LAST LESSON

import

```
>>> import random
>>> random.random()
0.16541488247158354
>>> from random import randint
>>> randint(1, 10)
6
>>> from random import randint as my_randint
>>> my_randint(1, 10)
3
>>> from random import * # not recommended
>>> choice([1, 2, 3])
2
```

PYTHON STANDARD LIBRARY

Batteries included

Standard library

os

```
>>> import os
>>> os.listdir()
['prezentace']
>>> os.getcwd()
'/home/psebek/projects/engeto/10'
>>> os.mkdir('test_dir')
>>> os.listdir()
['test_dir', 'prezentace']
```

os.path

```
>>> import os.path  # or just os
>>> os.path.exists('test_dir')
True
>>> os.path.isfile('test_dir')
False
>>> os.path.isdir('test_dir')
True
>>> path = os.path.join('test_dir', 'sub_folder', 'file.txt')
>>> path
'test_dir/sub_folder/file.txt'
>>> os.path.abspath(path)
'/home/psebek/projects/engeto/10/test_dir/sub_folder/file.txt'
>>> os.path.basename(path)
'file.txt'
```

THIRD-PARTY PACKAGES

PyPI

VIRTUAL ENVIRONMENT

Solves package versions conflicts

Tutorial venv

```
$ python3 -m venv .env
$ ls
.env
$ source .env/bin/activate
(.env) $ python --version
3.7.5
```

pip

Installs third-party packages

```
(.env) $ pip list
Package    Version
---------- -------
pip        19.1.1
setuptools 41.2.0
(.env) $ pip install tabulate
...
Successfully installed tabulate-0.8.6
(.env) $ python
>>> import tabulate
>>> print(tabulate.tabulate([[1, 2], [3, 4]]))
-  -
1  2
3  4
-  -
```

IMPORTING OWN MODULE

helpers.py

```
def avg(sequence):
        return sum(sequence) / len(sequence)
```

compute.py

```
import helpers
print(helpers.avg([1, 2, 3, 4])
```

terminal

```
$ python compute.py
2.5
```

IMPORTING OWN MODULE

helpers.py

```
def avg(sequence):
        return sum(sequence) / len(sequence)

print('We are in helpers.py')
print(avg([1, 2])
```

compute.py

```
import helpers
print('We are in compute.py')
print(helpers.avg([1, 2, 3, 4])
```

```
$ python compute.py
We are in helpers.py
1.5
We are in compute.py
2.5
```

IMPORTING OWN MODULE

helpers.py

```
def avg(sequence):
        return sum(sequence) / len(sequence)
if __name__ == '__main__': # <---
        print('We are in helpers.py')
        print(avg([1, 2])
```

compute.py

```
import helpers
print('We are in compute.py')
print(helpers.avg([1, 2, 3, 4])
```

```
$ python compute.py
We are in compute.py
2.5
```

MODULE VS PACKAGE

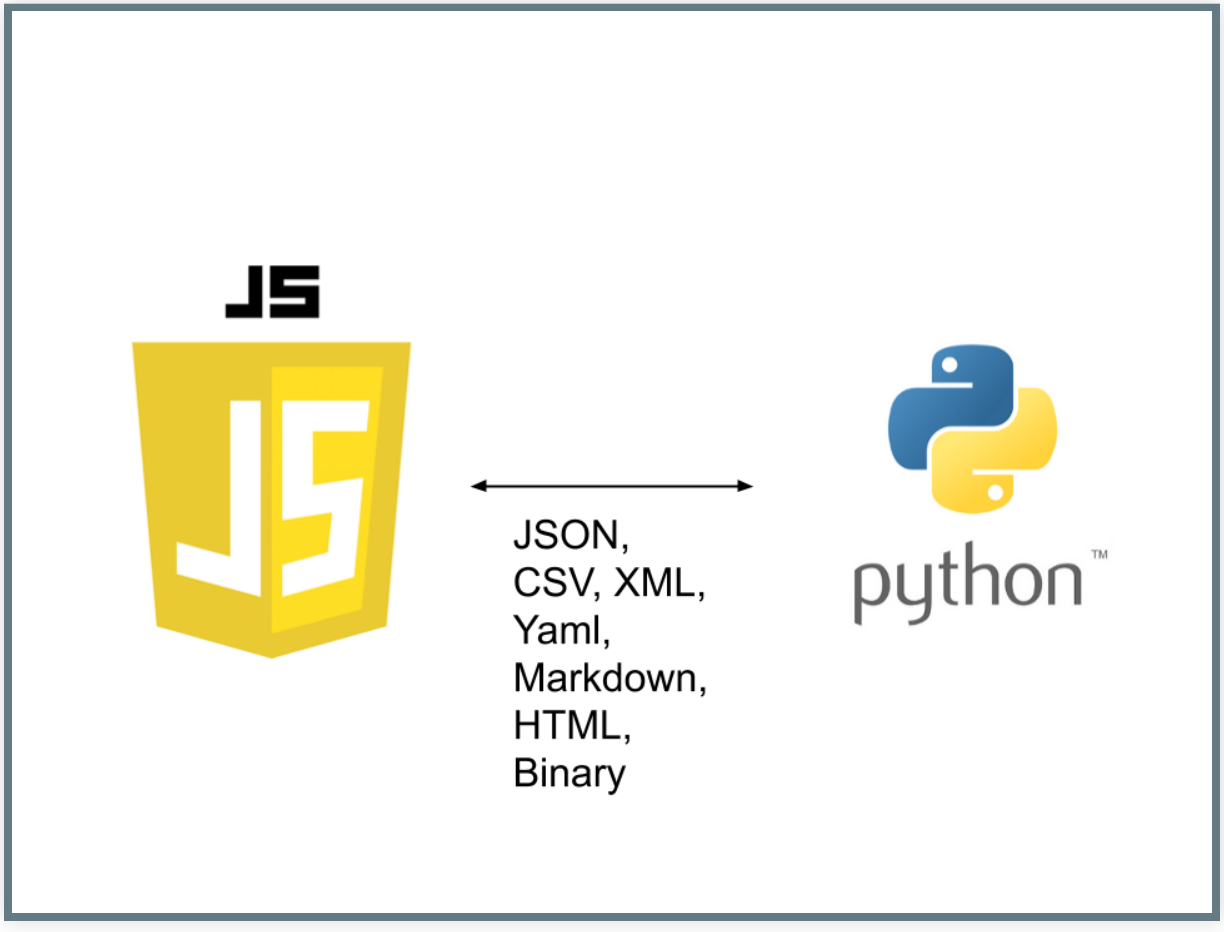
```
package/
        - __init__.py
        - a.py
        - b.py
```

```
import package.a
import package.b

package.a.function_1()
package.b.function_2()
```

TODAY'S 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 f:
    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
...
```

EXERCISE

Print every person from London.

```
$ python read_csv.py
['Smith', 'John', 'Smith, John', '32', 'London', 'Programmer', ''
['Murphy', 'Ann', 'Murphy, Ann', '29', 'London', 'Admin', 'Female
['Harris', 'Roy', 'Harris, Roy', '22', 'London', 'Junior Programm
['Galagher', 'Fred', 'Galagher, Fred', '38', 'London', 'Programme
['Murphy', 'John', 'Murphy, John', '35', 'London', 'Programmer',
```

Print only their full names

```
$ python read_csv.py
Smith, John
Murphy, Ann
Harris, Roy
Galagher, Fred
Murphy, John
```

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
```

EXERCISE

Read from `example.csv` and write to `londoners.csv` only people from London.

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':
```

EXERCISE

Read `example.json`, create and print a dictionary with keys:

- `fullName` (`firstName + ' ' + lastName`)
- `phoneNumbers`

```
{'fullName': 'John Smith', 'phoneNumbers': [{'type': 'home', 'num
```

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"
]
```

EXERCISE

Write a dictionary with your name and age to a file.

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',
...
```

EXERCISE

Use `requests` to first download content of https://swapi.co/api/people/ and then download information about Luke's homeworld.

END