

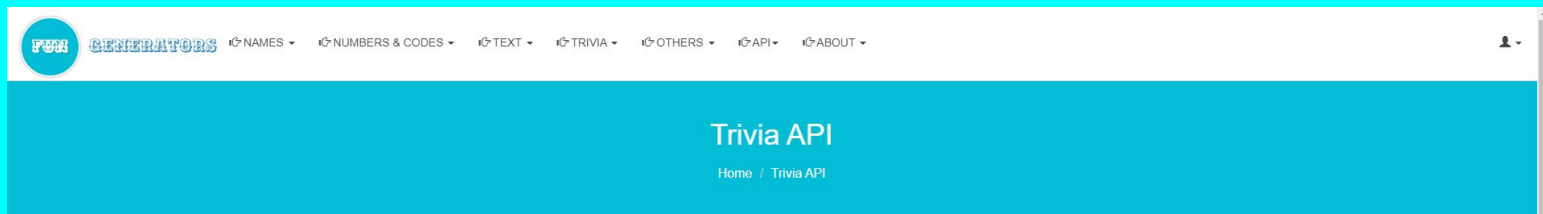
Python - Flask App Trivia Game

Kristian Reyes

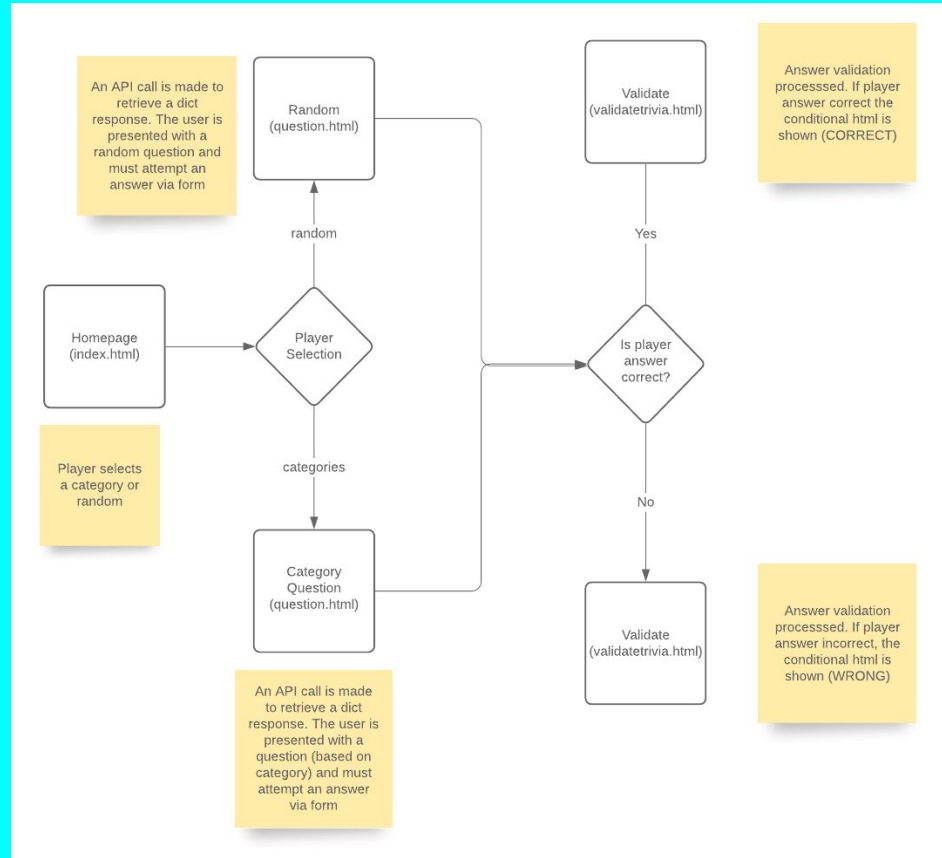
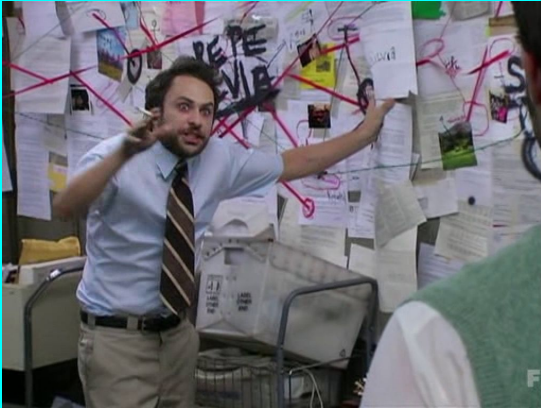
<https://docs.google.com/presentation/d/1E3bCMP18o7M5UwcwIZl6GybC3At6mrsWObF5JpWgBsk/edit?usp=sharing>

Trivia Game Overview

- A Flask-based Trivia Game
- Player selects a category type, is presented with a question
- Player submits answer via form and validation is completed
- Uses the Fun Generators API to retrieve information
 - <https://fungenerators.com/api/trivia/#apidoc>



Trivia Game Flow



Trivia Game Homepage

- Player selects a category type or random. HTML form sends a POST Method based on choice

Welcome to Trivia Challenge!

Player one, select a category or pick random

```
</div>
<form method="POST" action="http://127.0.0.1:5000/categories/">
  <input type="submit" name="category" value="Fashion" >
</form>
<br>
<form method="POST" action="http://127.0.0.1:5000/categories/">
  <input type="submit" name="category" value="Beer" >
</form>
<br>
<form method="POST" action="http://127.0.0.1:5000/random/">
  <input type="submit" name="category" value="Random" >
</form>
</body>
```

Trivia Game Question

- An API call is made based on choice. The appropriate Trivia class method is called. The new route (random or categories) displays the same question.html with dynamic templating.

Your question is:

What does the SI prefix 'pico' mean ?

Type in your answer.

You only have one attempt!


```
20 # route if player selects random category. API call for random is initiated. The question and answer submit form is rendered in the html
21 @app.route('/random/', methods=["GET", "POST"])
22 def random():
23     q1 = Trivia()
24     response_data = q1.randomtrivia()
25
26     q1.trivia_question = response_data['contents'][0]['question']
27     q1.trivia_correct_answer = response_data['contents'][0]['answer'][0]
28
29     answer = request.args.get("query")
30
31     return render_template('question.html', trivia_question=q1.trivia_question, trivia_correct_answer=q1.trivia_correct_answer, query=answer)
32
33 # route if player selects a category other than random. API call for category is initiated. The question and answer submit form is rendered in the html
34
35 @app.route('/categories/', methods=["GET", "POST"])
36 def category_q1():
37     q1 = Trivia()
38     category = request.form.get("category")
39     # print(category)
40
41     response_data = q1.categorytrivia(category)
42     # print(response_data)
43     # print(q1.trivia_question)
44     # print(q1.trivia_correct_answer)
45
46     q1.trivia_question = response_data[0]
47     q1.trivia_correct_answer = response_data[1]
48
49     answer = request.args.get("query")
50
51     return render_template('question.html', trivia_question=q1.trivia_question, trivia_correct_answer=q1.trivia_correct_answer, category=category, query=answer)
```

Trivia Game API Calls

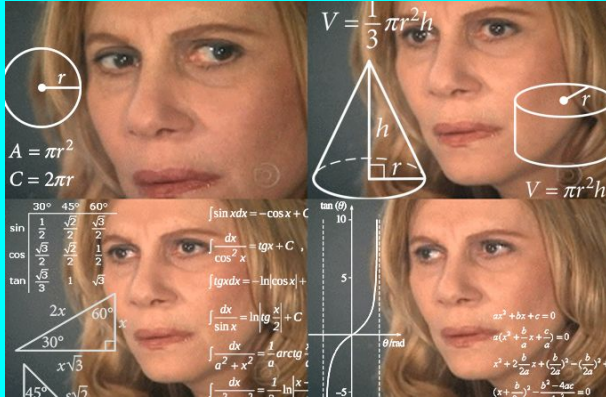
- The Trivia class contains methods for API calls and processing input

```
15 def categorytrivia(self,category):
16
17     response = requests.get('https://api.fungenerators.com/trivia/random?category='+ str(category),
18                             headers={'X-FunGenerators-API-Secret': 'vxMlxI9lADxza7qB8tEDowEf'})
19
20     response_data = response.json()
21
22
23     trivia_question = response_data['contents'][0]['question']
24     trivia_correct_answer = response_data['contents'][0]['answer'][0]
25
26     # return response_data
27     return trivia_question, trivia_correct_answer
28     # print("Your question is: " + trivia_question + "?")
29     # print("The correct answer is: " + trivia_correct_answer)
30
31 # function to call API based on random category
32
33 def randomtrivia(self):
34
35     response = requests.get(
36         'https://api.fungenerators.com/trivia/random',
37         headers={'X-FunGenerators-API-Secret': 'vxMlxI9lADxza7qB8tEDowEf'})
38
39     response_data = response.json()
40
41     # trivia_question = response_data['contents'][0]['question']
42     # trivia_correct_answer = response_data['contents'][0]['answer'][0]
43
44     return response_data
45
46     # print("Your question is: " + trivia_question + "?")
47     # print("The correct answer is: " + trivia_correct_answer)
48
```

API key passed in headers

Trivia Game Validation

- A validation is made to check if the player-submitted answer is correct



```
#function to validate player answer against correct answer.
```

```
def validate_trivia(answer, trivia_correct_answer):
```

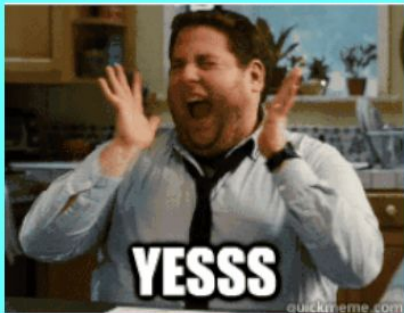
```
    if answer == trivia_correct_answer:
        answer = True
        return answer
```

```
    elif answer != trivia_correct_answer:
        answer = False
        return answer
```

Trivia Game Final

- Based on answer validation, the user is presented with conditional html

Your answer is CORRECT!



```
{% if query == True %}
    <h2> Your answer is CORRECT!</h2>
    
</body>
{% endif %}
```


Trivia Game Final

- Based on answer validation, the user is presented with conditional html

Your answer is WRONG!



```
1 <!doctype html>
2 <html lang="en">
3
4 <head>
5   <meta charset="utf-8">
6   <title>You are...</title>
7   <link rel="stylesheet" href="{{ url_for('static', filename='style.css')}}">
8 </head>
9
10 <body>|
11 {% if query == False %}
12
13   <h2> Your answer is WRONG!</h2>
14   
17   {% endif %}
```

