

# Flask and you

Luciano Camargo Cruz  
lccruz

**WWW**

**Python**

**Flask**

...



<https://growthboulevard.com/careers/>

**Pessoas > Tecnologia**

Web

# HTTP - HyperText Transport Protocol

- 1990 - Conceito de hipertexto
  - foi feito para transportar texto
- Tim Berners-Lee
- Para descrever os documentos, foi inventada uma linguagem de marcação chamada HTML

- First website

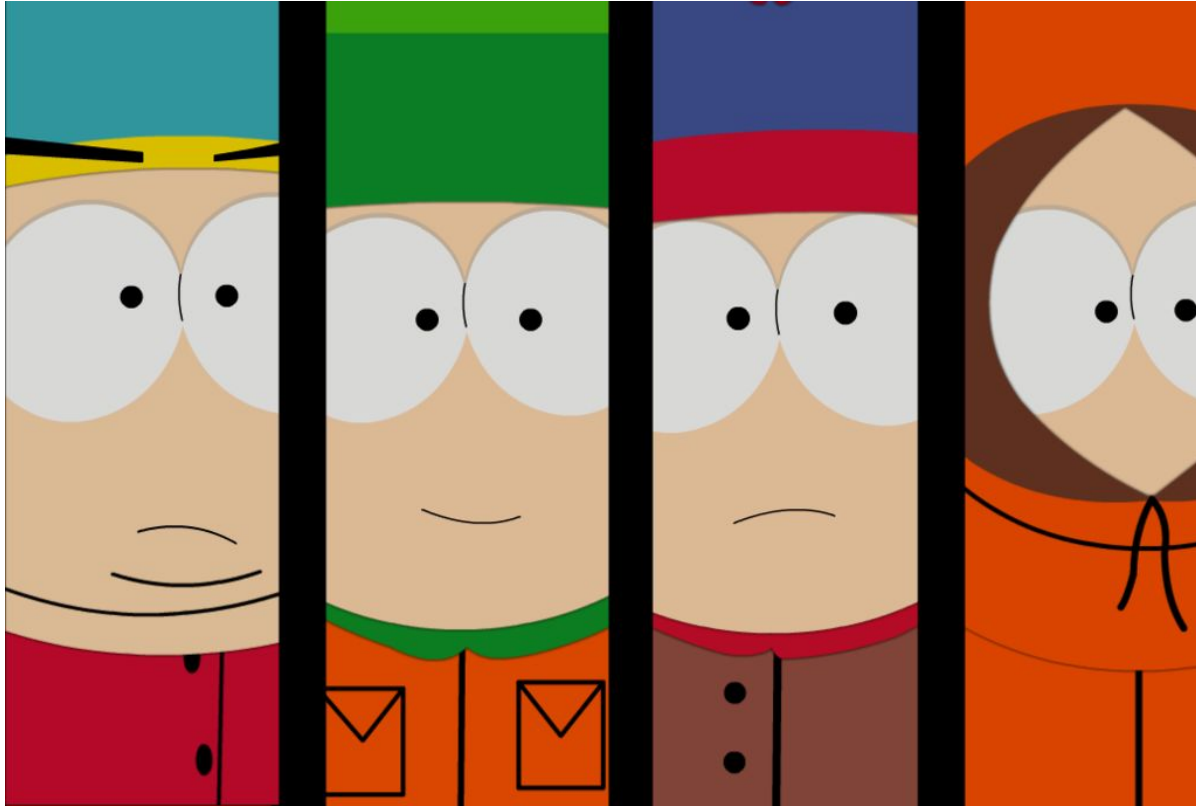


## **http://info.cern.ch - home of the first website**

From here you can:

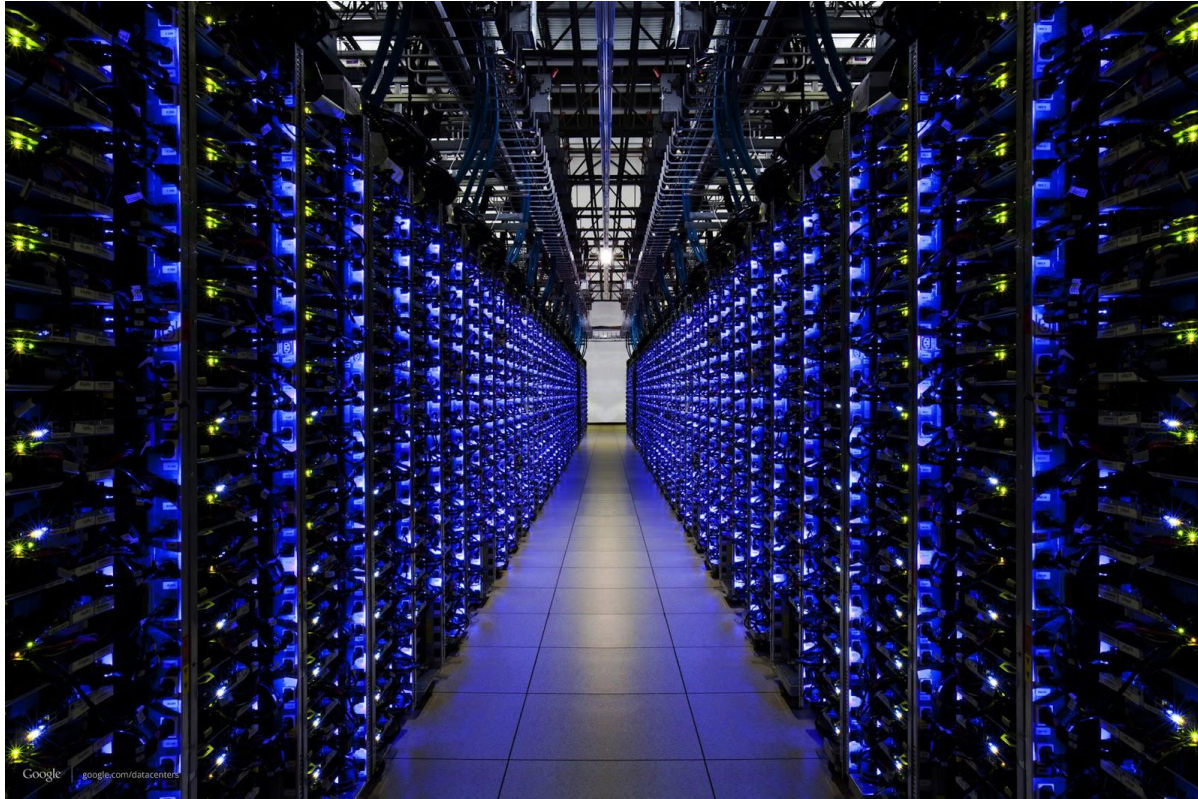
- [Browse the first website](#)
- [Browse the first website using the line-mode browser simulator](#)
- [Learn about the birth of the web](#)
- [Learn about CERN, the physics laboratory where the web was born](#)

**Para acessar um website, precisamos ?**

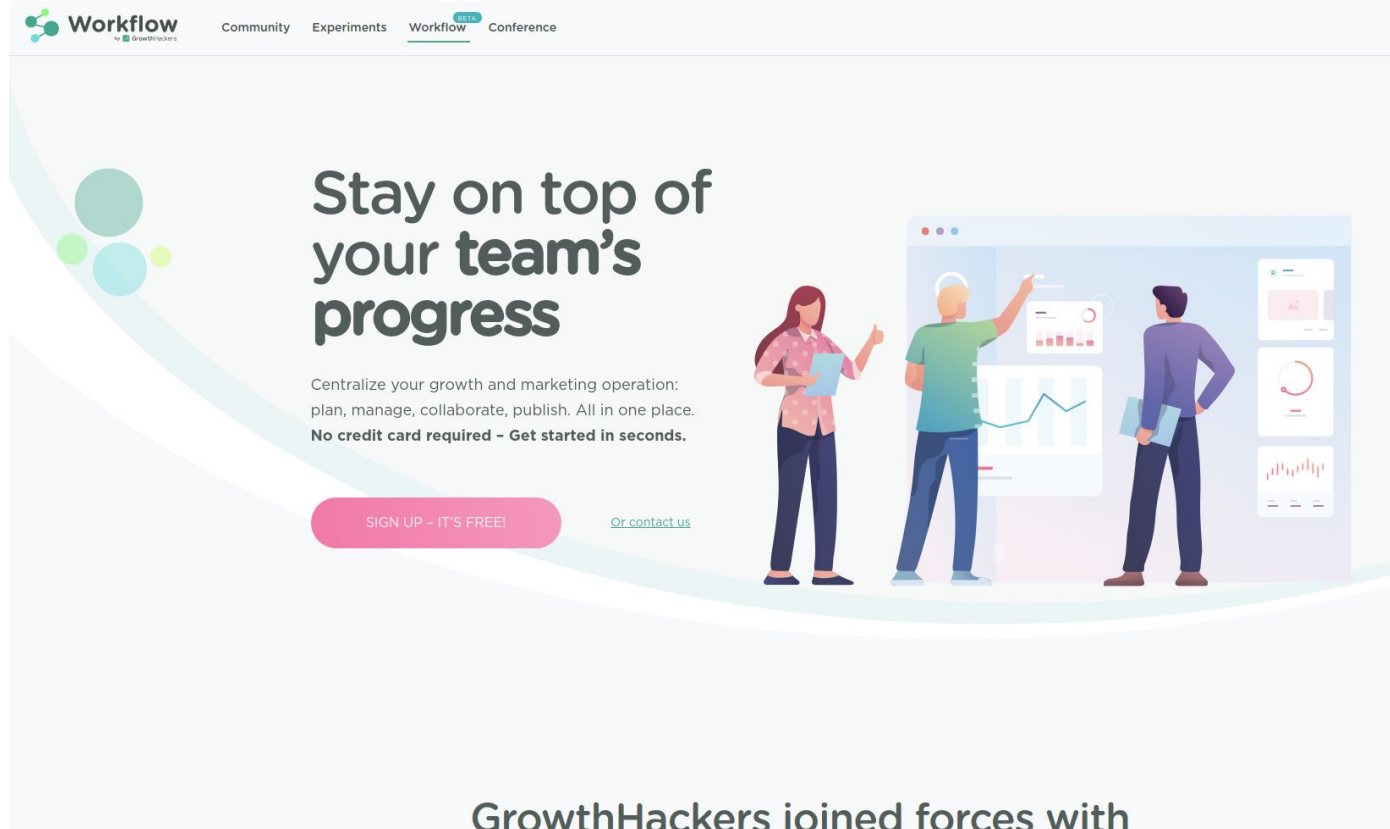




1 - O servidor onde a página está hospedada.



## 2 - A página.



The screenshot shows the landing page for 'Workflow by GrowthHackers'. The navigation bar at the top includes 'Community', 'Experiments', 'Workflow' (highlighted with a 'BETA' badge), and 'Conference'. The main headline reads 'Stay on top of your team's progress'. Below this, a sub-headline states: 'Centralize your growth and marketing operation: plan, manage, collaborate, publish. All in one place. No credit card required - Get started in seconds.' A pink button labeled 'SIGN UP - IT'S FREE!' is positioned to the left of a link that says 'Or contact us'. On the right side, an illustration depicts three people (two men and one woman) standing in front of a large digital screen displaying various data visualizations like bar charts, line graphs, and pie charts. The overall design is clean and modern with a light blue and white color palette.

Workflow by GrowthHackers

Community Experiments Workflow <sup>BETA</sup> Conference

# Stay on top of your team's progress

Centralize your growth and marketing operation: plan, manage, collaborate, publish. All in one place.  
**No credit card required - Get started in seconds.**

[SIGN UP - IT'S FREE!](#) [Or contact us](#)

GrowthHackers joined forces with

## 3 - A rede de comunicação.



4 - O navegador para exibir a página.





*“Python é uma Linguagem de Programação que nos permite trabalhar de forma rápida e integrar sistemas de forma eficaz.” ( [python.org](https://python.org) )*

- Código aberto
- Alto nível
- Criada em 1989 por Guido Van Rossum
- O nome foi inspirado no seriado britânico Monty Python
- Permite focar no problema, sem perder tempo na sintaxe
- Alta produtividade
- Comunidade livre, forte, diversificada, alegre e acolhedora
- Linguagem de verdade: orientada a objetos, funcional, estruturada
- **Tudo é um objeto**
- Legibilidade do código fonte



Flask







*Hey! Are you gonna pay for that?*



# Flask

Flask is a framework for Python based on Werkzeug, Jinja 2 and good intentions.  
And before you ask: It's BSD licensed!

# Flask

**Werkzeug** é uma biblioteca para desenvolvimento de apps **WSGI** (*Web Server Gateway Interface*). Possui a implementação básica para interceptar requests e lidar com response, controle de cache, cookies, status HTTP, roteamento de urls ...

**Jinja2** é um template engine escrito em Python.

**Good Intentions:** O Flask é Pythonico! Liberdade de estruturar seu app da maneira que desejar.

## Flask

Flask is a lightweight [WSGI](#) web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around [Werkzeug](#) and [Jinja](#) and has become one of the most popular Python web application frameworks.

Flask offers suggestions, but doesn't enforce any dependencies or project layout. It is up to the developer to choose the tools and libraries they want to use. There are many extensions provided by the community that make adding new functionality easy.

## Installing

Install and update using [pip](#):

```
pip install -U Flask
```

## A Simple Example

```
from flask import Flask

app = Flask(__name__)

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

```
$ env FLASK_APP=hello.py flask run
* Serving Flask app "hello"
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

## Contributing

For guidance on setting up a development environment and how to make a contribution to Flask, see the [contributing guidelines](#).

# Flask

on Jun 28, 2011	<b>0.7</b> ... fb1482d zip tar.gz
on Dec 31, 2010	<b>0.6.1</b> ... 774b7f7 zip tar.gz
on Jul 27, 2010	<b>0.6</b> ... 5cadd9d zip tar.gz
on Jul 6, 2010	<b>0.5</b> ... 4c937be zip tar.gz
on Jun 18, 2010	<b>0.4</b> ... 1592c53 zip tar.gz
on May 28, 2010	<b>0.3.1</b> ... 6b3e616 zip tar.gz
on May 27, 2010	<b>0.3</b> ... ce6e4cb zip tar.gz
on May 11, 2010	<b>0.2</b> ... e0fa5ae zip tar.gz
on Apr 16, 2010	<b>0.1</b> ... 8605cc3 zip tar.gz

[Previous](#)[Next](#)

[pallets](#) / [flask](#)[Sponsor](#)[Watch](#)

2,270

[Star](#)

46,655

[Fork](#)

12,948

[Code](#)[Issues](#) 18[Pull requests](#) 3[Security](#)[Insights](#)Branch: **master**[Commits on Sep 24, 2019](#)**Merge branch '1.1.x'**

davidism committed 3 days ago ✓

Verified



a3f0782

**Merge branch '1.0.x' into 1.1.x**

davidism committed 3 days ago ✓

Verified



7 added 0 df

**Merge branch '0.12.x' into 1.0.x**

davidism committed 3 days ago ✓

Verified



563ef46

**Merge pull request #3376 from pallets/migrate-docs** ...

davidism committed 3 days ago

Verified



e007949

**update docs for new theme**

davidism committed 3 days ago

Verified



4ed2065

[Commits on Sep 23, 2019](#)**Merge pull request #3344 from hharutyunyan/fix-the-contibution-docs** ...

davidism committed 4 days ago ✓

Verified



4a8e75b

**Merge pull request #3346 from bearnun/master** ...





Let's go



# Flask

```
from flask import Flask
```

```
app = Flask(__name__)
```

```
@app.route('/')  
def hello_world():
```

```
    return 'Hello, World!'
```

```
if __name__ == '__main__':
```

```
    app.run(host='127.0.0.1', port='5000', debug=True)
```

# Web - REST

Representational State Transfer, é um modelo para se projetar arquiteturas de software distribuído.

REST é um dos modelos de arquitetura que foi descrito por **Roy Fielding**, um dos principais criadores do protocolo HTTP, em sua tese de doutorado e que foi adotado como o modelo a ser utilizado na evolução da arquitetura do protocolo HTTP.

# Web - REST

*“REST provides a set of architectural constraints that, when applied as a whole, emphasizes scalability of component interactions, generality of interfaces, independent deployment of components, and intermediary components to reduce interaction latency, enforce security, and encapsulate legacy systems.” [Roy Fielding]*

# Web - REST

## Comunicação Stateless

Requisições feitas a um serviço REST devem conter todas as informações para que sejam interpretadas e executadas. Qualquer informação de estado deve ser mantido no cliente.

*“We next add a constraint to the client-server interaction: communication must be stateless in nature, as in the client-stateless-server (CSS) style of Section 3.4.3 (Figure 5-3), such that each request from client to server must contain all of the information necessary to understand the request, and cannot take advantage of any stored context on the server. Session state is therefore kept entirely on the client.” [Roy Fielding]*

# Web - REST

## Métodos HTTP

Método	URI	Utilização
GET	/artistas	Recuperar os dados de todos os artistas.
GET	/artista/id	Recuperar os dados de um determinado artista.
POST	/artista	Criar um novo artista.
PUT	/artista/id	Atualizar os dados de um determinado artista.
DELETE	/artista/id	Excluir um determinado artista.

# Web - REST

## Códigos HTTP

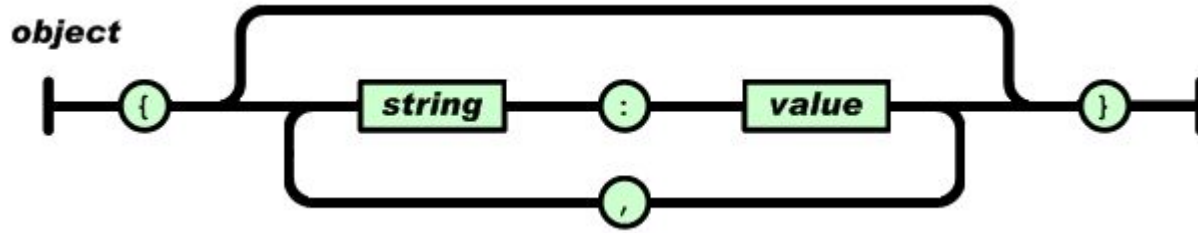
Método	Utilização
2xx	Requisição foi processada com sucesso
3xx	Indica ao cliente uma ação a ser tomada para que a requisição possa ser concluída.
4xx	Indica erro(s) na requisição causado(s) pelo cliente.
5xx	Indica que a requisição não foi concluída devido a erro(s) ocorrido(s) no servidor.

# Web - REST - 500

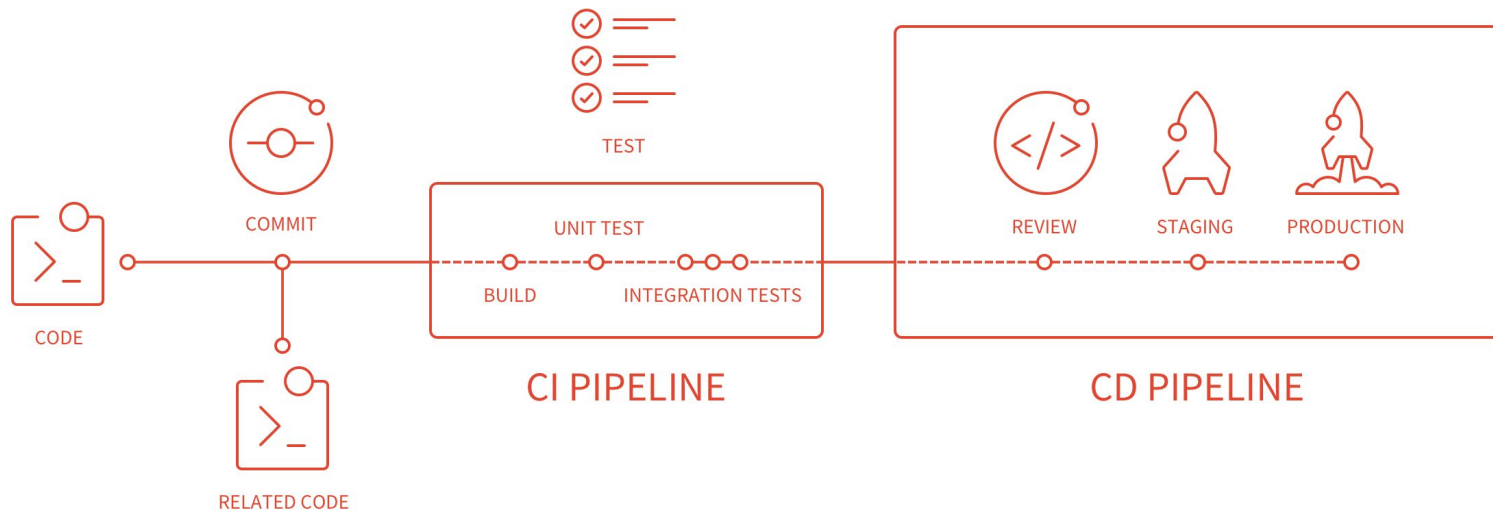




JSON - (*JavaScript Object Notation* - Notação de Objetos JavaScript) é uma formatação leve de troca de dados.



# Continuous Integration and Deployment (CI/CD)



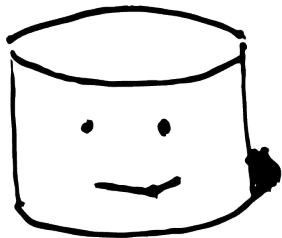


## The Python SQL Toolkit and Object Relational Mapper

SQLAlchemy is the Python SQL toolkit and Object Relational Mapper that gives application developers the full power and flexibility of SQL.

It provides a full suite of well known enterprise-level persistence patterns, designed for efficient and high-performing database access, adapted into a simple and Pythonic domain language.

# Marshmallow



marshmallow

**Simplified object serialization.**

In short, marshmallow schemas can be used to:

- Validate input data.
- Deserialize input data to app-level objects.
- Serialize app-level objects to primitive Python types. The serialized objects can then be rendered to standard formats such as JSON for use in an HTTP API.

# Flask-migrate

Flask-Migrate is an extension that handles SQLAlchemy database migrations for Flask applications using Alembic. The database operations are made available through the Flask command-line interface or through the Flask-Script extension.

Alembic is a lightweight database migration tool for usage with the SQLAlchemy Database Toolkit for Python.

# Flask-migrate

**> > pip install Flask-Migrate**

**flask db init**

**flask db migrate**

**flask db upgrade**



# Obrigado, Perguntas ?



**Luciano Camargo Cruz**  
**lccruz**

Repositório exemplo: <https://github.com/lccruz/helloflask>