

DATA SCIENCE FUNDAMENTALS

LESSON 6

Hay Kranen

Wednesday October 10th, 2018



TODAY'S PROGRAMME

How are we doing?

Movie database

Recap

Break

HTTP API's

HTTP exercise

HTTP + JSON

Lunch break

TODAY'S PROGRAMME

How are we doing?

Movie database

Recap

Break

HTTP API's

HTTP exercise

HTTP + JSON

Lunch break

Lesson 5

- Dicts
- Dict indexing
- Replacing values in a dict
- Dict methods: keys(), values(), get()
- Lists with dicts with lists with dicts, wow
- Looping over dicts using items()



How are you doing?

It's important for Jonas and me to get a grip on how you are doing. Please enter these three questions so we get a global view on how you're doing.

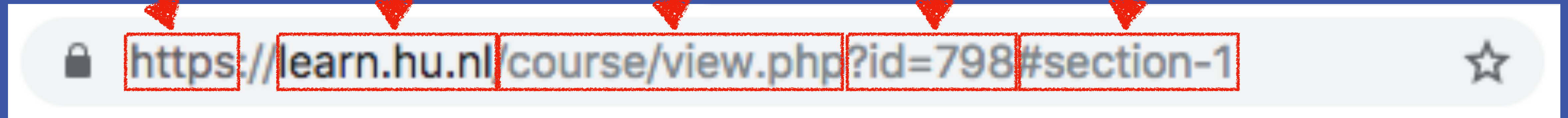
MOVIE DATABASE

RECAP

дНТТР

Who has ever used a stateless
REST API?

Protocol (sub)domain path query fragment

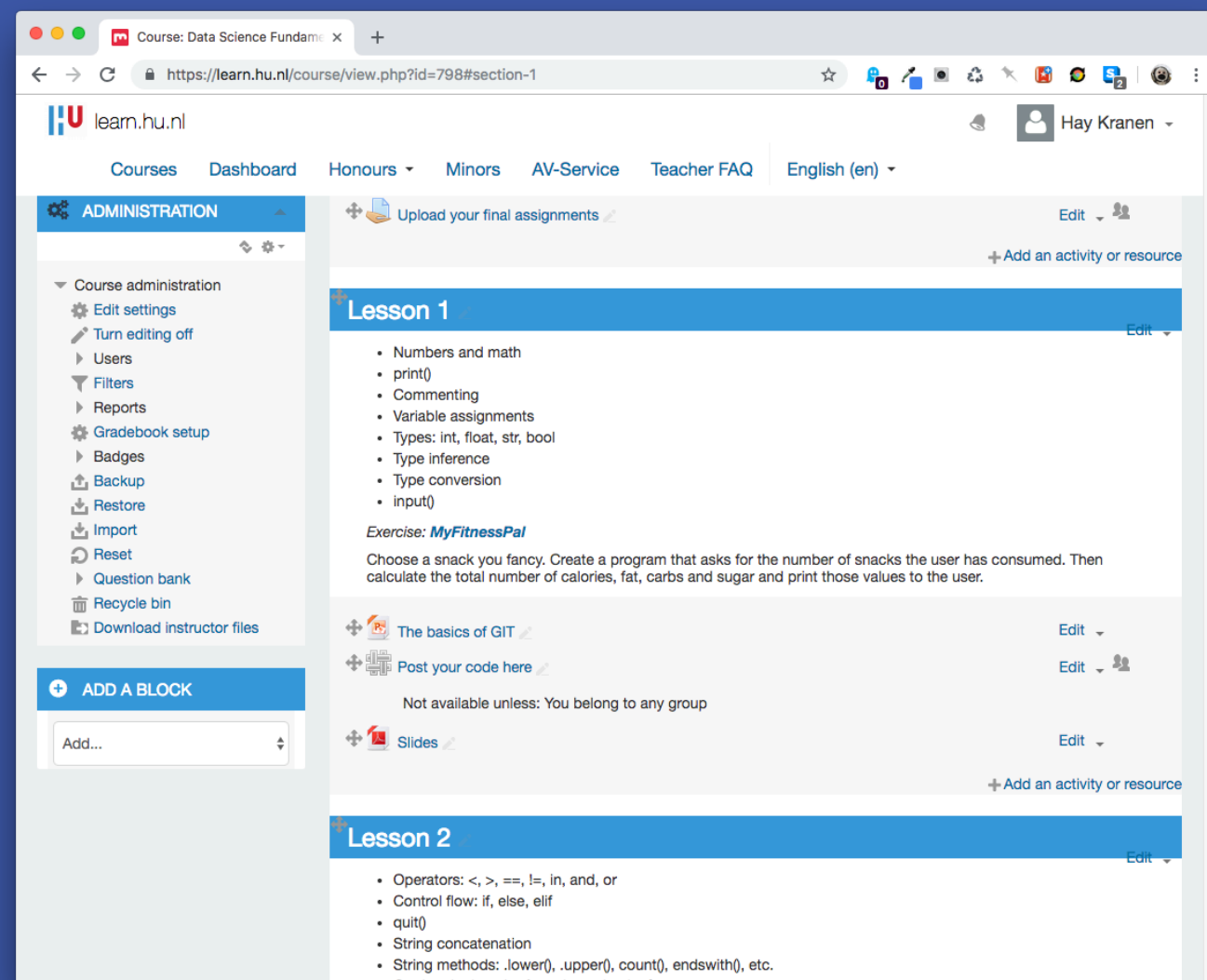




https://learn.hu.nl/course/view.php?id=798#section-1



HTML



```
1 <!DOCTYPE html>
2 <html dir="ltr" lang="en" xml:lang="en">
3
4 <head>
5   <title>Course: Data Science Fundamentals</title>
6   <link rel="shortcut icon" href="https://learn.hu.nl/theme/image.php/aardvark/theme/1538122251/favicon" />
7   <link rel="apple-touch-icon-precomposed" href="https://learn.hu.nl/theme/image.php/aardvark/theme/1538122251/apple-touch-icon" />
8   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
9   <meta name="keywords" content="moodle, Course: Data Science Fundamentals" />
10  <link rel="stylesheet" type="text/css" href="https://learn.hu.nl/theme/yui_combo.php?rollup/3.17.2/yui-moodlesimple-min.css" /><script
11  id="firstthemesheet" type="text/css">/** Required in order to fix style inclusion problems in IE with YUI **</script><link
12  rel="stylesheet" type="text/css" href="https://learn.hu.nl/theme/styles.php/aardvark/1538122251/all" />
13  <script type="text/javascript">
14    //
15    var M = {}; M.yui = {};
16    M.cfg =
17    {
18      "wwwroot": "https://learn.hu.nl",
19      "sesskey": "cHQ2Y9TUMl",
20      "loadingicon": "https://learn.hu.nl/theme/image.php/aardvark/core/1538122251/i/loading_small",
21      "themerev": "1538122251",
22      "slasharguments": 1,
23      "theme": "aardvark",
24      "jsrev": "1538122251",
25      "admin": "admin",
26      "svgicons": true
27    };
28    var yui1ConfigFn = function(me) {
29      if (/^skin$/ .test(me.name)) {
30        me.type = 'css';
31        me.path = me.path.replace(/\.js$/, '.css');
32        me.path = me.path.replace(/\/yui2-skin/, '/assets/skins/sam/yui2-skin');
33      }
34      var yui2ConfigFn = function(me) {
35        var parts = me.name.replace(/^moodle-/, '').split('-');
36        component = parts.shift();
37        module = parts[0];
38        min = 'min';
39        if (/^skin$/ .test(me.name)) {
40          parts.pop();
41          me.type = 'css';
42          min = '';
43        }
44        me.path = component + '/' + module + '/' + filename + min + '.' + me.type;
45        else {
46          me.path = component + '/' + component + '.' + me.type;
47        }
48      };
49      YUI_config =
50      {
51        "debug": false,
52        "base": "https://learn.hu.nl/lib/yuilib/3.17.2/",
53        "comboBase": "https://learn.hu.nl/theme/yui_combo.php?",
54        "combine": true,
55        "filter": null,
56        "insertBefore": "firstthemesheet",
57        "groups": {
58          "yui2": {
59            "base": "https://learn.hu.nl/lib/yuilib/2in3/2.9.0/build/",
60            "comboBase": "https://learn.hu.nl/theme/yui_combo.php?",
61            "combine": true,
62            "ext": false,
63            "root": "2in3/2.9.0/build/",
64            "patterns": {
65              "yui2": {
66                "group": "yui2",
67                "configFn": yui2ConfigFn
68              }
69            },
70            "name": "moodle",
71            "base": "https://learn.hu.nl/theme/yui_combo.php?",
72            "m": "1538122251/",
73            "combine": true,
74            "comboBase": "https://learn.hu.nl/theme/yui_combo.php?",
75            "ext": false,
76            "root": "m/1538122251/",
77            "patterns": {
78              "moodle": {
79                "group": "moodle",
80                "configFn": yui2ConfigFn
81              },
82              "moodle-core-notification-dialogue": {
83                "requires": [
84                  "moodle-core-notification-alert",
85                  "moodle-core-notification-confirm",
86                  "moodle-core-notification-exception",
87                  "moodle-core-notification-ajaxexception"
88                ]
89              },
90              "moodle-core-notification-alert": {
91                "requires": [
92                  "base",
93                  "node",
94                  "panel",
95                  "escape",
96                  "event-key",
97                  "dd-plugin",
98                  "moodle-core-widget-focusafterclose",
99                  "moodle-core-lockscroll"
100               ],
101               "moodle-core-notification-alert": {
102                 "requires": [
103                   "moodle-core-notification-dialogue"
104                 ]
105               },
106               "moodle-core-notification-confirm": {
107                 "requires": [
108                   "moodle-core-notification-dialogue"
109                 ]
110               },
111               "moodle-core-notification-ajaxexception": {
112                 "requires": [
113                   "moodle-core-</pre></div>
```

Client

GET
POST
PUT
DELETE

Request

METHOD URL

Headers (metadata)

Server

200 OK
301 MOVED
403 FORBIDDEN
404 NOT FOUND
500 SERVER ERROR

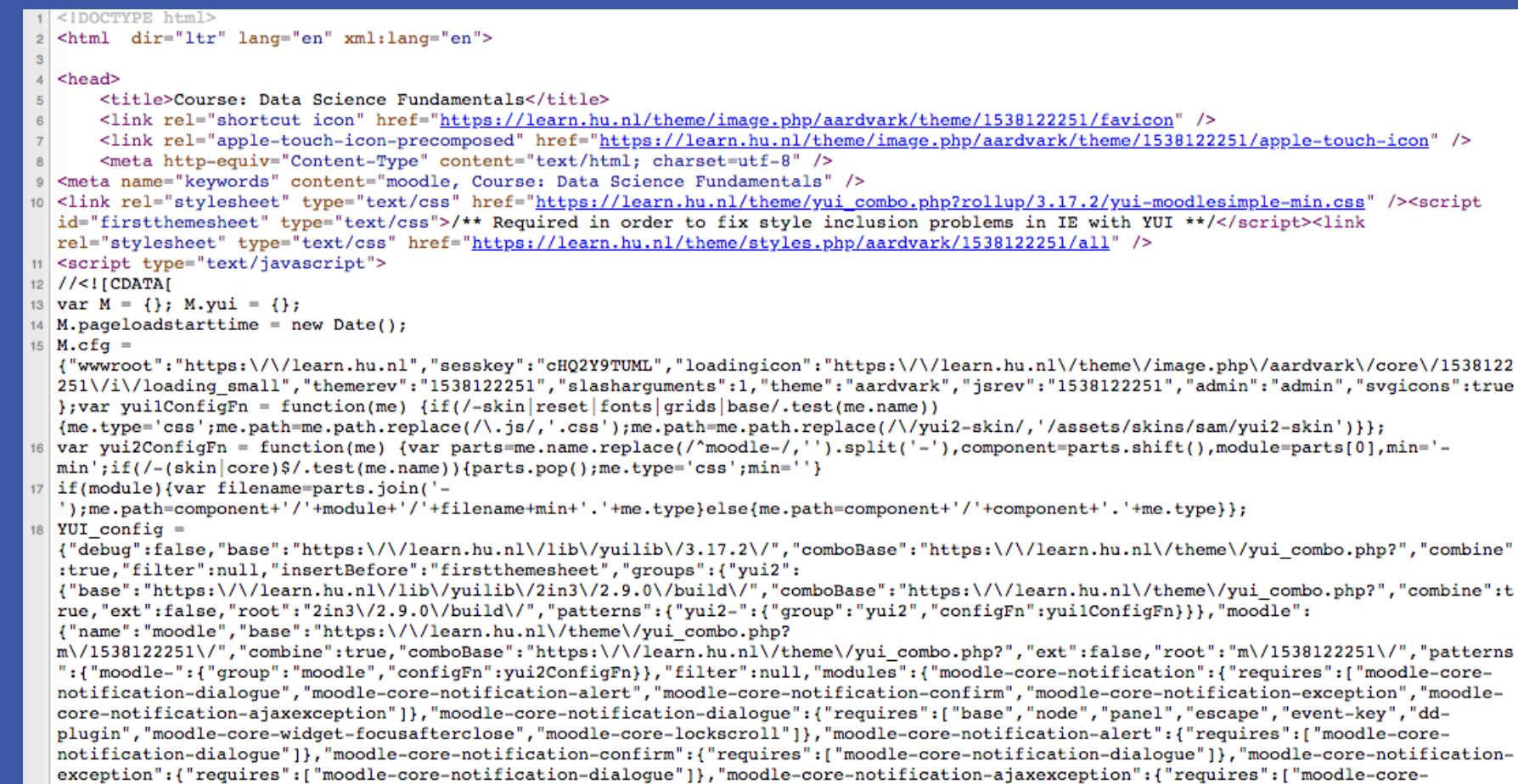
Response

Data

Headers (metadata)

Client

```
HTTP/1.1 200 OK
Content-Length: 139964
Content-Type: text/html;
charset=utf-8
Server: Microsoft-IIS/10.0
Set-Cookie:
ASP.NET_SessionId=yudsrvaerzqysn
eo3hpjbqbb; path=/; HttpOnly
Date: Tue, 09 Oct 2018 14:39:42
GMT
```



https://en.wikipedia.org/api/rest_v1/page/summary/Lasagne



JSON

```
hay@haybook: ~/htdocs/git/hu-dsf/exercises — -bash — 76x26
hay@haybook:~/htdocs/git/hu-dsf/exercises [master] $ python wikisum.py
Enter an article: Lasagne
Lasagne: dish in Italian cuisine

Lasagne is a type of wide, flat pasta, possibly one of the oldest types of pasta. Lasagne, or the singular lasagna, commonly refers to a culinary dish made with stacked layers of pasta alternated with sauces and ingredients such as meats, vegetables and cheese, and sometimes topped with melted grated cheese. Typically, the cooked pasta is assembled with the other ingredients and then baked in an oven. The resulting lasagne casserole is cut into single-serving square portions.
hay@haybook:~/htdocs/git/hu-dsf/exercises [master] $
```

```
{
  "type": "standard",
  "title": "Lasagne",
  "displaytitle": "Lasagne",
  "namespace": {
    "id": 0,
    "text": ""
  },
  "titles": {
    "canonical": "Lasagne",
    "normalized": "Lasagne",
    "display": "Lasagne"
  },
  "pageid": 140607,
  "thumbnail": {
    "source": "https://upload.wikimedia.org/wikipedia/commons/thumb/b/ba/Lasagne_-_stonesoup.jpg/312px-Lasagne_-_stonesoup.jpg",
    "width": 312,
    "height": 320
  },
  "originalimage": {
    "source": "https://upload.wikimedia.org/wikipedia/commons/b/ba/Lasagne_-_stonesoup.jpg",
    "width": 2000,
    "height": 2050
  },
  "lang": "en",
  "dir": "ltr",
  "revision": "861875462",
  "tid": "86b6b9ba-c762-11e8-a955-440379963805",
  "timestamp": "2018-09-30T17:42:19Z",
  "description": "dish in Italian cuisine",
  "content_urls": {
    "desktop": {
      "page": "https://en.wikipedia.org/wiki/Lasagne",
      "revisions": "https://en.wikipedia.org/wiki/Lasagne?action=history",
      "edit": "https://en.wikipedia.org/wiki/Lasagne?action=edit",
      "talk": "https://en.wikipedia.org/wiki/Talk:Lasagne"
    },
    "mobile": {
```


Why would you use an API?

Tips when using web API's

- * Use DevTools (or Firefox) to view JSON in your browser
- * Try writing API calls in the browser first
- * Try using `curl` on the command line



Requests
http for humans

URL checker

Create a program that asks the user for an URL, does a GET request for that URL and displays the useful information.

- * Import the **requests** library
- * Ask the user for an URL and **strip()** it
- * Check the **status_code**, if it is not **200**, display an error and exit the program
- * Loop over the **headers** and display the key and value of every header.

Extend your program

- * Display the first ten lines of the text of the page the user provided

Tips

- * Everything you need for this exercise is in the 'lesson 6' chapters of the **examples-3** Jupyter Notebook.
- * If you can't import the **requests** library you might need to install it using the **pip** command line utility
- * Use **requests.get()** to do a GET request
- * The **status_code** is an int
- * F-strings are really useful for this exercise
- * Remember the **.items()** method of dictionaries

HTTP + JSON

Assignment 3

Wikipedia extracts

Create a program that asks for the name of an article and uses the Wikipedia REST API to get the short description and extract of that article.

- * Import the **requests** and **json** libraries
- * Ask the user for an article and **strip** it and replaces the spaces with underscores (**_**)
- * Format the API endpoint with the article
- * Use **requests.get()** to get the data
- * Check if we got a 200 status code, otherwise abort the program
- * Display the title of the article, description and extract

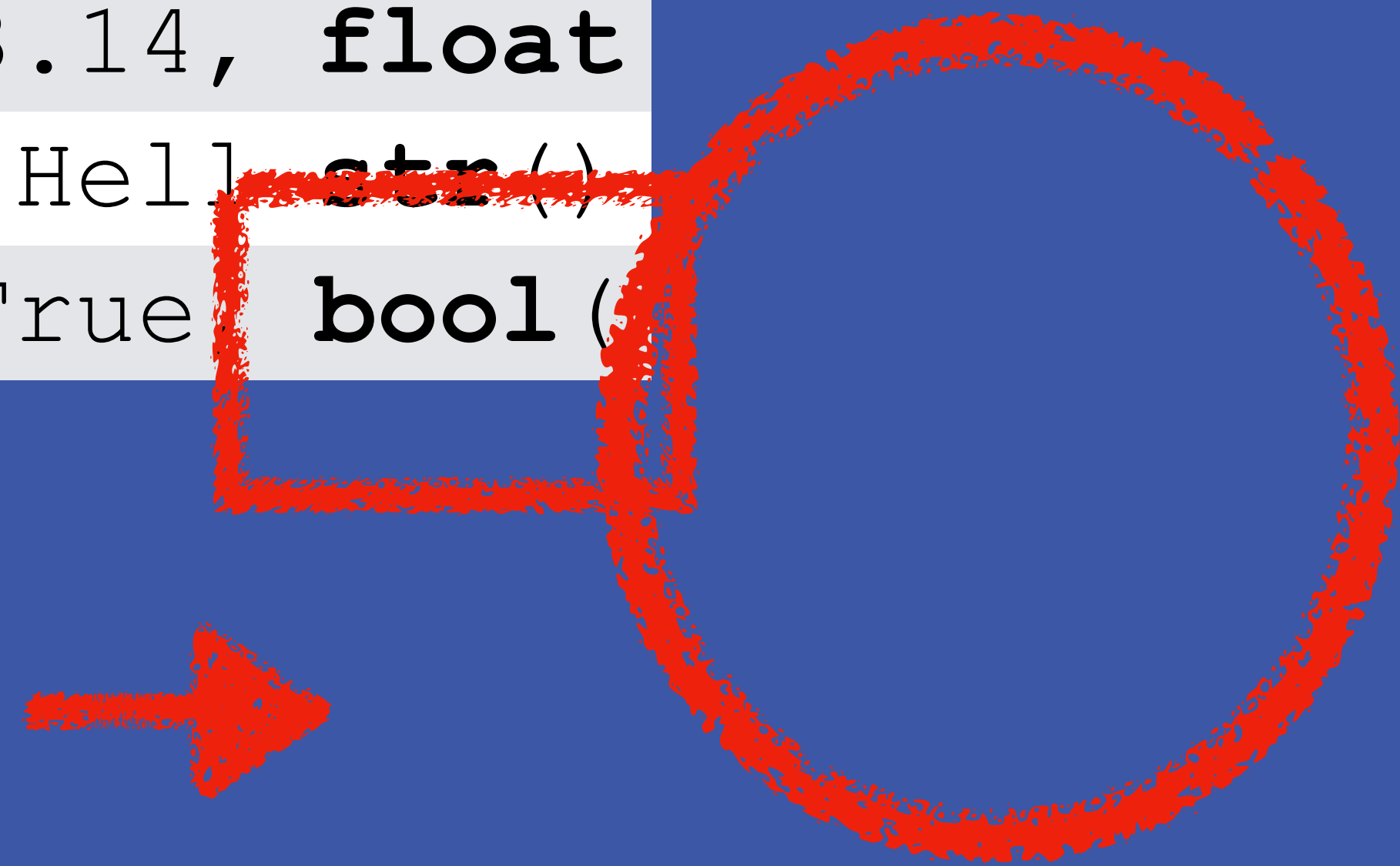
Extend your program

- * Instead of just showing the description and extract in one language, show it in **three** languages of your choice.
- * Look at the JSON output in a webbrowser, maybe you can display more interesting properties.

Tips

- * Everything you need for this exercise is in the 'lesson 6' chapters of the **examples-3** Jupyter Notebook.
- * Use the **replace()** method to replace spaces with strings.
- * Use **json.loads()** to convert the **text** of the GET request to a dictionary
- * Assigning the dictionary values to a variable (e.g. `extract = data["extract"]`) is handy here.
- * Again, F-strings are very useful in this program

Type	Examp	Conve
Integ	42,	int()
Float	3.14,	float
Strin	"Hell	str()
Boole	True	bool()



```
age = 20

if age < 20:
    print("option 1")
elif age <= 20 and age > 20:
    print("option 2")
else:
    print("option 3")
```

Compilation
Interpretation

