
self.app.testing = True  
self.app_context = self.app.test_request_context()  
self.app_context.push()  
self.client = self.app.test_client()
```

1. Coloca a classe Flask em modo de teste
2. Pede ao app um contexto de testes
3. Efetivamente coloca a aplicação nesse modo
4. Da acesso aos requests (geralmente o test_client é usado com with)

CODE !!