

HW 7: Flask

Objectives

- Be able to install Flask and use Flask to create basic web applications
- Be able to use templates to integrate python and HTML code to create dynamic web pages

Supporting Materials

Starter code:

- NONE - no more training wheels.

Lectures:

- Week 13 - Flask

Other

- New York Times API
<https://developer.nytimes.com/>

Main Assignment

For this assignment, you will be creating a basic Flask application, that greets a user and tells them the top technology news stories for that day.

Part 1 - Setting Up Flask

1. Create a GitHub Repo.
2. Create a file called `top_headlines.py` which will contain the commands to set up the flask app that retrieves the top technology news stories today.
3. Create a python file called `"secrets.py"` for API Key within your repo directory
4. Create a directory called `"templates"` within your repo directory

After part 1, you should have 3 files within the local repo directory: (1) `"top_headlines.py"`, (2) `"secrets.py"` and `"README.md"`, and 1 directory called

“templates”.

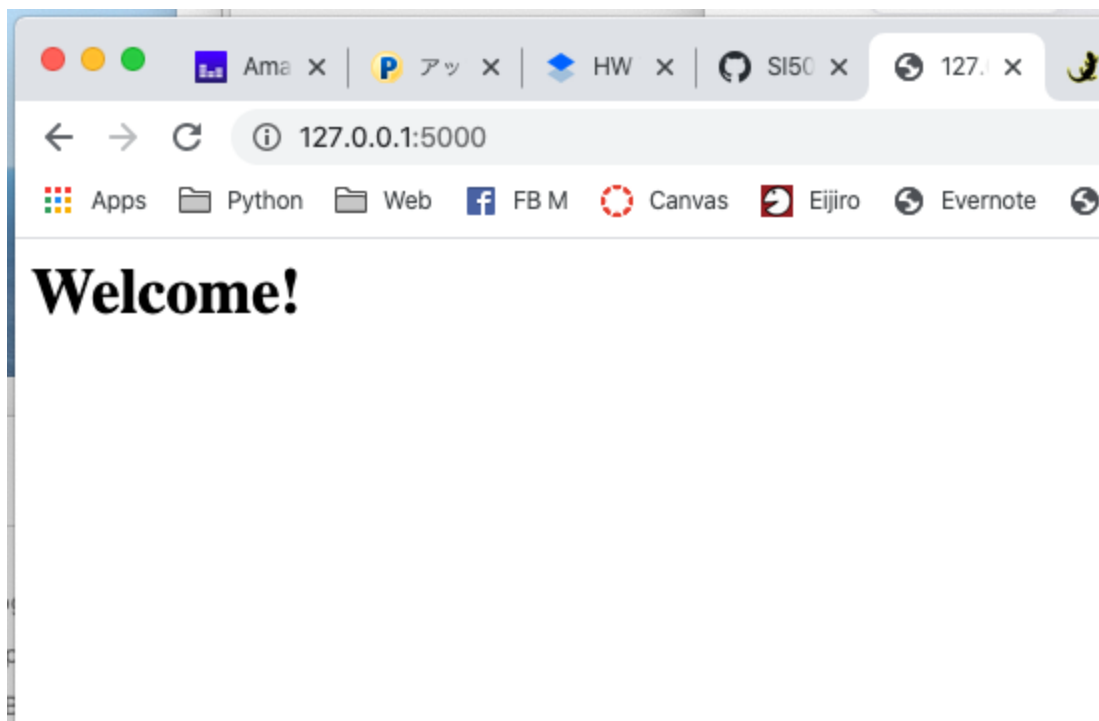
Part 2 - Adding Content

For both components in part 2, you can use as many or as few functions as you would like.

Note there is no starter code or tests for this assignment - the rubric is designed to be scored based on the behavior of your app. We’re taking the training wheels off here.

1. Default Home Page

If the user goes to the default homepage (<http://localhost:5000>), your output should look like the following (use `<h1>` tag to make the font larger):



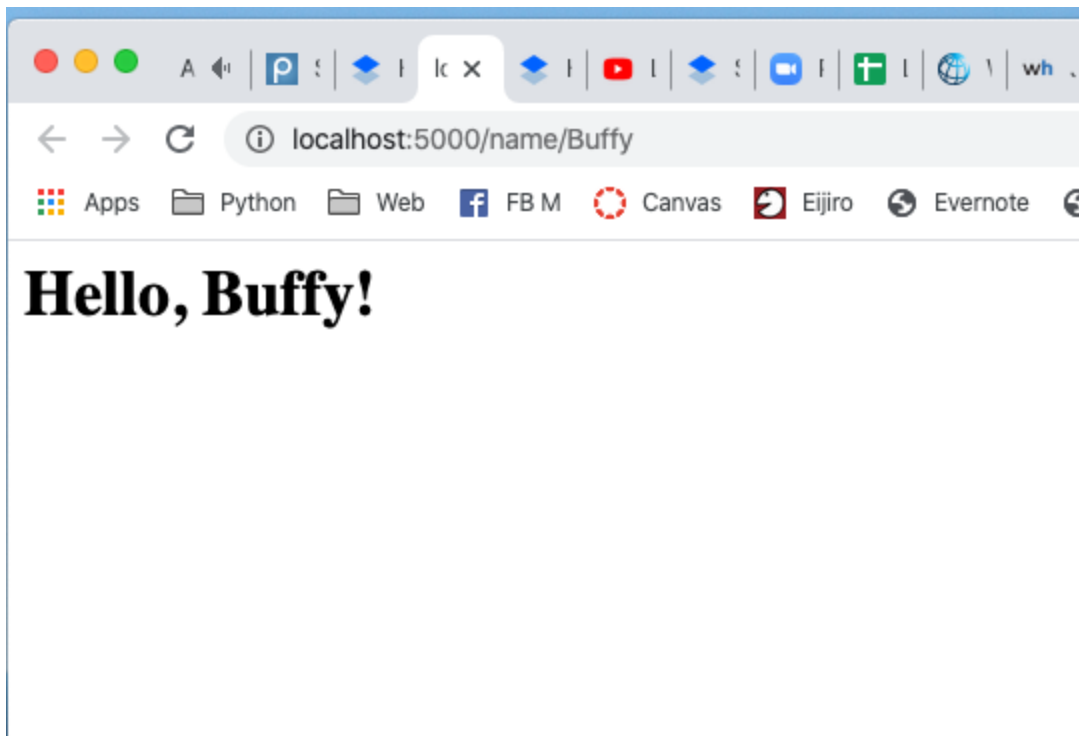
Sample Output 1

2. Get Users Name

Create a template called “name.html” under the directory “templates” with the following content:

```
<h1>Hello, {{name}}!</h1>
```

In your “top_headlines.py” file, utilize the template to incorporate an individual's name. For example, <http://localhost:5000/name/Bufy> should return the following:



Sample Output 2

3. Query New York Times API

In your “top_headlines.py” file, write code to retrieve the top 5 articles from the New York Times from the **technology** section. Make sure to include your API key in a file called “secrets.py”.

You need to obtain an API key for **Top Stories API** on New York Times. The link to get the API key and the documentation can be found at <https://developer.nytimes.com/>.

After issuing the API key, the screen should be the following:

Developers Home APIs Covid-19 Data Get Started [siano@unich.edu](#)

SI 507

Overview

App Name: **SI 507**

Description: **SI 507 Unich assignment**

App ID: [REDACTED]

API Keys

Key	Secret	Status	Created	Expires
[REDACTED]	Show	Active	2020-04-07	never

ADD KEY

Public APIs

APIs *
Authenticate at least once API for this app.

- Movie Reviews API: Search for movie reviews.
- Most Popular API: Popular articles on NYTimes.com.
- Times NY API: Real-time feed of NYT article publishes.
- Article Search API: Search for New York Times articles.
- NYT RSS section feeds: NYT RSS section feeds.
- Community API: Get user comments. (BETA)
- Best Sellers API: Get NYT Best Sellers Lists and lookup book reviews.
- Semantic API: Get semantic terms (people, places, organizations, and locations).
- Times Vocab API: NYT controlled vocabulary.
- Top Stories API: Get articles currently on a section front or the home page.
- Article API: Get all NYT article metadata for a given month.
- Geo API: Geographic linked data. (DEPRECATED)

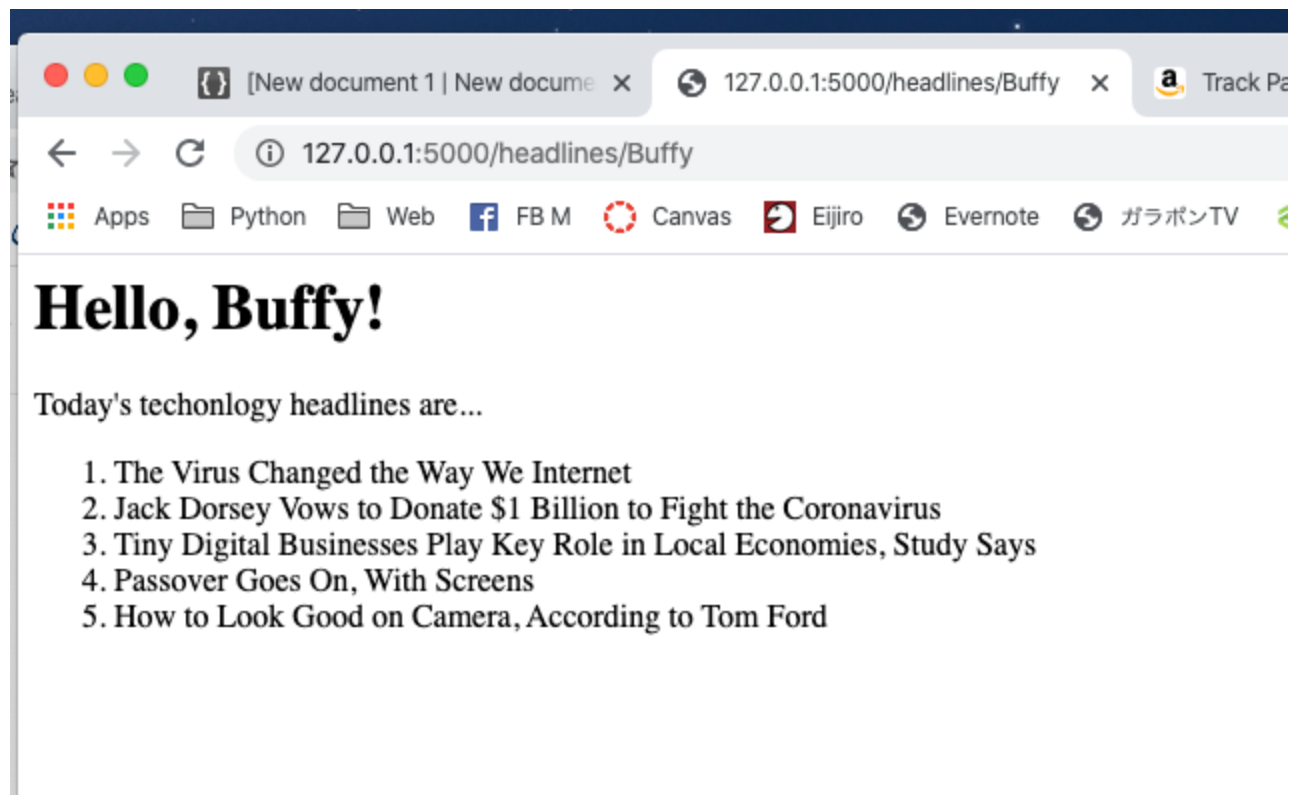
Check "Top Stories API"

"secrets.py" will contain a variable named **api_key**.

Format the top 5 articles with the corresponding titles as an ordered list.

Remember: Do NOT share your API key on Github. (use ".gitignore")

For example, if you were to run "top_headlines.py" on April 8th and go to the URL <http://localhost:5000/headlines/Bufy>, your final output of part 2 should look like this:



Sample Output 3

The text displayed should have different header tags (i.e. `<h1>`, `<h2>`). Experiment with the different header tags, until your webpage looks like the example above.

What to Submit

1. Push up all your code and files that you made to the Github Classroom. Make sure your files are organized as described above. Do NOT submit your "secrets.py" file.
2. Submit a GitHub URL on Canvas.

Rubric

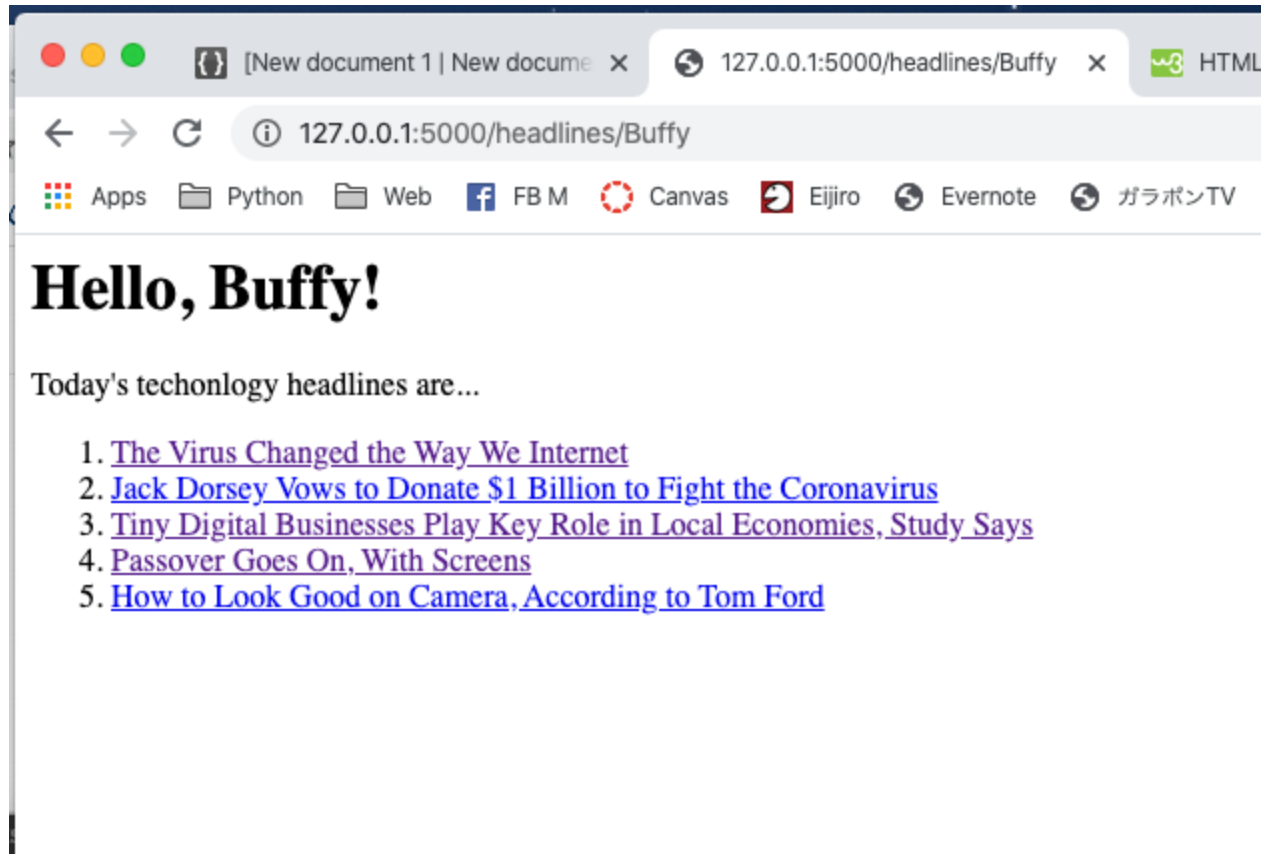
Req	Description	Category	Point Value
1	Part 2: When we run the "top_headlines.py", Flask environment becomes active.	Behaviour	10

	<p>i.e. Terminal should show something like below.</p> <pre>Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)</pre>		
2	Part 2-1: http://localhost:5000 doesn't show error.	Behaviour	10
3	Part 2-1: http://localhost:5000 shows the text "Welcome!". (See the sample output 1.)	Behaviour	10
4	Part 2-2: http://localhost:5000/name/Buffy doesn't show error.	Behaviour	10
5	Part 2-2: http://localhost:5000/name/Buffy shows the text "Hello Buffy!" (See the sample output 2.)	Behaviour	10
6	Part 2-2: When we replace "Buffy" in the URL with another name (e.g., "Biff"), the output changes correctly.	Behaviour	10
7	Part 2-3: http://localhost:5000/headlines/Buffy doesn't show error.	Behaviour	10
8	Part 2-3: http://localhost:5000/headlines/Buffy shows five lines of headlines.	Behaviour	10
9	Part 2-3: http://localhost:5000/headlines/Buffy shows NYT headlines of "Technology" category.	Behaviour	10
10	Part 2-3: http://localhost:5000/headlines/Buffy shows with a correct font tag. (See the sample output 3.)	Behaviour	10
	Total		100

Extra Credit 1 - Add links to your headlines

Change your program so that the user can see the headlines with links to the actual story on NYT. This functionality should be accessible at the route `/links/<name>`.

For example if a user goes to <http://localhost:5000/links/Bufy> they should see:



Sample Output 4

(and the links work).

Rubrics

Req	Description	Category	Point Value
11	http://localhost:5000/links/Bufy shows five headlines with links. (See the sample output 4.)	Behaviour	1
12	Each Link correctly opens the page. (The grader will try at least one of the links.)	Behaviour	1

	Total		2
--	-------	--	---

Extra Credit 2 - Add thumbnails and make a table

Change your program so that the user can see the headlines with links and thumbnails as a table. If you're unfamiliar with HTML tables you can learn how they work via this [HTML Tables tutorial](#). This functionality should be accessible at the route /images/<name>. For example if a user goes to <http://localhost:5000/images/Bufy> they should see:

localhost:5000/images/Bufy

localhost:5000/images/Bufy

Apps

Python

Web

FB M

Canvas

Eijiro

Evernote

ガラポンTV

Room

MBus

Hello, Buffy!

Today's techonlogy headlines are...

Title	Thumbnail
The Virus Changed the Way We Internet	<p>Change in average daily traffic from Jan. 21-March 24</p>
Jack Dorsey Vows to Donate \$1 Billion to Fight the Coronavirus	
Tiny Digital Businesses Play Key Role in Local Economies, Study Says	
Passover Goes On, With Screens	
Uber and Lyft Drivers Face Hurdles to Stimulus Bill Benefits	

Sample Output 5

Rubrics

--	--	--	--

Req	Description	Category	Point Value
13	http://localhost:5000/images/Buffy shows five headlines with thumbnails. (See the sample output 5.)	Behaviour	1
14	http://localhost:5000/images/Buffy shows information using a table.	Behaviour	1
	Total		2