

FPROplay & FPROtest

Ricardo Cruz • Researcher @ FEUP & INESC TEC

FPRO Web - Chromium

FPRO Web

Inseguro | fpro.fe.up.pt/play/first_programs/hello_world

FPRO Play

Hello World ★

Print the string "Hello world!"

```
print("Hello world!")
```

Nothing has been submitted yet.

Let us know if you find problems: fpro@fe.up.pt

FPRO 2018/2019

FPRO Web - Chromium

FPRO Web

Inseguro | fpro.fe.up.pt/test/RE09

1. Hash Collisions

	Input	Expected Output	Student Output	Mark
1	[]	{}	{}	100%
2	[1, 289, 9493, 123, 5, 1241, 7, 6, 70, 124, 13, 17, 22]	{1: 2, 3: 1, 6: 2, 5: 1, 0: 2, 7: 3, 4: 2}	{1: 2, 3: 1, 6: 2, 5: 1, 0: 2, 7: 3, 4: 2}	100%
3	[21, 43, 56, 72, 131, 211, 311]	{3: 2, 7: 1, 1: 1, 5: 2, 4: 1}	{3: 2, 7: 1, 1: 1, 5: 2, 4: 1}	100%
4	[90, 120, 80, 10, 60, 80, 90, 900, 45, 18, 81]	{1: 7, 3: 1, 0: 2, 6: 1}	{1: 7, 3: 1, 0: 2, 6: 1}	100%

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Sat Nov 24 16:49:32 2018
4
5 @author: unknown
6 """
7
```

Let us know if you find problems: fpro@fe.up.pt

FPRO 2018/2019

Agenda

In this lecture, we will present in broad terms how FPROtest and FPROplay work.

- ▶ flask package
 - ▶ Python running in the server (not the client)
 - ▶ What are decorators
 - ▶ How to use Flask
- ▶ Running Python code as string
- ▶ Retrieval and manipulation of remote strings (scraping).

1

2

3

4

5

6

7

8

9

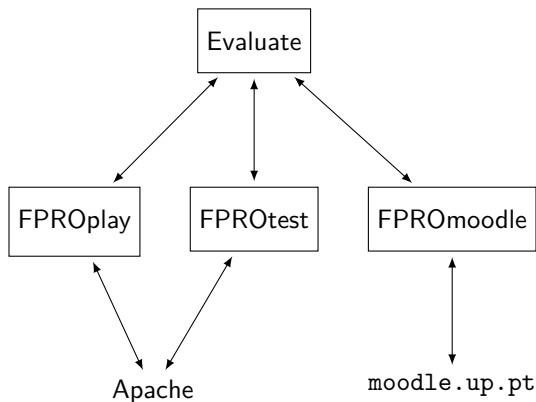
10

11

12

13

FPROweb structure



1

2

3

4

5

6

7

8

9

10

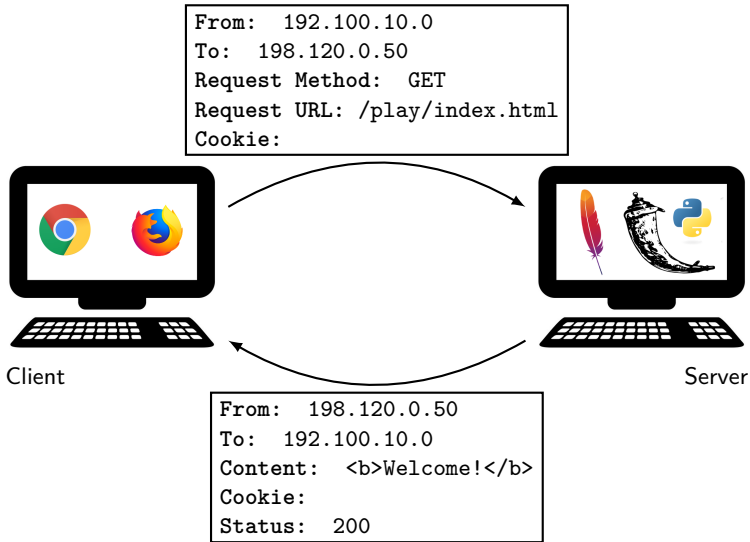
11

12

13

Flask web development

How the Internet Works



1

2

3

4

5

6

7

8

9

10

11

12

13

Flask

```
1 from flask import Flask
2 app = Flask(__name__)
3
4 @app.route("/play/index.html")
5 def f():
6     return "<b>Welcome!</b>"
```

1

2

3

4

5

6

7

8

9

10

11

12

13

Demonstration: Let's make a website where the user inputs a **function**, and the output is its **derivative**.¹

$$f(x) = x^2 + 3x + 5 \quad \Rightarrow \quad \frac{df}{dx} = 2x + 3$$

¹<https://www.dropbox.com/s/j1jp2x40puguzht/derivada.py?dl=0>

Decorator

```
1 from flask import Flask
2 app = Flask(__name__)
3
4 @app.route("/play/index.html")
5 def f():
6     return "<b>Welcome!</b>"
```

What is the @ symbol?

Demonstration: Example of a decorator: convert a function which requires *radians* so that it now requires *degrees*.²

²<https://www.dropbox.com/s/ciwc73zd69sa1pe/decorator.py?dl=0>

1

2

3

4

5

6

7

8

9

10

11

12

13

Evaluating Code

Evaluation Module

Three modes:

- ▶ function
- ▶ input()
- ▶ replace

A sub-process runs your code:

Run code once:

```
1 prog = {}  
2 exec(code, prog)
```

Call function:

```
1 student_output = prog[function_name](*inputs)
```

eval	Run a single expression and return its value
exec	Run multiple statements

They both have support for a `globals` and `locals` dictionary.

1

2

3

4

5

6

7

8

9

10

11

12

13

Flask

Demonstration: A web application that uses `exec()` to evaluate a function developed by the user.

1

2

3

4

5

6

7

8

9

10

11

12

13

Evaluation Module

Check if correct:

```
1 correct = int(expected_output == student_output)
```

Otherwise, check if there are minor failures:

```
1 minor_fail = 0
2 # wrong case
3 minor_fail += int(student_output.lower() ==
4                    expected_output.lower())
5 # wrong type
6 if type(student_output) != type(expected_output):
7     try: minor_fail += int(type(expected_output)(
8         student_output) == expected_output)
9     except: pass
```

1

2

3

4

5

6

7

8

9

10

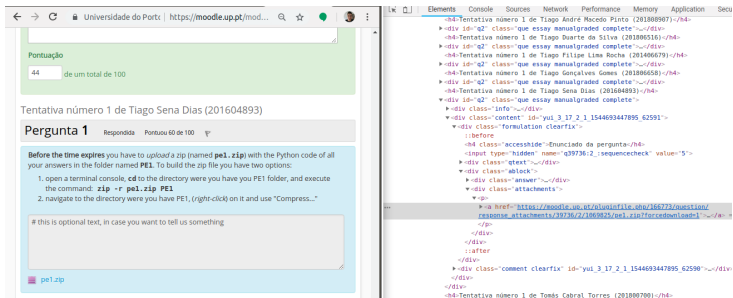
11

12

13

Web Scraping

Moodle Module



```
1 import requests
2 text = request.get(url)
```

```
<h4>Tentativa número 2 de xxxxxxxxxxxxxxxx (201806554)</h4>
<div id="q2" class="que essay manualgraded complete">
<b>Before the time expires</b> you have to <i>upload</i> a zip (named
<ol>
<li>open a terminal console, <b><tt>cd</tt></b> to the directory were
<li>navigate to the directory where you have PE1, (<i>right-click</i>))
</ol></div><div class="ablock"><div class="answer">
<p><a href="https://moodle.up.pt/pluginfile.php/166773/question/respon
```

1

2

3

4

5

6

7

8

9

10

11

12

13

Moodle Module

Scraping:

```
1 from lxml import html
2 tree = html.fromstring(response.text)
3 names = tree.xpath('//form/div/h4/text()')
4 answers = tree.xpath("//form/div/div[starts-with(
    @id, 'q')]")
```

In the end,

```
1 requests.post(post_url, data=payload)
```

1

2

3

4

5

6

7

8

9

10

11

12

13

Flask

Demonstration: A web application that goes to OLX.pt and shows pictures of the first cars using a certain keyword.

1

2

3

4

5

6

7

8

9

10

11

12

13

Conclusion

Conclusion

Conclusion related to learning programming Python

1

2

3

4

5

6

7

8

9

10

11

12

13

