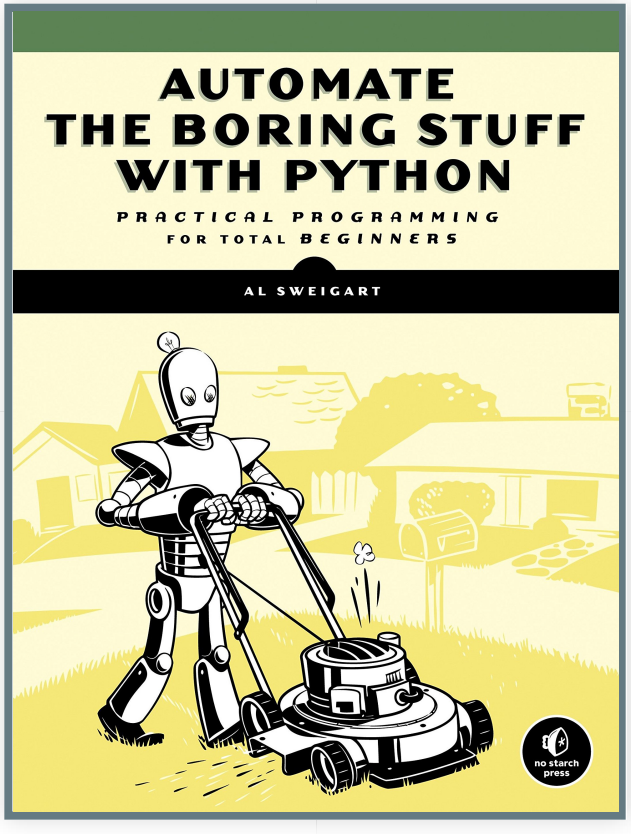
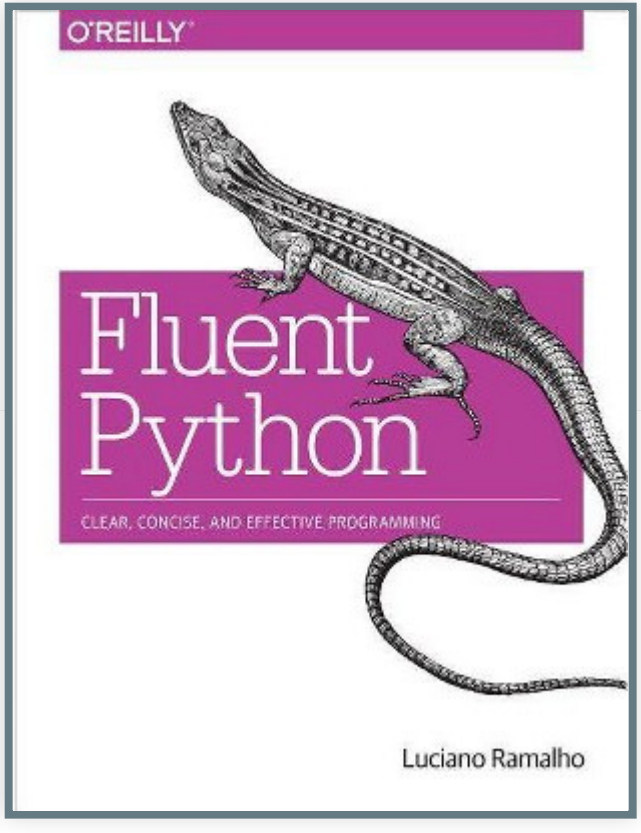
RESOURCES


AUTOMATE
THE BORING STUFF
WITH PYTHON
PRACTICAL PROGRAMMING
FOR TOTAL BEGINNERS
AL SWEIGART
no starch
press

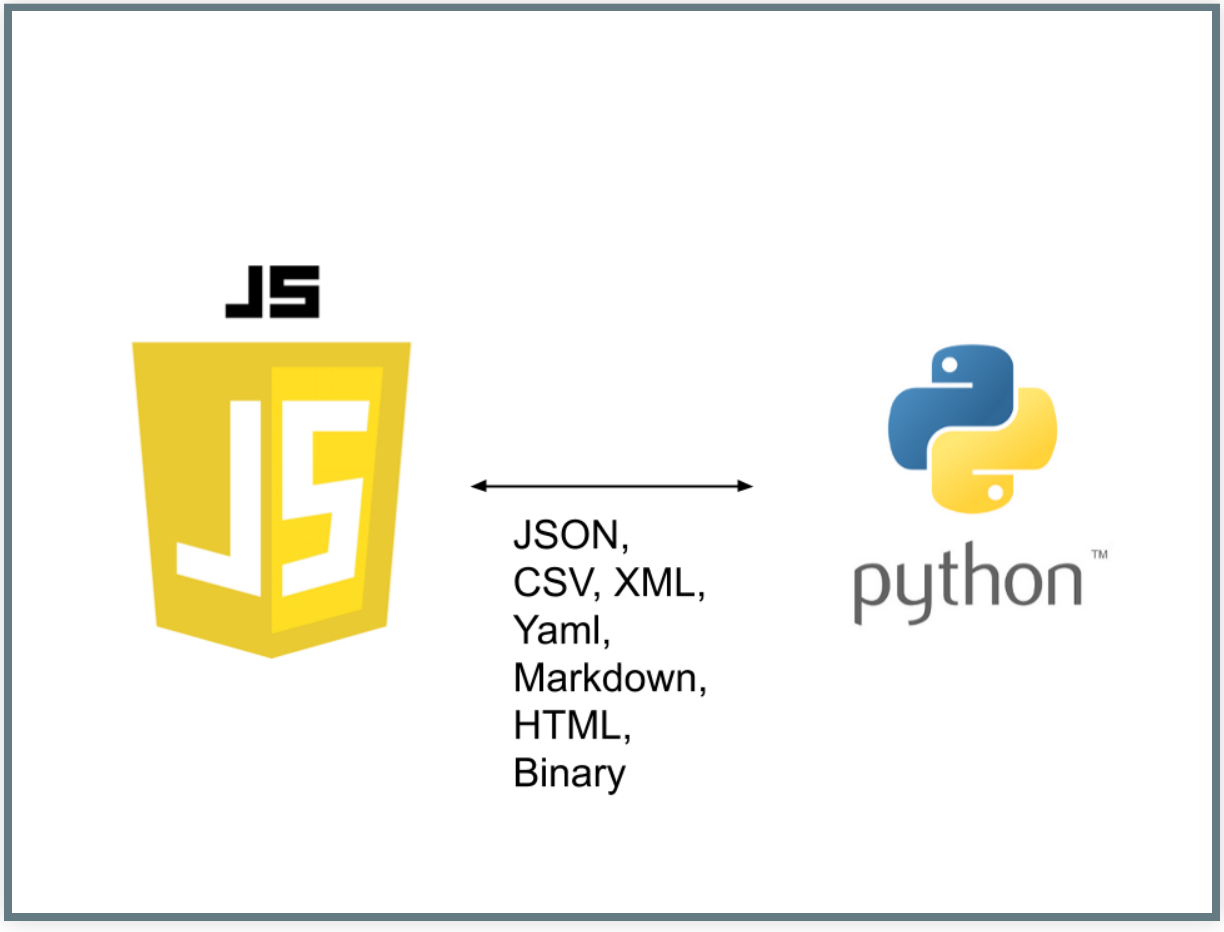

O'REILLY
Fluent
Python
CLEAR, CONCISE, AND EFFECTIVE PROGRAMMING
Luciano Ramalho

RESOURCES

https://junior.guru/
https://naucse.python.cz/
https://pyvo.cz/

LAST LESSON

FILE FORMATS



CSV

- Comma-separated values
- Each record on a new line
- If a value contains comma wrap it to double-quotes
- "Excel" format
- Used for tabular data
- example.csv z webu Engeto

```
Surname,Name,Full Name,Age,City,Job,Gender
Smith,John,"Smith, John",32,London,Programmer,
Doe,Joe,"Doe, Joe",34,Liverpool,,Male
Murphy,Ann,"Murphy, Ann",29,London,Admin,Female
Cook,Floyd,"Cook, Floyd",28,,Tester,Male
```

READING CSV

```
import csv
with open('example.csv') as file:
    reader = csv.reader(file)
    for row in reader:
        print(row)
```

```
['Surname', 'Name', 'Full Name', 'Age', 'City', 'Job', 'Gender']
['Smith', 'John', 'Smith, John', '32', 'London', 'Programmer', ''
['Doe', 'Joe', 'Doe, Joe', '34', 'Liverpool', '', 'Male']
['Murphy', 'Ann', 'Murphy, Ann', '29', 'London', 'Admin', 'Female
...
```

WRITING CSV

```
import csv

data = [
    ['Surname', 'Name', 'Full Name', 'Age', 'City', 'Job', 'Gende
    ['Smith', 'John', 'Smith, John', '32', 'London', 'Programmer'
    ['Doe', 'Joe', 'Doe, Joe', '34', 'Liverpool', '', 'Male'],
]

with open('write.csv', 'w') as file:
    writer = csv.writer(file)
    for row in data:
        writer.writerow(row)
```

```
Surname,Name,Full Name,Age,City,Job,Gender
Smith,John,"Smith, John",32,London,Programmer,
Doe,Joe,"Doe, Joe",34,Liverpool,,Male
```

CSV DICTREADER

```
import csv

with open('example.csv') as file:
    reader = csv.DictReader(file)
    for row in reader:
        print(row)
```

```
OrderedDict([('Surname', 'Smith'), ('Name', 'John'), ('Full Name'
OrderedDict([('Surname', 'Doe'), ('Name', 'Joe'), ('Full Name', '
OrderedDict([('Surname', 'Murphy'), ('Name', 'Ann'), ('Full Name'
```

CSV DICTWRITER

```
import csv

data = [
    {'Surname': 'Smith', 'Name': 'John', 'Full Name': 'Smith, Joh
    {'Surname': 'Doe', 'Name': 'Joe', 'Full Name': 'Doe, Joe', 'A
    {'Surname': 'Murphy', 'Name': 'Ann', 'Full Name': 'Murphy, An
]

with open('write.csv', 'w') as file:
    writer = csv.DictWriter(file, fieldnames=['Surname', 'Name', 'F
    for row in data:
        writer.writerow(row)
```

```
Smith,John,"Smith, John",32,London,Programmer,
Doe,Joe,"Doe, Joe",34,Liverpool,,Male
Murphy,Ann,"Murphy, Ann",29,London,Admin,Female
```

JSON

- JavaScript Object Notation
- Very close to printed Python dictionary or list
- Data types: Number, String, Boolean, Array, Object, `null`

```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 27,
  "address": {
        "streetAddress": "21 2nd Street",
        "city": "New York",
        "state": "NY",
        "postalCode": "10021-3100"
  },
  "phoneNumbers": [
        {
          "type": "office",
          "number": "646 555-4567"
        },
```

```
        {
          "type": "mobile",
          "number": "123 456-7890"
        }
  ],
  "children": [],
  "spouse": null
}
```

PYTHON TO JSON MAPPING

- Non-string dictionary keys are converted to strings.
- `True` is mapped to `true`, `False` is mapped to `false`.
- `None` is mapped to `null`.
- Single quotes on strings are converted to double quotes.
- `dict` is converted to `Object`.
- `list` is converted to `Array`.
- Some object cannot be automatically converted, e.g. `set`.

READING JSON

Wikipedia JSON example

```
import json

with open('example.json') as file:
    person = json.load(file)
    print(person)
```

```
{'firstName': 'John', 'lastName': 'Smith', 'isAlive': True, 'age':
```

WRITING JSON

```python
import json

person = {
    'fullName': 'John Smith',
    'phoneNumbers': [
        {'type': 'home', 'number': '212 555-1234'},
        {'type': 'office', 'number': '646 555-4567'},
        {'type': 'mobile', 'number': '123 456-7890'}
    ]
}

with open('write.json', 'w') as file:
    json.dump(person, file, indent=2)
```

```json
{
  "fullName": "John Smith",
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    },
```

```
    {
      "type": "mobile",
      "number": "123 456-7890"
    }
  ]
}
```

JSON IS FINE FOR LISTS TOO

```
data = ['John', 'Jane', 'Jennifer']

with open('write.json', 'w') as file:
    json.dump(data, file, indent=2)
```

```
[
  "John",
  "Jane",
  "Jennifer"
]
```

REQUESTS

Python HTTP library.

```
$ pip install requests
```

```
>>> import requests
>>> response = requests.get('https://seznam.cz')
>>> response.status_code
200
>>> response.text
'<!DOCTYPE html><html lang="cs" class="html-no-js html-no-flex htm
```

API

Application Programming Interface

https://swapi.co

```
>>> import requests
>>> response = requests.get('https://swapi.co/api/people/1')
>>> response.text
'{"name":"Luke Skywalker","height":"172","mass":"77","hair_color"
>>> response.json()
{'name': 'Luke Skywalker',
 'height': '172',
 'mass': '77',
 'hair_color': 'blond',
 'skin_color': 'fair',
...
```

TODAY'S LESSON

HTML

HTTP

WEB SCRAPING

HTML

- Hypertext Markup Language
- W3C tutorial

```
<body>
    <h1>This is a heading</h1>
    <p>And this is the paragraph number 1</p>
    <p> This is the paragraph number 2</p>
</body>
```

THIS IS A HEADING

And this is the paragraph number 1

This is the paragraph number 2

HTML ELEMENT

```
<opening_tag>Element Content</closing_tag>
```

```
<p>This is a <strong>strong</strong>  <em>statement</em>. <strong
```

This is a **strong** *statement*. ***Watch out!***

Check more at https://htmlcheatsheet.com/

TAG ATTRIBUTES

```
<a href="https://www.w3schools.com/html/">W3C tutorial</a>
```

W3C tutorial

HTML DOCUMENT ORGANIZATION

EXERCISE

Change content of a web page using developer tools.

Python je supr dupr jazyk

Můžete si importovat TAKHLE velke tanky.

Dnes	Zítra	Pozítří	Středa	Aktuální srážky
2 °C	3 °C	2 °C	3 °C	

Předpověď na 9 dní

NEJNOVĚJŠÍ

zobrazovat sport

19:36 Arsenal pod Ljungbergem jen remizoval, Leicester je po výhře druhý

19:33 Opozice žádá zveřejnění auditu a případnou rezignaci Andreje Babiše

19:33 Neudržitelná. Jonesová přepsala český ligový rekord, nasázela 50 bodů

19:32 České Budějovice jsou k nezastavení, vyhrály i Slavia, Plzeň a Sparta

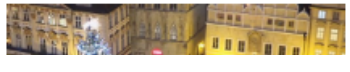
19:30 ONLINE: Litvínov utíká Pardubicím. Kometa otočila, Mladá Boleslav vede

Další dnešní články (140)

KOMENTÁŘ: Pyrrhovo vítězství. Novotný udělal radost ruské propagandě

Provokatér, bavič, bulvární novinář a také starosta MČ Praha - Řeporyje Pavel Novotný

iDNES tv | Živě | Slow | Pořady | Rozstřel



HTTP

Hypertext Transfer Protocol
Client - Server architecture

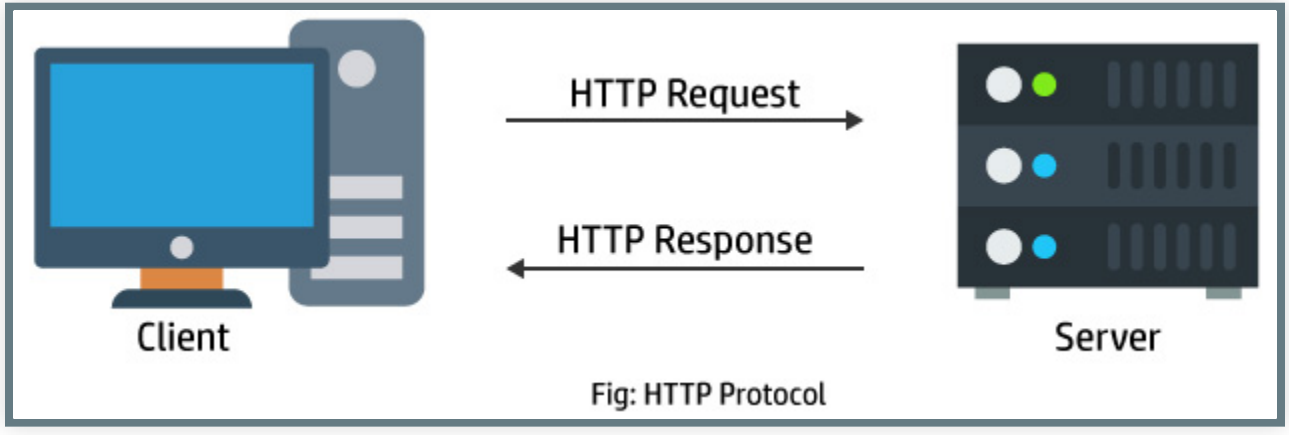


Fig: HTTP Protocol

```
GET / HTTP/1.1
Host: www.example.com
```

Response:

```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 138
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
ETag: "3f80f-1b6-3e1cb03b"
Accept-Ranges: bytes
Connection: close

<html>
  <head>
        <title>An Example Page</title>
  </head>
  <body>
        <p>Hello World, this is a very simple HTML document.</p>
```

URL

```
protocol://host/path?query#fragment
```

```
https://api.openweathermap.org/data/2.5/weather?q=London
```

HTTP STATUS CODES

- 1xx: Informational
- 2xx: Success
- 3xx: Redirection
- 4xx: Client Error
- 5xx: Server Error

HTTP METHODS

GET

requests a given document or a video - in general a resource - this method is used to retrieve data

POST

method used to send data to a server, e.g. for authentication or registration etc.

PUT

similar to POST method in that we send some data to the server, but the server should update a given (existing) record or other resource to a new value, we send it

DELETE

removes a given resource

HEAD

asks for a response identical to that of a GET request, but without the response body

OPTIONS

asks server, what methods are supported by a given resource - given web page etc.

EXERCISE

Check HTTP requests and responses using web browser for your favorite web.
Check status codes, queried domains, file types and headers.

WEB SCRAPING

- Download HTML from the server
- Retrieve needed information

Beautiful Soup

```
$ pip install beautifulsoup4
$ pip install requests
```

BEAUTIFUL SOUP

```
>>> import requests
>>> from bs4 import BeautifulSoup as BS
>>> r = requests.get("https://example.com")
>>> soup = BS(r.text)

>>> soup.p
<p>This domain is for use in illustrative examples in documents.
domain in literature without prior coordination or asking for per

>>> soup.find_all('p')
[<p>This domain is for use in illustrative examples in documents.
 domain in literature without prior coordination or asking for pe
 <p><a href="https://www.iana.org/domains/example">More informati

>>> soup.a['href']
'https://www.iana.org/domains/example'
```

EXERCISE

Gather all links from http://example.webscraping.com

```
#
/places/default/user/register?_next=/places/default/index
/places/default/user/login?_next=/places/default/index
/places/default/index
/places/default/search
/places/default/view/Afghanistan-1
/places/default/view/Aland-Islands-2
/places/default/view/Albania-3
/places/default/view/Algeria-4
/places/default/view/American-Samoa-5
/places/default/view/Andorra-6
/places/default/view/Angola-7
/places/default/view/Anguilla-8
/places/default/view/Antarctica-9
/places/default/view/Antigua-and-Barbuda-10
/places/default/index/1
```

```
for a in soup.find_all('a'):
    print(a['href'])
```

```
>>> import urllib.urlparse
>>> urllib.parse.urljoin('http://example.webscraping.com/', '/pla
http://example.webscraping.com/places/default/view/Afghanistan-1'
```

HTML ID ATTRIBUTE

```
>>> results = soup.find(id='results')
>>> results.find_all('a')
[<a href="/places/default/view/Afghanistan-1"><img src="/places/s
 <a href="/places/default/view/Aland-Islands-2"><img src="/places
 <a href="/places/default/view/Albania-3"><img src="/places/stati
 <a href="/places/default/view/Algeria-4"><img src="/places/stati
 <a href="/places/default/view/American-Samoa-5"><img src="/place
 <a href="/places/default/view/Andorra-6"><img src="/places/stati
 <a href="/places/default/view/Angola-7"><img src="/places/static
 <a href="/places/default/view/Anguilla-8"><img src="/places/stat
 <a href="/places/default/view/Antarctica-9"><img src="/places/st
 <a href="/places/default/view/Antigua-and-Barbuda-10"><img src="
```

EXERCISE

Gather all links to states on a page

```
http://example.webscraping.com/places/default/view/Afghanistan-1
http://example.webscraping.com/places/default/view/Aland-Islands-
http://example.webscraping.com/places/default/view/Albania-3
http://example.webscraping.com/places/default/view/Algeria-4
http://example.webscraping.com/places/default/view/American-Samoa
http://example.webscraping.com/places/default/view/Andorra-6
http://example.webscraping.com/places/default/view/Angola-7
http://example.webscraping.com/places/default/view/Anguilla-8
http://example.webscraping.com/places/default/view/Antarctica-9
http://example.webscraping.com/places/default/view/Antigua-and-Ba
```

```
BASE_PATH = 'http://example.webscraping.com/'
results = soup.find(id="results")
for a in results.find_all('a'):
        print(urllib.parse.urljoin(BASE_PATH, a['href']))
```

EXERCISE

Create a record about each country.

```
>>> scrape_country('http://example.webscraping.com/places/default
{'country': 'Afghanistan', 'population': 29121286}
```

Hint:

```
soup.find(id='places_country__row').find('td', class_='w2p_fw').get
```

```
def scrape_country(url):
    r = requests.get(url)
    s = BS(r.text)
    country = s.find(id='places_country__row').find('td', class_='w
    population = s.find(id='places_population__row').find('td', cla
    population = int(population.replace(',', ''))
    return {'country': country, 'population': population}
```

EXERCISE

Scrape all pages, scrape each country and store the list of all records to json.

END