Flask for Data Scientists

Introduction to Flask

Hello!

I’m Valentino.

You can find me at:

vc@valentino.io
https://github.com/vc1492a

Prerequisites

There's a few things we need for the demo later.

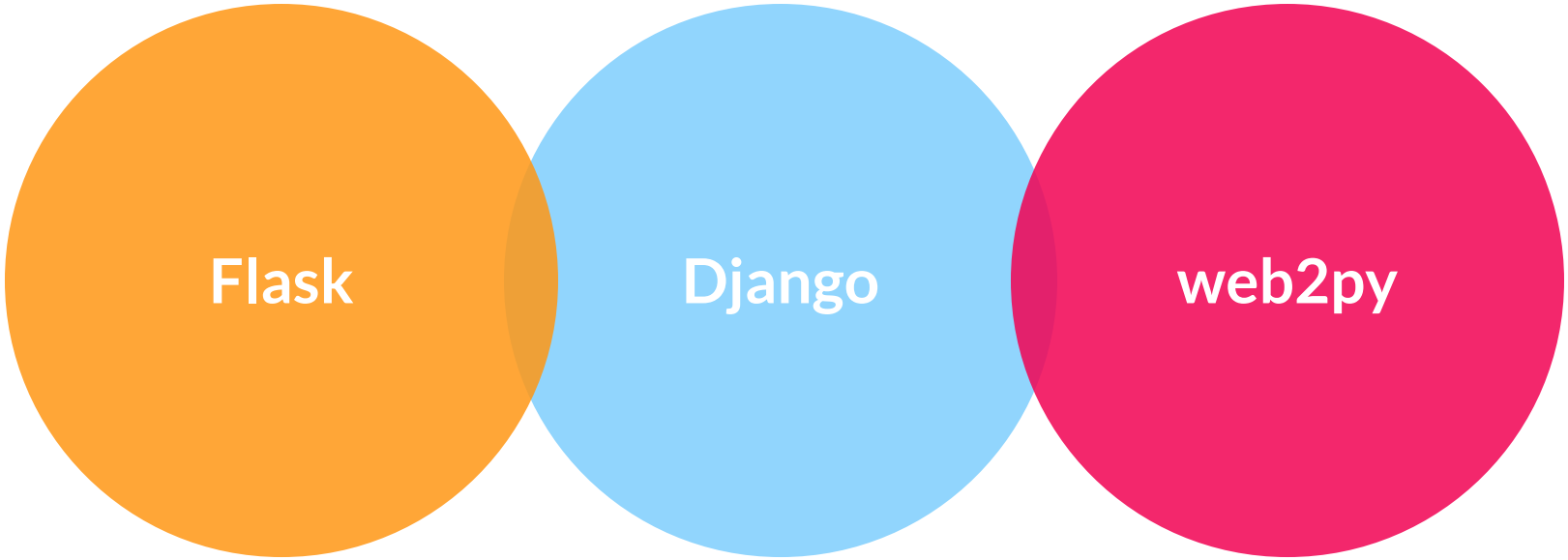
- PyCharm IDE or Sublime Text
- Python 3.5
  - If you're still using Python 2.7, shame on you!
- Werkzeug
- Jinja2
- Flask
- Pandas

1.
What is Flask?

“Flask is a microframework for Python based on Werkzeug, Jinja 2 and good intentions.” - Flask Team

Flask is a web framework written in Python that can be used to build web applications rapidly and easily.

Some of the Python web framework world.



There's many, many more...

Flask is easy.

Python is easy, therefore so is Flask.

Flask is fast.

Route setup requires just a few lines of code.

Flask is fun.

Building on top of Flask is intuitive.

What Flask Isn't.

Full-Stack Framework

Flask is a micro-framework, with no database, form validation, or other provided common functions. It's up to you!

If you want out-of-the-box support for things, use Django.

Built for High Traffic

While popular sites such as Pinterest use the Flask framework, Flask is not natively ready to support lots of web traffic. Good rule: less than ten (10) concurrent clients.

If you want robust, high-traffic reliability, use Django.

A Data Store

Django includes an ORM for storage out of the box. Flask does not, leaving you to supply SQL, MongoDB, or another database of your choice.

If you want a virtual object database, use Django.

But Django (IMO) is too bloated for small projects and has a steeper learning curve. Flask is ideal for most needs of data scientists.

2.
How can I use it?

How can I use it?

Personal Website

You can use Flask for your own personal website and portfolio. This is an ideal use case, requiring almost nothing on top of the base framework.

My website is built in Flask, using MongoDB and Flask-Login on top of the base framework.

Web Applications

Flask is natively ideal for minimal web applications without a substantial user base, high-traffic apps are also possible with added effort.

Popular websites Pintrest and LinkedIn use Flask in some way.

Dashboards

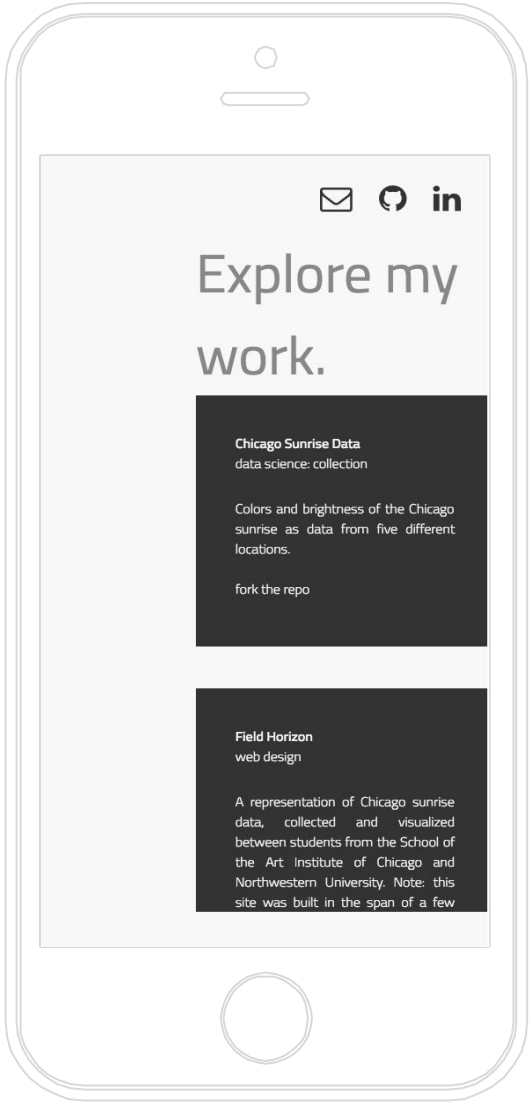
Flask can also be used for internal tools such as a dashboard incorporating data from business processes.

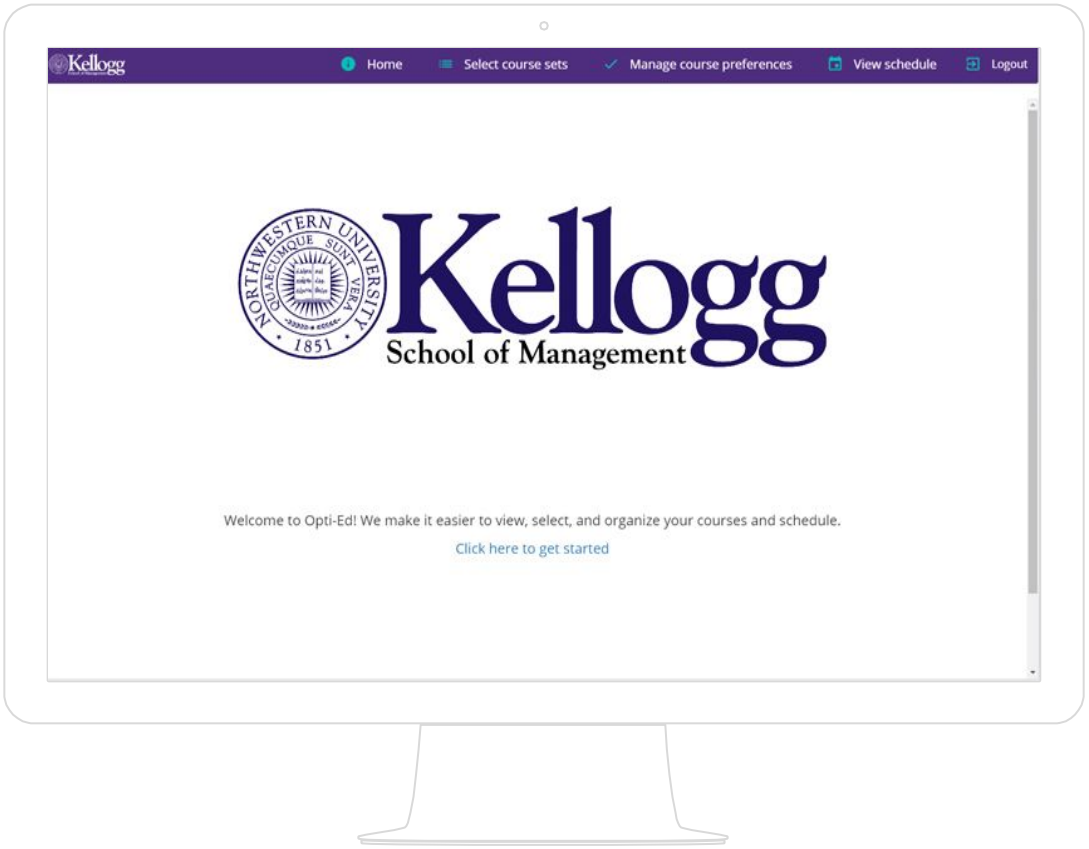
My summer 2016 project at JPL was an internal visualization tool built on top of Flask.

These are just a few examples of what's possible.

Personal Website

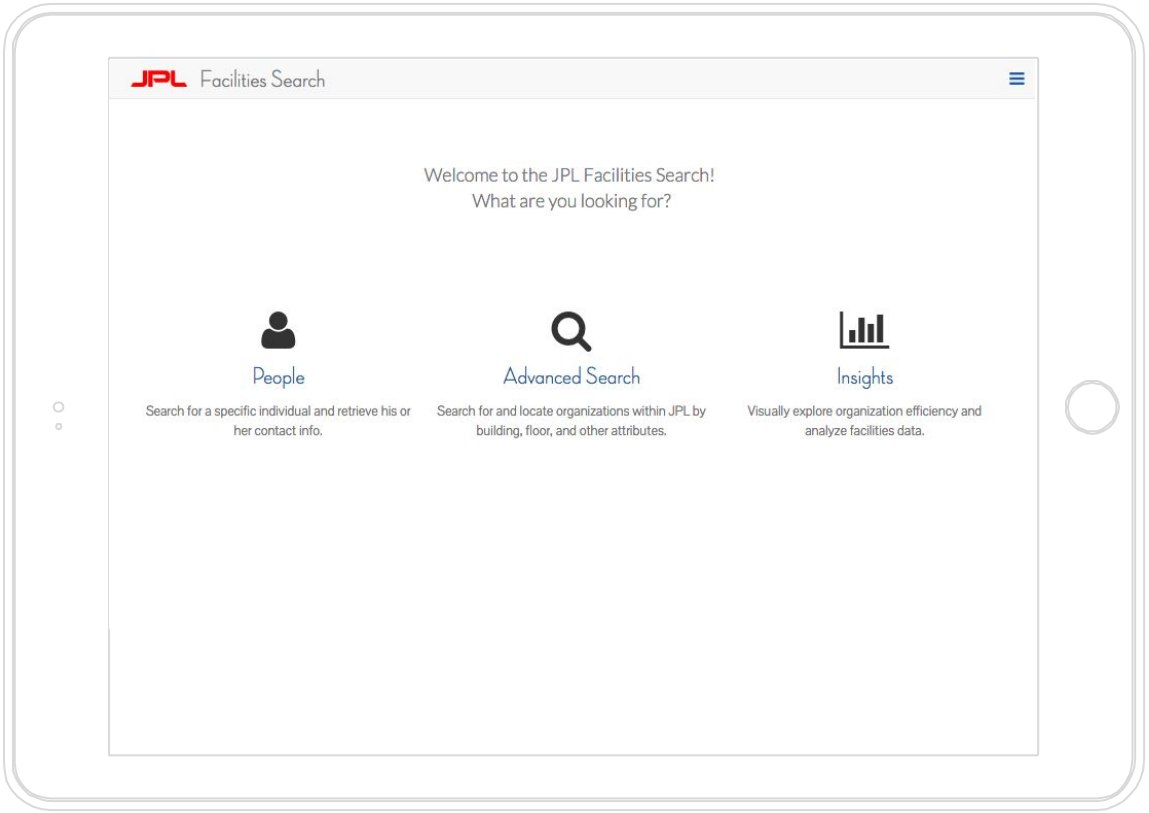
Build a portfolio to show all the neat work you've done.





Web Applications

Demonstrate the worth of your cool concept.



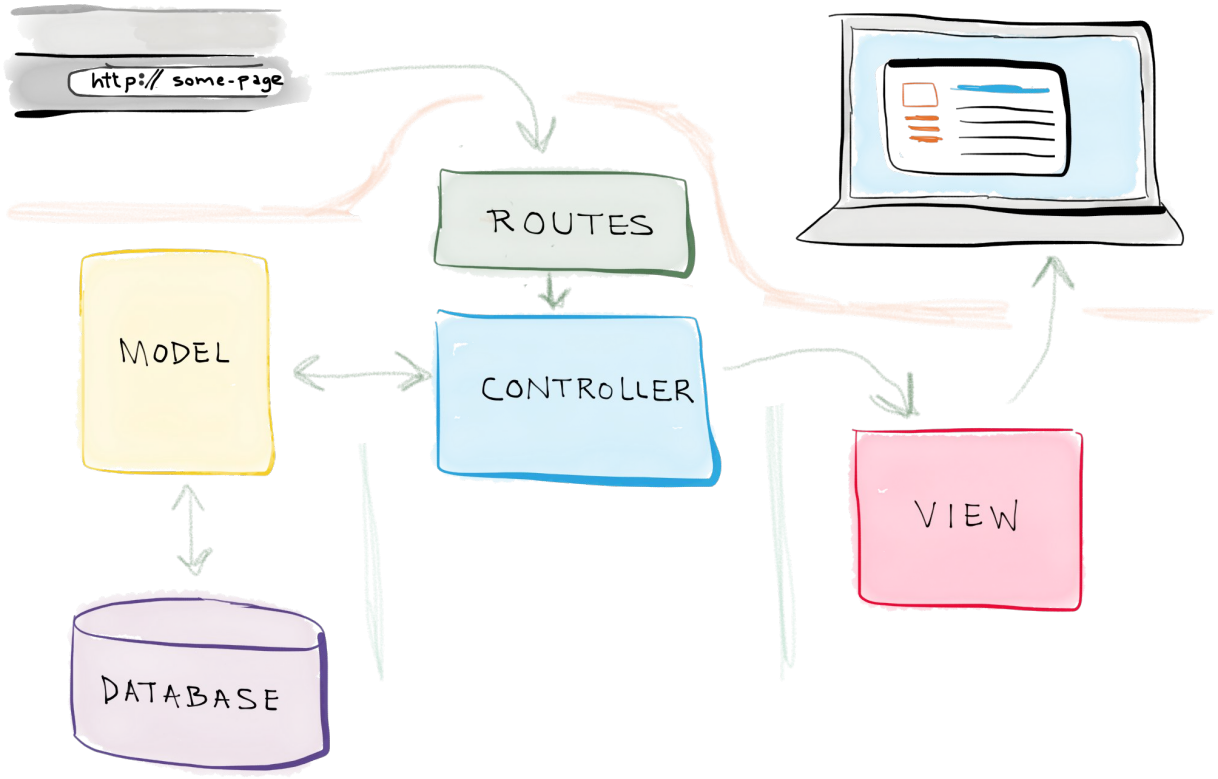
Dashboards

Use visualization to explore data or show the results of your model.

3.
How does it work?

MVC - Conceptual Overview

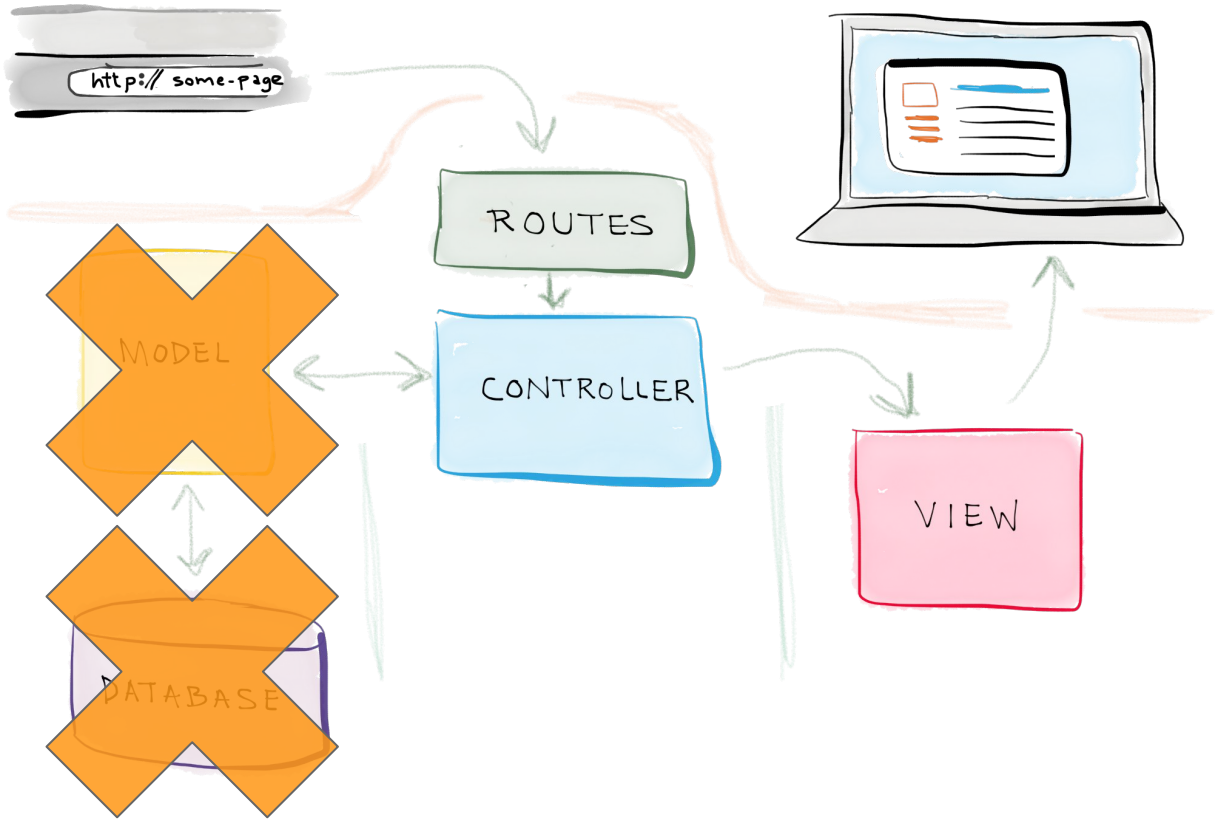
This is a typical **Model-View-Controller** architecture present in web frameworks such as Django.



source: goo.gl/uuQ7Eb

Flask - Conceptual Overview

Flask **does not use the MVC architecture**, and has no models to connect to a database natively.



source: goo.gl/uuQ7Eb

Routing - Simple Example

Just like most web frameworks, Flask uses routes. Routes pass objects to the front-end for display; HTML templates, JSON objects, Pandas dataframes, etc. Below is the simplest possible route.

```
# this is the route for our home page
# the @ symbol is called a decorator
@app.route('/')
def home():
    return 'hello world'
```

Routing - Passing Objects

Flask has great templating capability, and it's easy to pass objects into your templates.

```
@app.route('/blog/<url>', methods=['GET', 'POST'])
def blog_route(url):
    url = request.path.split('/')[-1]
    post = blog_posts.find_one({'post.url': url})
    session['blog_url'] = url
    return render_template('blog_template.html',
        post=post)
```

Routing - Calling Functions

You can call functions from your Flask routes, too. Here, we delete a blog post by calling the *remove_post* function. Q: does this app use the MVC framework?

```
@app.route('/delete_post', methods=['GET', 'POST'])
@login_required
def delete_post():
    url = session['blog_url']
    user = current_user.get_id()
    remove_post(url, user)
    return redirect('/profile')
```

Variables - Simple Example

Flask (well, Jinja 2) uses {{ }} symbols to refer to objects passed from your route. Here, we refer to the *post* object, which itself contains a *post* object and an attribute, *title*. It's organized as a dictionary.

```
<div class="title">{{ post.post.title }}</div>
<div class="subtitle">{{ post.post.subtitle }}</div>
```

Variables - For Loops

Using {% *some code* %} within your template allows you to execute more complex tasks.

```
{% for post in posts %}
  <div class="blog-title">
    <a href="/blog/{{ post.post.url }}">{{ post.post.title }}</a>
  </div>
  <div class="subtitle">{{ post.post.subtitle }}</div>
{% endfor %}
```

Variables - If Statements

You can also execute if statements as well. Here, we restrict views to only to users currently logged in to our app.

```
{% if current_user.is_authenticated %}
  <div class="menu-key" id="menu1">
    <a href="/delete_post" class="portfolio-link">delete</a>
  </div>
{% endif %}
```

Let's review some concepts.

Python is Awesome, Use It

Using Python for your data analysis work enables easy use of external APIs, web frameworks for integrating predictive models into products, and other tasks.

Flask is a Micro-Framework

Flask doesn't use an MVC architecture. Form validation, database abstraction, and other functionality is up to you to write and include!

Routes are Passageways

Flask uses routes for rendering templates and calling functions from within your app. You can define functions that can be called by your routes, too.

Variables are Handy

Use variables in your templates after passing objects to the front-end from your routes. Customize views using if statements.

Extensibility is Easy

Because Flask is a Python framework, it's easy to include new APIs, add in a MongoDB or SQL database, and other elements to increase functionality.

It's Good to Think Big

What you can do with Flask is really up to you and your imagination. Some ideas: Slack integration, optimization with PuLP, send texts with Twilio.

4.
Interactive Demo

Application setup

Flask application initialization.

Routing

Setting up routes and rendering templates.

Application setup

Flask application initialization.

Routing

Setting up routes and rendering templates.

Flask + Pandas

Using Flask with Pandas let's you do cool stuff.

https://goo.gl/94hgQn

Fork the repo, then pull to a local directory.

5.
Other Topics

Other Flask Topics

Deployment

Using Heroku with Flask is an easy, intuitive way to published production versions of your app.

MongoDB

Integrate a unstructured, schema-free database into your Flask app. Fun fact: MongoDB has mapReduce functionality.

Flask-Login

Build a secure login portal with Flask-Login, MongoDB, and a simple validation technique using Bcrypt.

Bare-bones Blogging

Text-only blogging functionality using simple HTML forms and MongoDB.

MailGun

Integrate MailGun services and send automated emails to users for password resets and other activity.

Prediction API

Build a prediction API using Flask with SciPy, and run models within your app.

Thanks!

Any questions?

You can find me at:

vc@valentino.io
https://github.com/vc1492a