http://bit.ly/2FYAerq

Flask

Josh Archibald - January 17, 2020
jarchibald@college.harvard.edu

Why Flask?

Walking

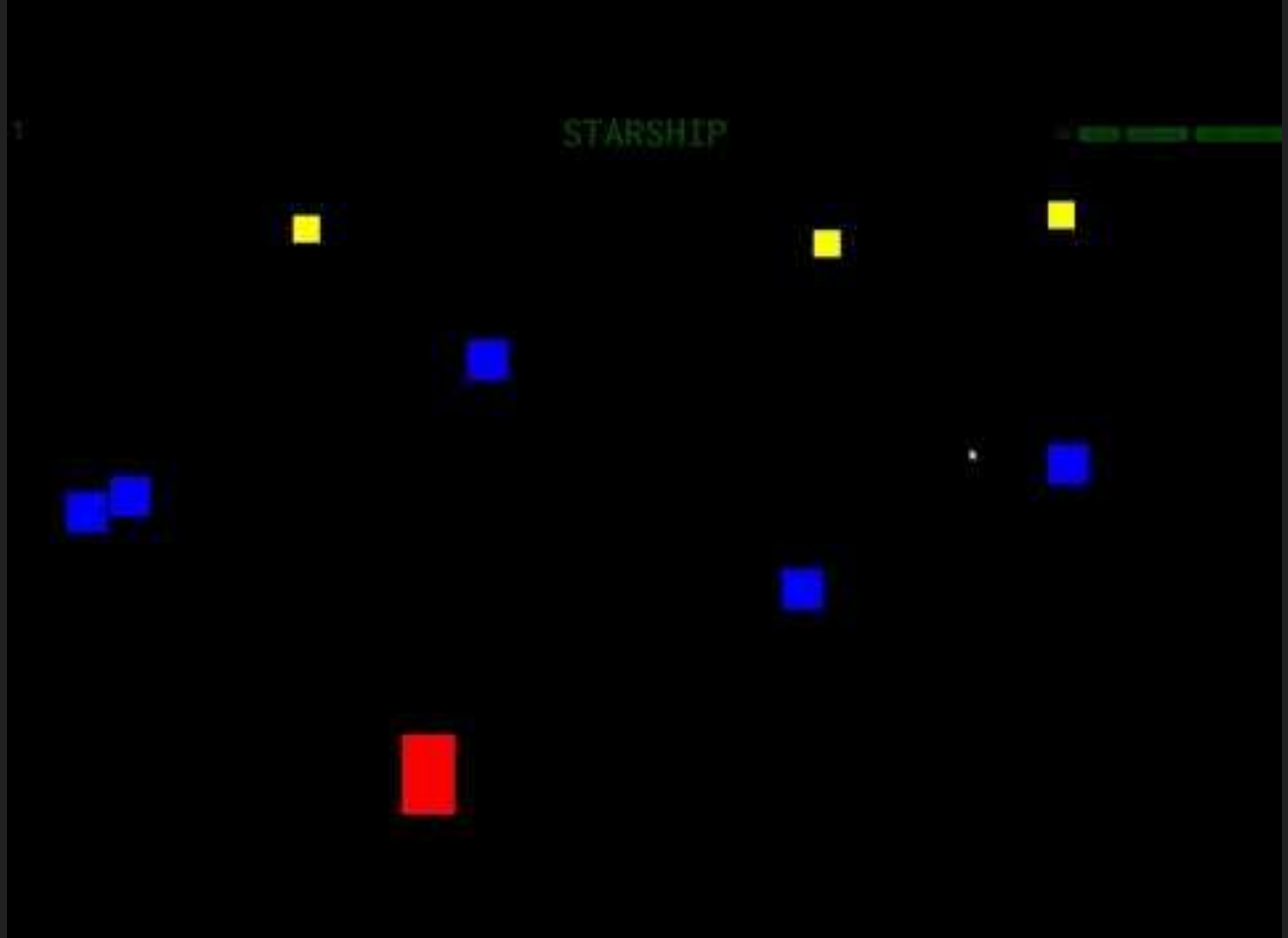
A guide to places near Harvard

The Charles River

Belltowers, bridges, boathouses, and...Mather? *Yuck!*

In all seriousness, and in the name of preventing a disaster on Housing Day, seeing Mather's cold concrete façade from Weeks Bridge is my favorite part of walking the Charles. I love Mather. I love Mather so much that it would only be just to *not* place me in Mather. Whatever! One can't have everything one wants.

Weeks Bridge during the day.


STARSHIP

Client-side

Server-side

Client

- User interface
  - Structure/content (HTML)
  - Design (CSS)
  - Interaction (JS)
- All done by the user's browser

Server

- Generally, anything to do with stored information
  - User registration/login
  - Storing info in a database
  - Processing orders
- Done on a server

Flask

- Simple microframework for server-side programming
- Uses Python
- Can work with a SQL database (and many others)
- Dynamic HTML generation using Jinja templating language
- Used in industry (https://stackshare.io/flask)
  - Netflix
  - Lyft
  - Zillow
- https://flask.palletsprojects.com/en/1.1.x/

MVC

M - Model - data-related logic (SQL)

V - View - the user interface presented (HTML/CSS/JS)

C - Controller - the interface between the model and view (Python)

Quick Python review

- Functions
- Modules
- Dictionaries

Functions

In functions.py,

- Write a function, square(*n*), that returns the square of *n*. For example, square(3) should return 9.
- Write a function, main(), that prints the squares of the numbers 0-9.

Modules

In modules.py,

- Import the square function from functions.py.
- Print out the value of 10 squared, using the square function you just imported.

Dictionaries

In dictionary.py,

- Create a new dictionary mapping first names to ages. For our purposes, let's say that Renee is 19 and Fernando is 29.
- Turns out that our data was wrong for Fernando -- he's actually 23. Write a second line of code that will just change Fernando's age to the correct number.
- This weekend is Renee's birthday. Write a third line of code to increment Renee's age by one.
- Finally, print the ages dictionary to see if your code works properly.

Routes

Routes

“bind a function to a URL”

cs50.harvard.edu/hls

cs50.harvard.edu/hls

cs50.harvard.edu/

cs50.harvard.edu/

Example: Hello, World!

Example: Different routes

/

Hello world!

/hello

Hello Josh!

Routes with variables

```
@app.route("/hello/<string:name>")
```

```
@app.route("/hello/<string:name>")
```

```
@app.route("/hello/<string:name>")
```

```
@app.route("/hello/<string:name>")
```

Example: Hello, __

/

Hello world!

/hello/Josh

Hello Josh!

/hello/Fernie

Hello Fernie!

/hello/Vibha

Hello Vibha!

Templates

Intro to Jinja templating: extending a layout

layout.html

```
{% block content %} {% endblock %}
```

index.html

```
{% extends "layout.html" %}
{% block content %}
<h1>Something goes here!</h1>
{% endblock %}
```

Example: Templates, Inheritance

Static HTML

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Hello world</title>
  </head>
  <body>
    <h1>Hi there!</h1>
  </body>
</html>
```

Static HTML

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Hello world</title>
  </head>
  <body>
    <h1>Hi there!</h1>
  </body>
</html>
```

Intro to Jinja templating: logic in HTML!?

```
{{ variable_name }}


{% if variable_name == "test" %}
…
{% else %}
...
{% endif %}
```

https://jinja.palletsprojects.com/en/2.10.x/

Example: Using variables in templates

/

<h1>Hello world!</h1>

/hello/Josh

<h1>Hello Josh!</h1>

/hello/Fernie

<h1>Hello Fernie!</h1>

POST vs. GET

GET request example

https://www.google.com/search?q=Budapest

GET request example

https://www.google.com/search?q=Budapest

https://bad.security.net/login?username=jsmith&password=dolphin

Example: Form submissions

/

Name: Josh SUBMIT  <h1>Hello Josh!</h1>

Example: User registration and login

Sessions

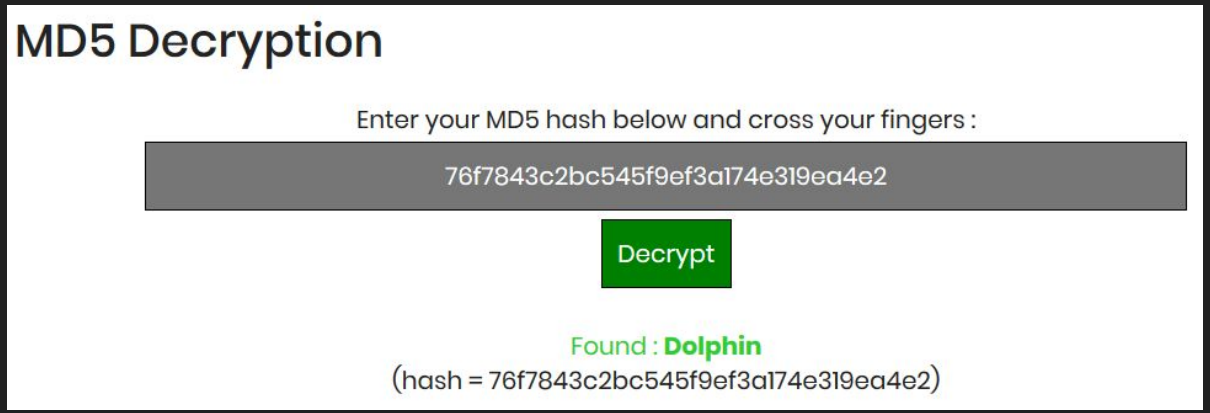
Hashes

MD5 (don’t use!)

- password → 5f4dcc3b5aa765d61d8327deb882cf99
- Elena Kagan → bfb23925528b7fa63e77aeed31c3532e
- dolphin → 36cdf8b887a5cffc78dcd5c08991b993
- Dolphin → 76f7843c2bc545f9ef3a174e319ea4e2
- ...

MD5 (don’t use!)

- password → 5f4dcc3b5aa765d61d8327deb882cf99
- Elena Kagan → bfb23925528b7fa63e77aeed31c3532e
- dolphin → 36cdf8b887a5cffc78dcd5c08991b993
- Dolphin → 76f7843c2bc545f9ef3a174e319ea4e2
- ...



A good hash function should

- Use only the data being hashed
- Use all of the data being hashed
- Be deterministic (same result every time given same input; no randomness!)
- Uniformly distribute data
- Generate very different hash codes for very similar data.

Principle of password security: compare hashes

- Don’t store the passwords in plaintext -- if the database is hacked, passwords are in the open
- Use advanced hash functions (not MD5) to store the irreversible hashes instead of passwords
- Compare the hashes

In the context of Flask...

```
from werkzeug.security import check_password_hash, generate_password_hash
```

```
check_password_hash(stored_hash, password_plaintext)
```

```
generate_password_hash(password_plaintext)
```

Example: User registration and login

Flask

Josh Archibald - January 17, 2020
jarchibald@college.harvard.edu