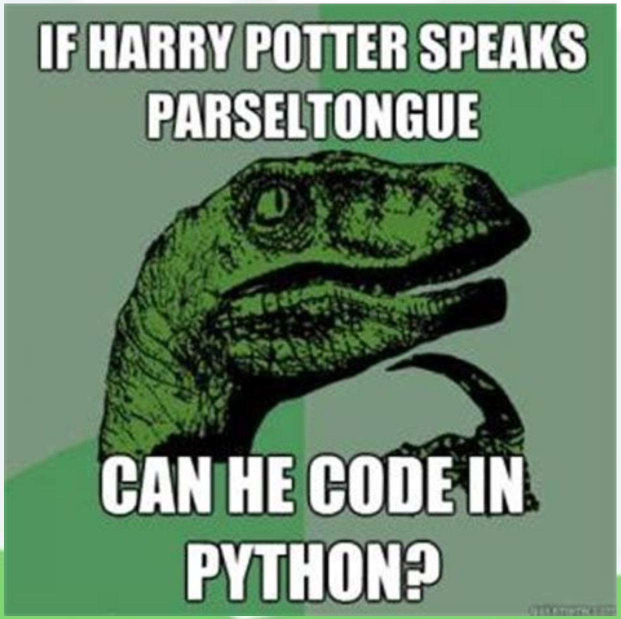
1. gyakorlat



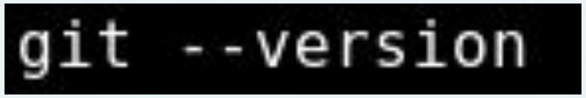
Git/Python

Félév követelmények

- Pusholni az órai anyagot
- Heti házikat pusholni határidőre
- Opcionális beadandók

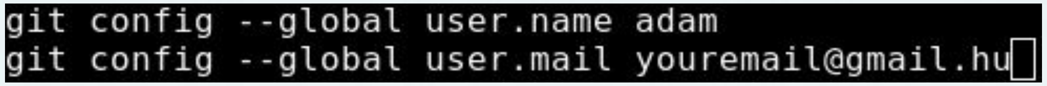
Git local

- Fent van-e a git



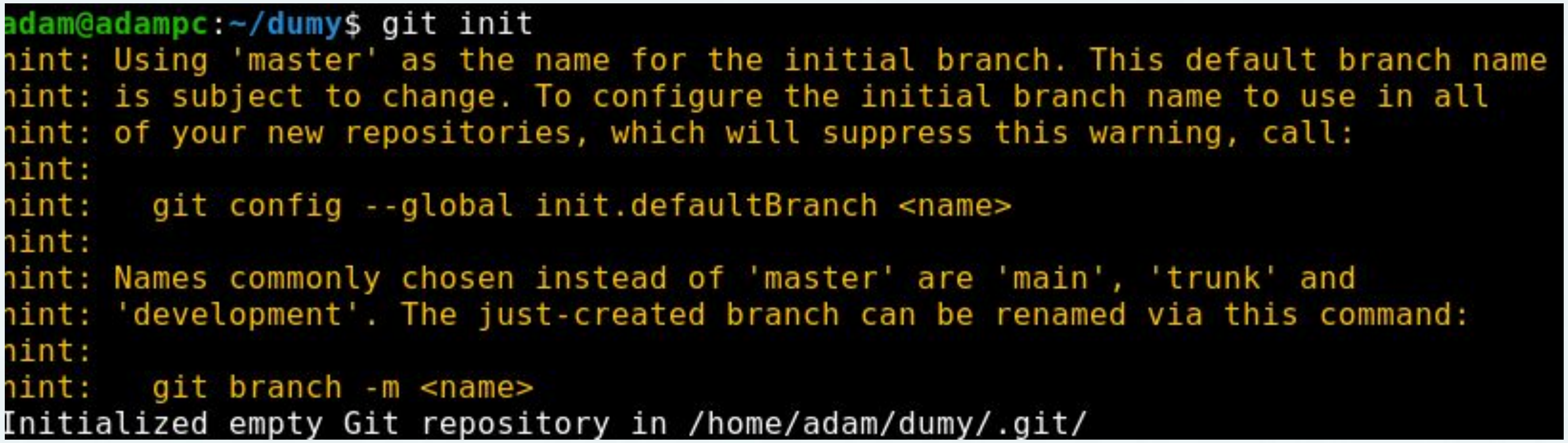
```
git --version
```

- Felhasználó/email init



```
git config --global user.name adam
git config --global user.mail youremail@gmail.hu
```

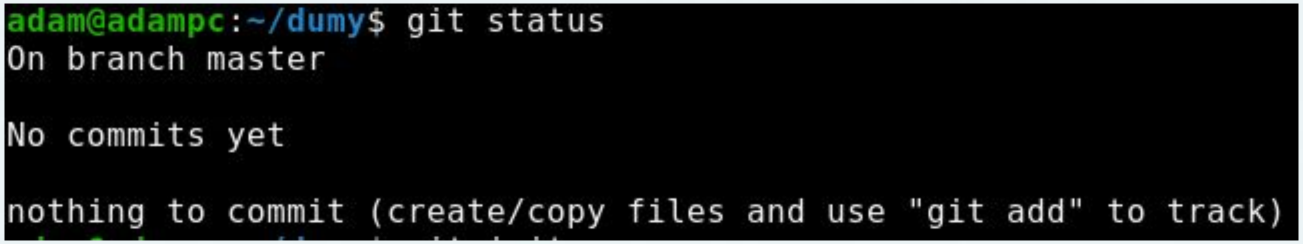
- Loca repo init



```
adam@adampc:~/dumy$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/adam/dumy/.git/
```

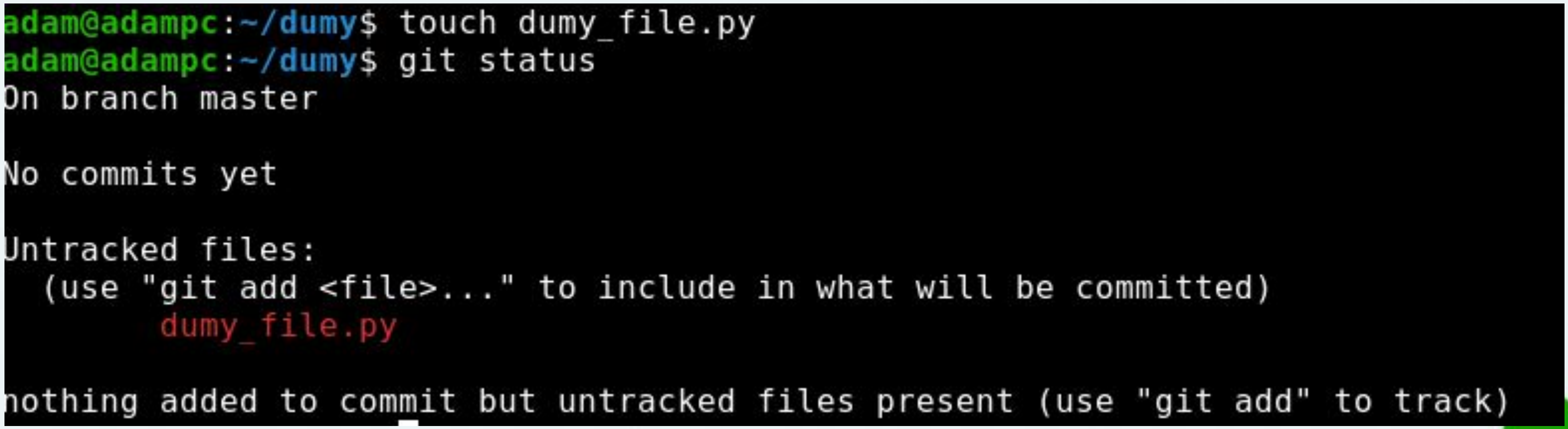
Git local

- Status



```
adam@adampc:~/dumy$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
```

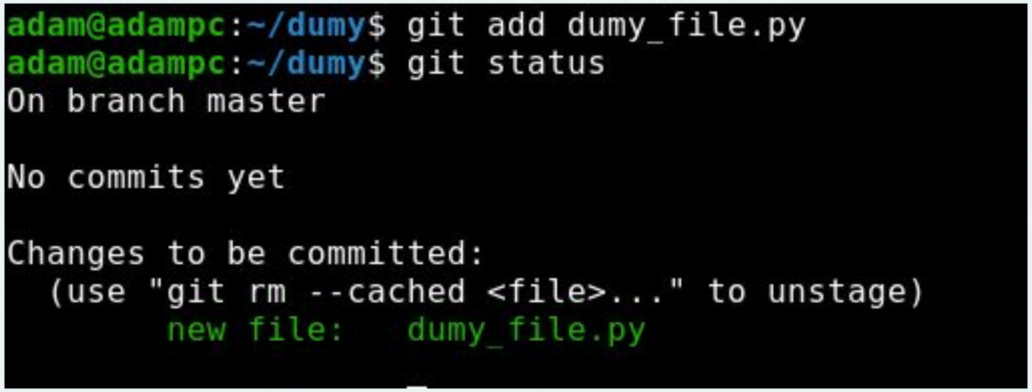
- Status file létrehozás után
-



```
adam@adampc:~/dumy$ touch dumy_file.py
adam@adampc:~/dumy$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        dumy_file.py

nothing added to commit but untracked files present (use "git add" to track)
```

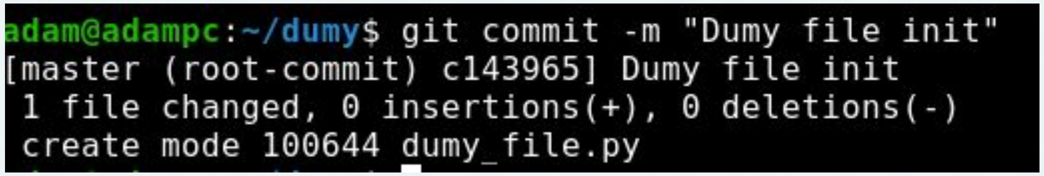
Git local

- Staged file



```
adam@adampc:~/dumy$ git add dumy_file.py
adam@adampc:~/dumy$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   dumy_file.py
```

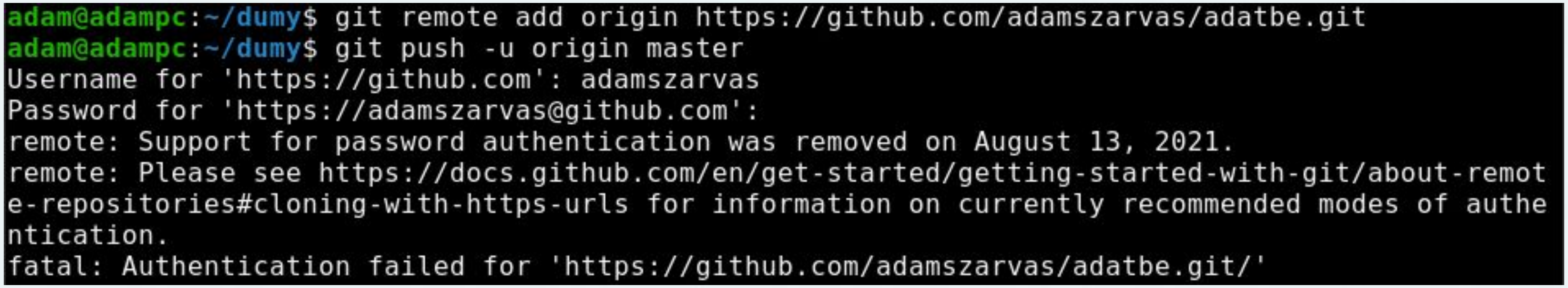
- Commit staged files



```
adam@adampc:~/dumy$ git commit -m "Dumy file init"
[master (root-commit) c143965] Dumy file init
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 dumy_file.py
```

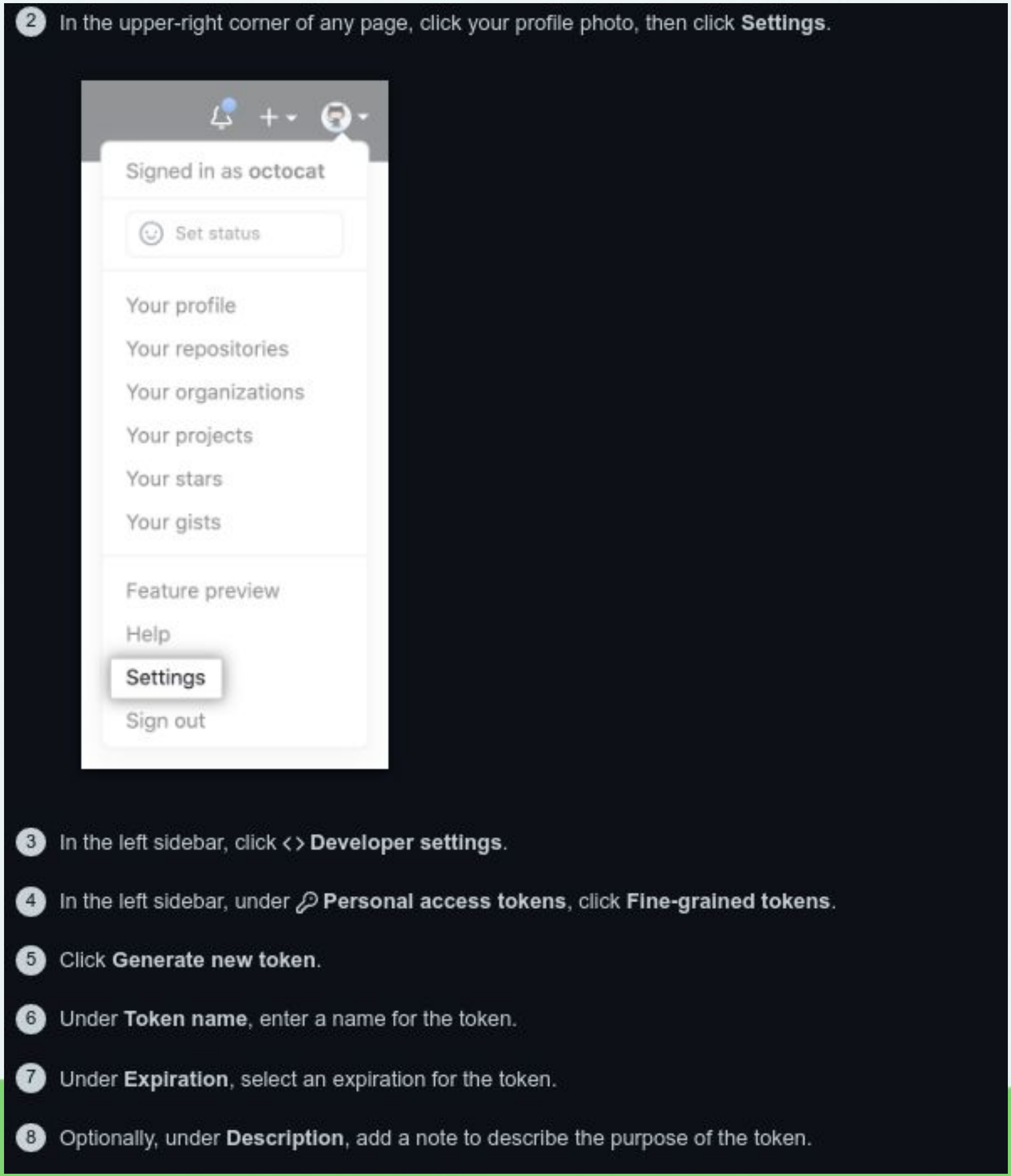
GitHub

- Connect remote and local



```
adam@adampc:~/dumy$ git remote add origin https://github.com/adamszarvas/adatbe.git
adam@adampc:~/dumy$ git push -u origin master
Username for 'https://github.com': adamszarvas
Password for 'https://adamszarvas@github.com':
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/en/get-started/getting-started-with-git/about-remot
e-repositories#cloning-with-https-urls for information on currently recommended modes of authe
ntication.
fatal: Authentication failed for 'https://github.com/adamszarvas/adatbe.git/'
```

Git token



2 In the upper-right corner of any page, click your profile photo, then click **Settings**.

3 In the left sidebar, click <> **Developer settings**.

4 In the left sidebar, under **Personal access tokens**, click **Fine-grained tokens**.

5 Click **Generate new token**.

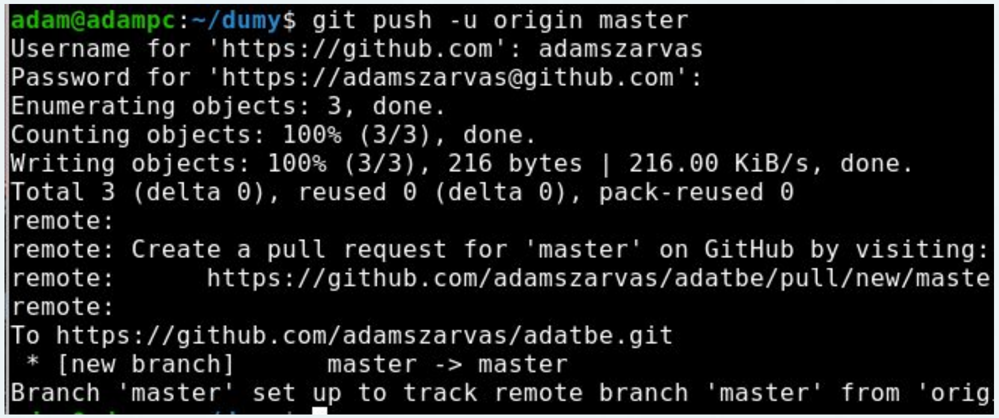
6 Under **Token name**, enter a name for the token.

7 Under **Expiration**, select an expiration for the token.

8 Optionally, under **Description**, add a note to describe the purpose of the token.

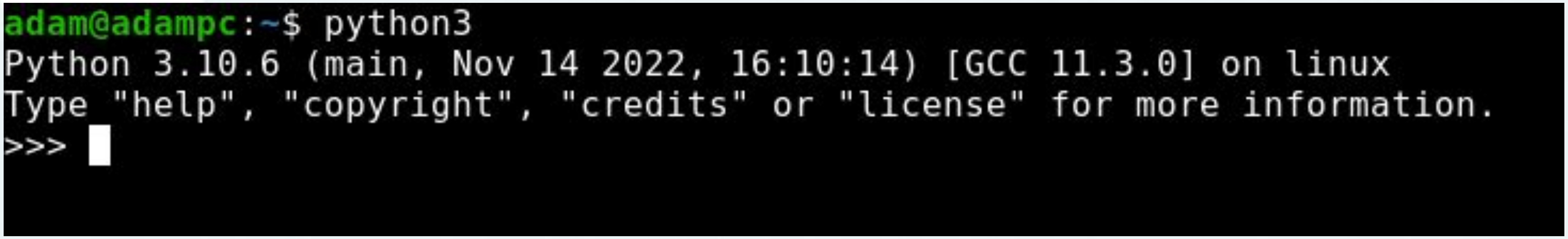
GitHub

- Push commit
- *master helyett main a default GitHub-on



```
adam@adampc:~/dumy$ git push -u origin master
Username for 'https://github.com': adamszarvas
Password for 'https://adamszarvas@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 216 bytes | 216.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/adamszarvas/adatbe/pull/new/maste
remote:
To https://github.com/adamszarvas/adatbe.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'orig
```

Python

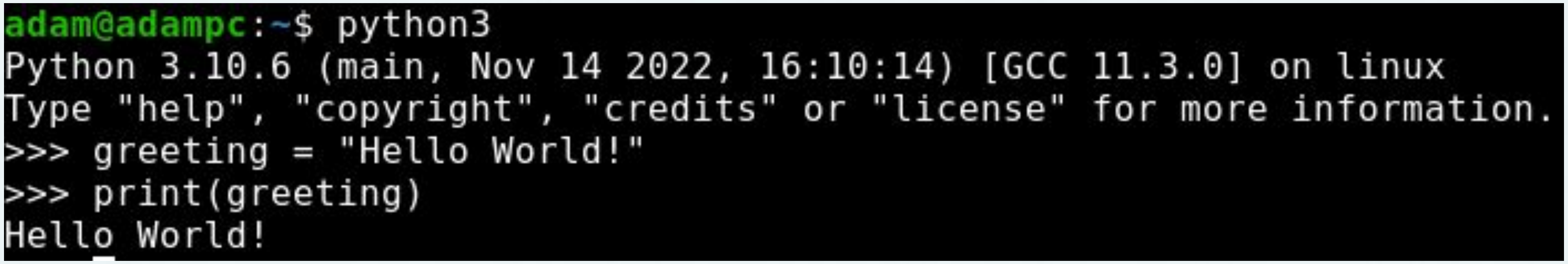


```
adam@adampc:~$ python3
Python 3.10.6 (main, Nov 14 2022, 16:10:14) [GCC 11.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

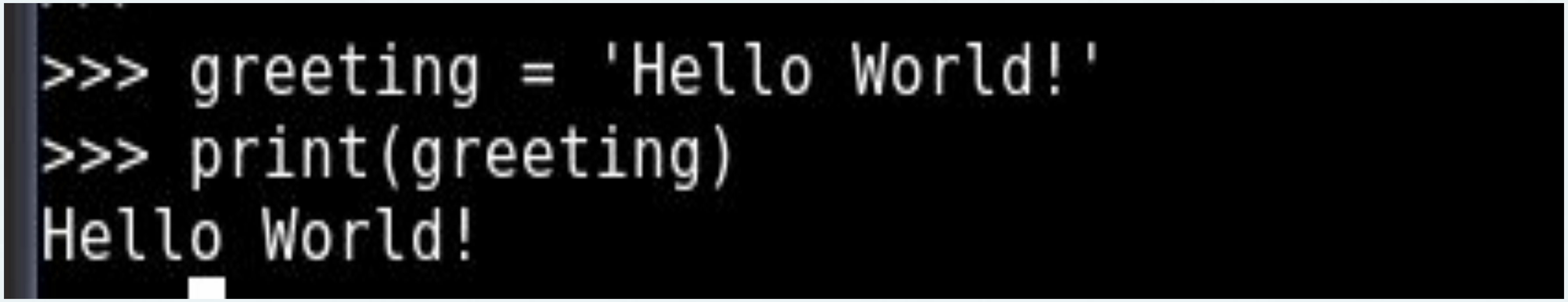
Feladat

- Hozz létre egy változót ezzel az értékkel:
- “Hello World!”
- Írasd ki a változó értékét egyben!
- Írasd ki a változó értékét egyesével!
- Készíts egy függvényt ami kiírja a változó értékét!
- Készíts egy .py kiterjesztésű fájlt a mappádba és az előző feladatokat írd le oda is

Feladat

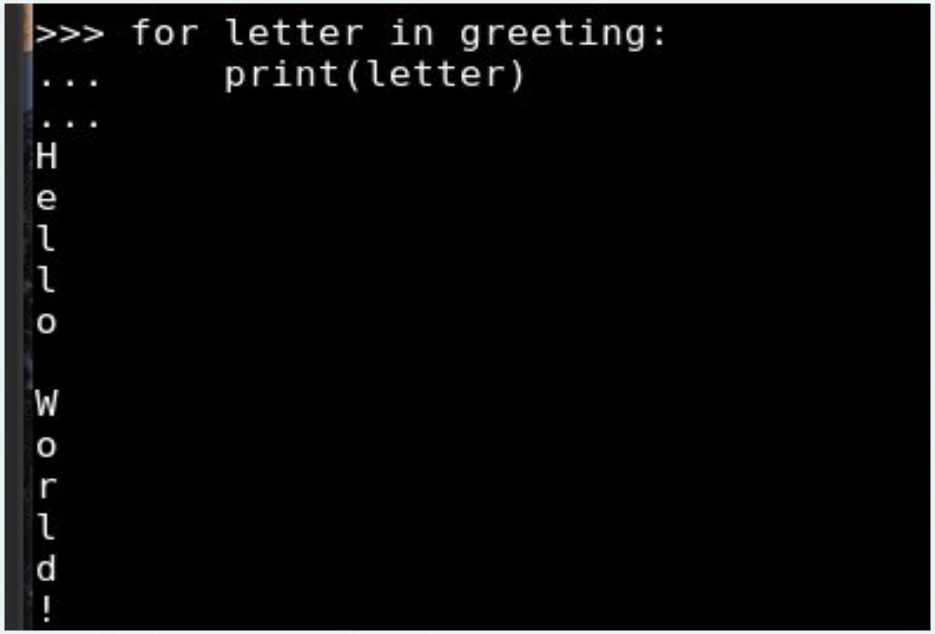


```
adam@adampc:~$ python3
Python 3.10.6 (main, Nov 14 2022, 16:10:14) [GCC 11.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> greeting = "Hello World!"
>>> print(greeting)
Hello World!
```

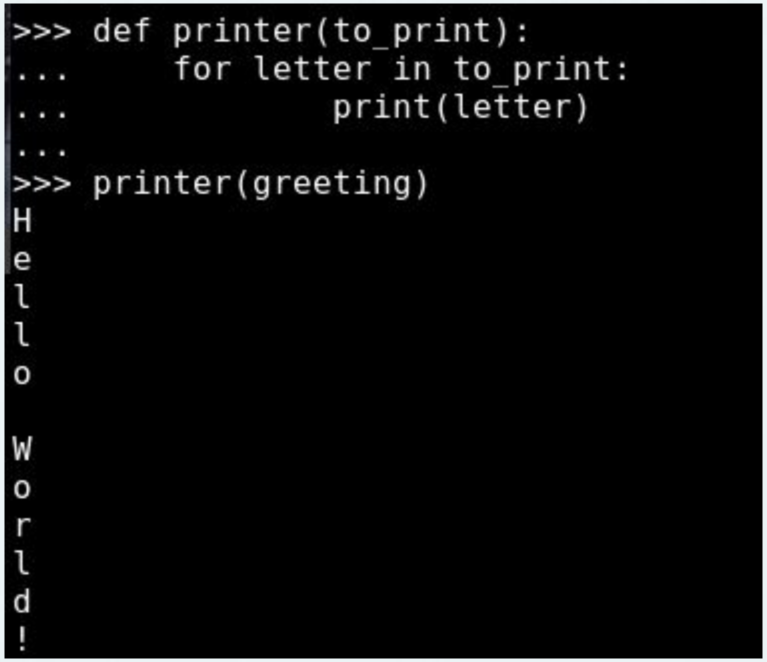


```
>>> greeting = 'Hello World!'
>>> print(greeting)
Hello World!
```

Feladat



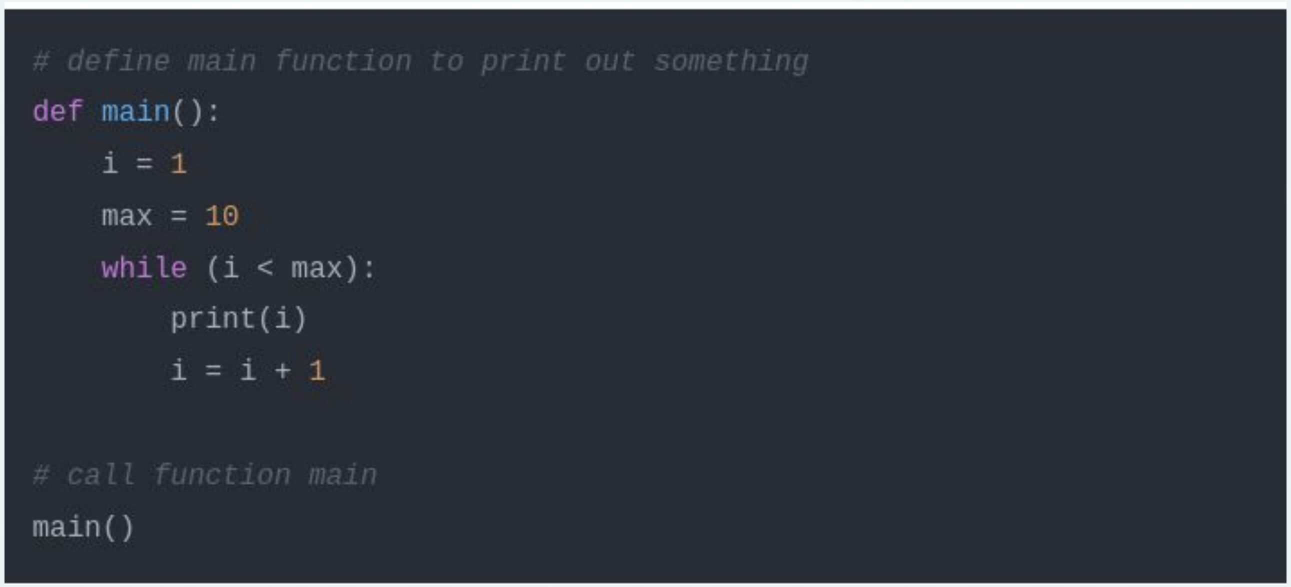
```
>>> for letter in greeting:
...     print(letter)
...
H
e
l
l
o
 
W
o
r
l
d
!
```



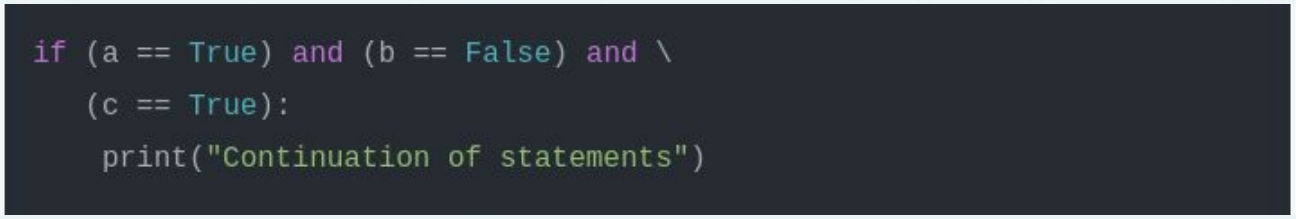
```
>>> def printer(to_print):
...     for letter in to_print:
...             print(letter)
...
>>> printer(greeting)
H
e
l
l
o
 
W
o
r
l
d
!
```

Python

- Syntax



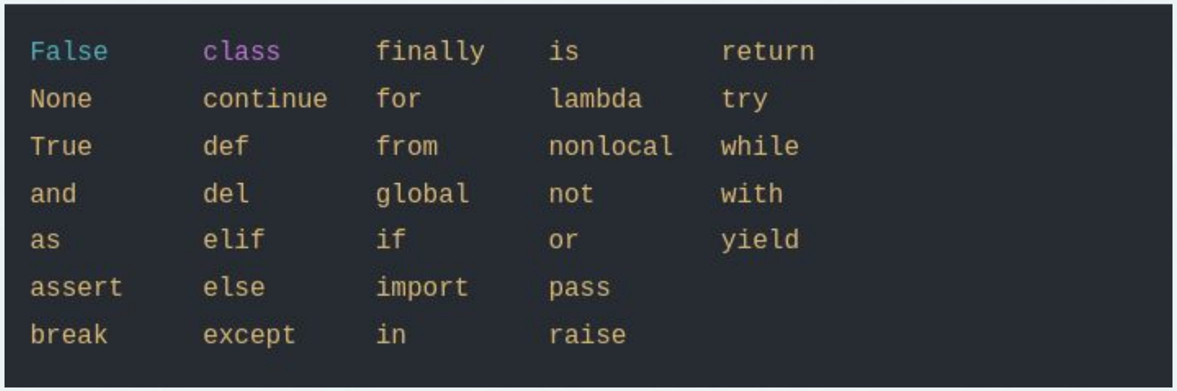
```python
# define main function to print out something
def main():
    i = 1
    max = 10
    while (i < max):
        print(i)
        i = i + 1

# call function main
main()
```



```python
if (a == True) and (b == False) and \
   (c == True):
    print("Continuation of statements")
```

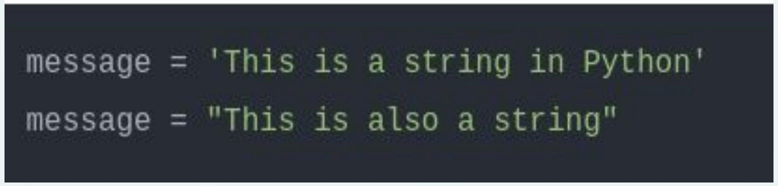
Python

- Keywords

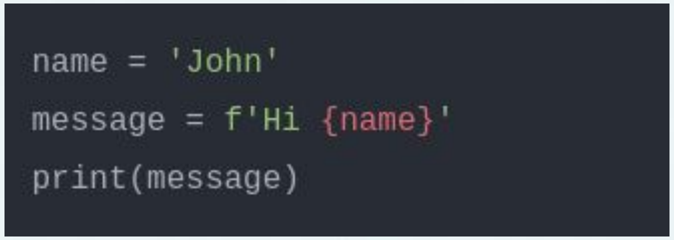


```
False      class      finally    is         return
None       continue   for        lambda     try
True       def        from       nonlocal   while
and        del        global     not        with
as         elif       if         or         yield
assert     else       import     pass
break      except     in         raise
```

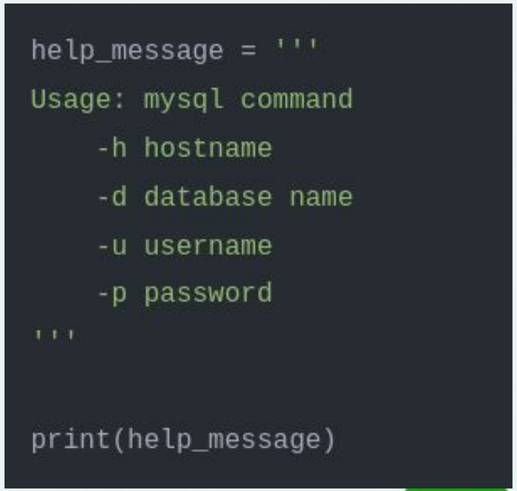
- Strings



```
message = 'This is a string in Python'
message = "This is also a string"
```



```
name = 'John'
message = f'Hi {name}'
print(message)
```



```
help_message = '''
Usage: mysql command
    -h hostname
    -d database name
    -u username
    -p password
'''

print(help_message)
```

Python

- If-else

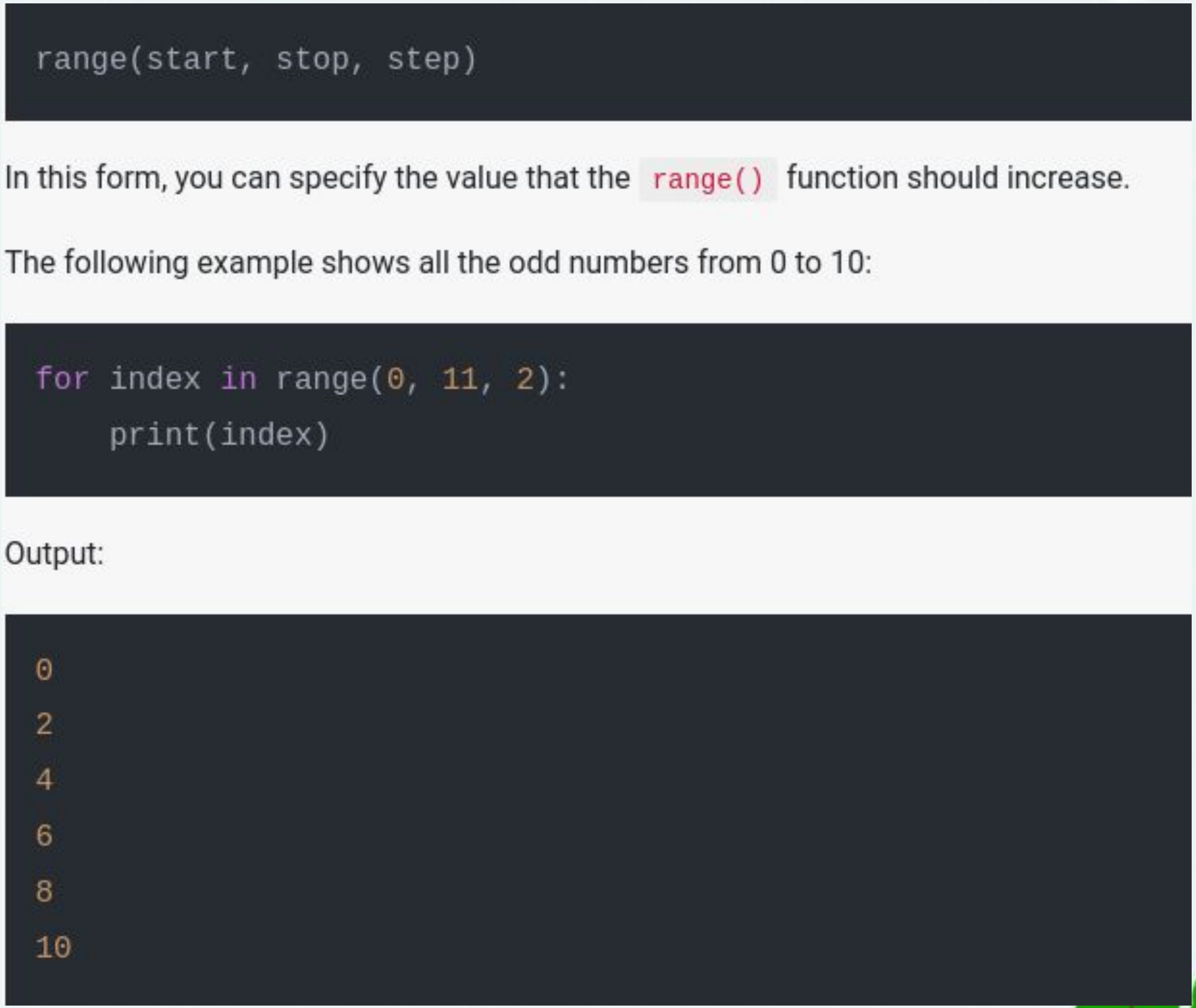
```
value_if_true if condition else value_if_false
```

```
condition ? value_if_true : value_if_false
```

```
if if-condition:
    if-block
elif elif-condition1:
    elif-block1
elif elif-condition2:
    elif-block2
...
else:
    else-block
```

Python

- For loop



```
range(start, stop, step)
```

In this form, you can specify the value that the `range()` function should increase.

The following example shows all the odd numbers from 0 to 10:

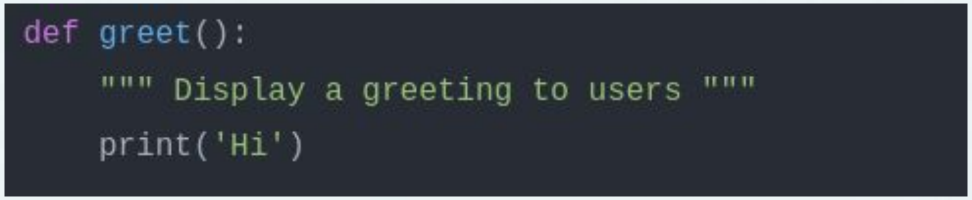
```
for index in range(0, 11, 2):
    print(index)
```

Output:

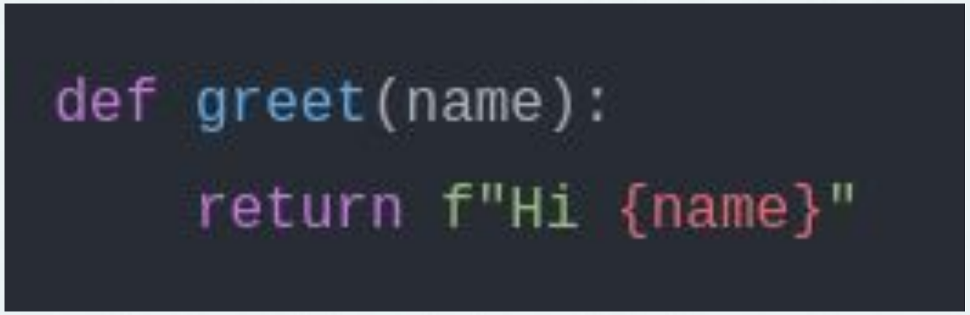
```
0
2
4
6
8
10
```

Python

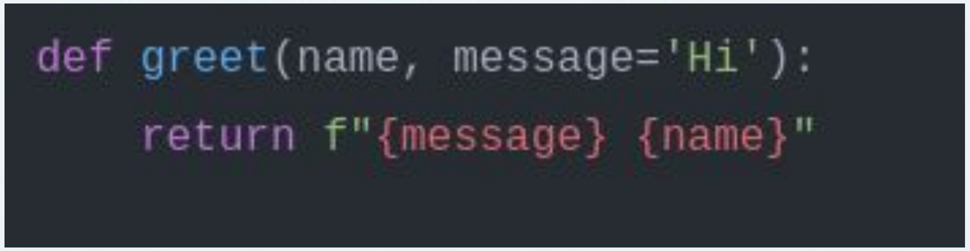
- Functions



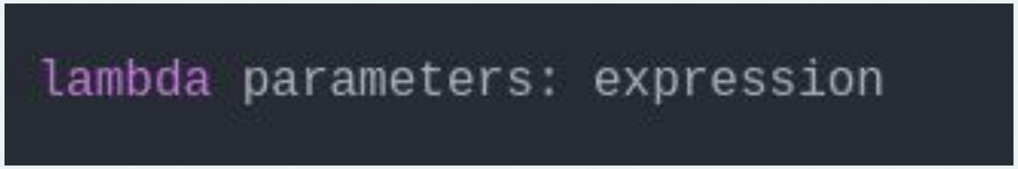
```python
def greet():
    """ Display a greeting to users """
    print('Hi')
```



```python
def greet(name):
    return f"Hi {name}"
```



```python
def greet(name, message='Hi'):
    return f"{message} {name}"
```



```python
lambda parameters: expression
```

Python

- List

```
empty_list = []
```

```
numbers = [1, 3, 2, 7, 9, 4]
```

```
coordinates = [[0, 0], [100, 100], [200, 200]]
```

```
numbers = [1, 3, 2, 7, 9, 4]
print(numbers[1])
```

```
numbers = [1, 3, 2, 7, 9, 4]
print(numbers[-1])
print(numbers[-2])
```

```
numbers = [1, 3, 2, 7, 9, 4]
numbers.append(100)

print(numbers)
```

```
numbers = [1, 3, 2, 7, 9, 4]
del numbers[0]

print(numbers)
```

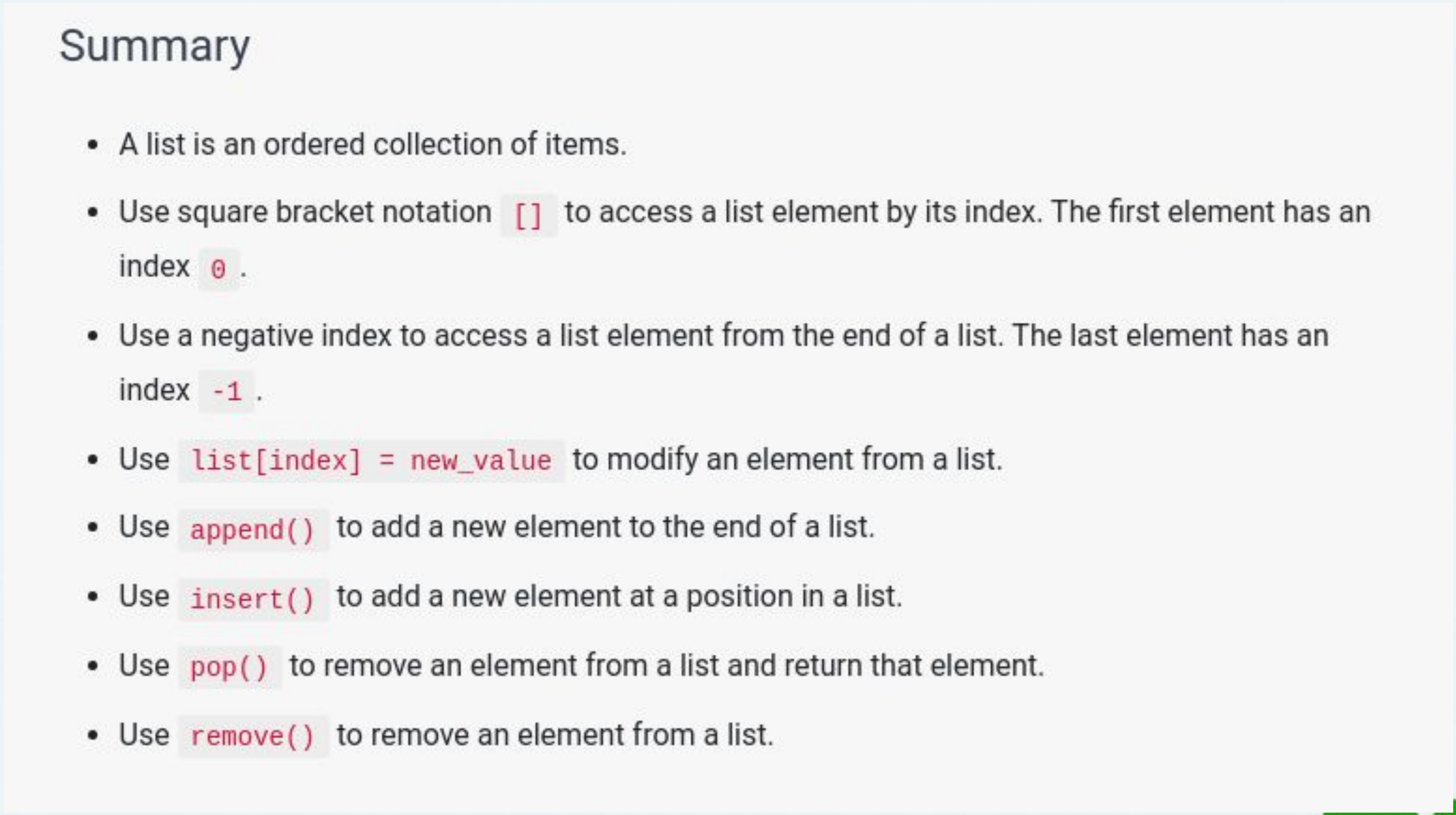
```
numbers = [1, 3, 2, 7, 9, 4]
numbers[0] = 10

print(numbers)
```

Output:

```
[10, 3, 2, 7, 9, 4]
```

Python

- Lists

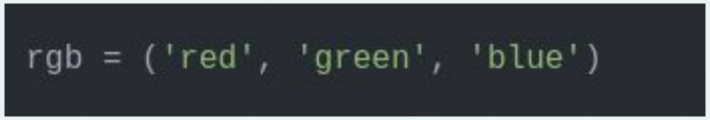


Summary

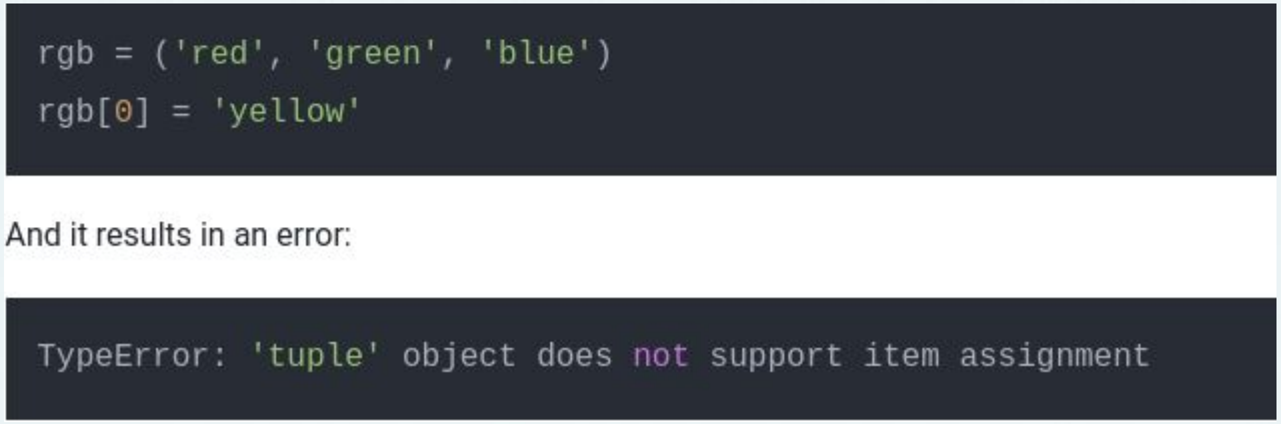
- A list is an ordered collection of items.
- Use square bracket notation `[]` to access a list element by its index. The first element has an index `0`.
- Use a negative index to access a list element from the end of a list. The last element has an index `-1`.
- Use `list[index] = new_value` to modify an element from a list.
- Use `append()` to add a new element to the end of a list.
- Use `insert()` to add a new element at a position in a list.
- Use `pop()` to remove an element from a list and return that element.
- Use `remove()` to remove an element from a list.

Python

- Tuple



```
rgb = ('red', 'green', 'blue')
```



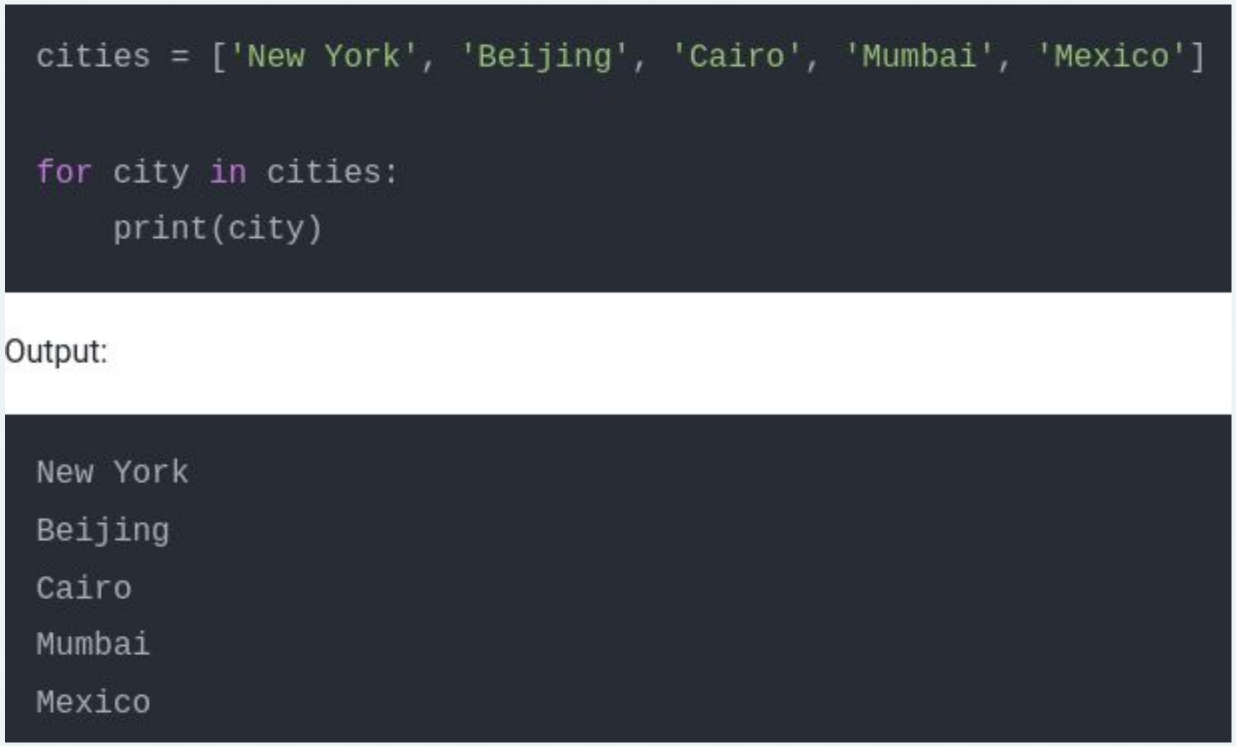
```
rgb = ('red', 'green', 'blue')
rgb[0] = 'yellow'
```

And it results in an error:

```
TypeError: 'tuple' object does not support item assignment
```

Python

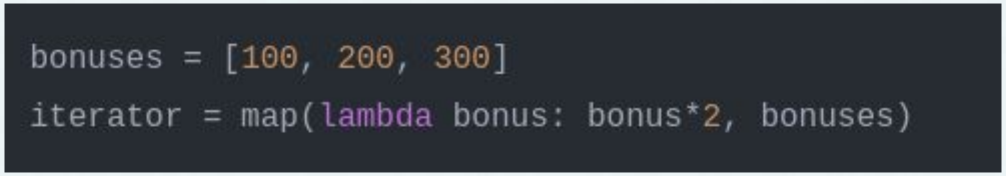
- Iterate over list



```
cities = ['New York', 'Beijing', 'Cairo', 'Mumbai', 'Mexico']

for city in cities:
    print(city)
```

Output:

```
New York
Beijing
Cairo
Mumbai
Mexico
```



```
bonuses = [100, 200, 300]
iterator = map(lambda bonus: bonus*2, bonuses)
```

Python

- List comprehension

```
[output_expression for element in list]
```

```
squares = [number**2 for number in numbers]
```

```
[output_expression for element in list if condition]
```

```
mountains = [
    ['Makalu', 8485],
    ['Lhotse', 8516],
    ['Kanchendzonga', 8586],
    ['K2', 8611],
    ['Everest', 8848]
]

highest_mountains = [m for m in mountains if m[1] > 8600]
```

Python

- Dictionaries

```python
empty_dict = {}
```

```python
person = {
    'first_name': 'John',
    'last_name': 'Doe',
    'age': 25,
    'favorite_colors': ['blue', 'green'],
    'active': True
}
```

```python
dict[key] = new_value
```

```python
del dict[key]
```

```python
for key in person.keys():
    print(key)
```

```python
for value in person.values():
    print(value)
```

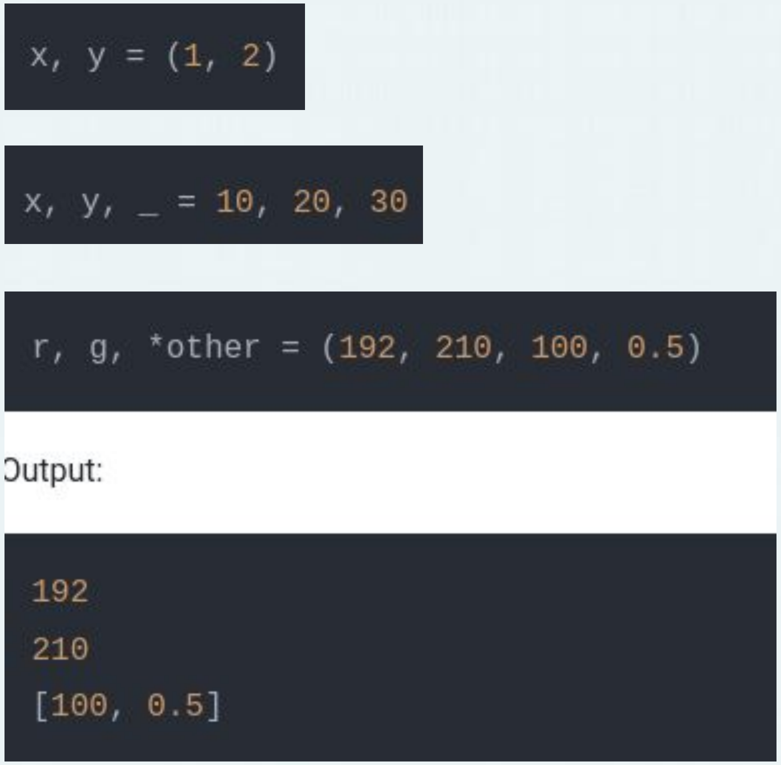
```python
for key, value in person.items():
    print(f"{key}: {value}")
```

utput:

```
first_name: John
last_name: Doe
age: 25
favorite_colors: ['blue', 'green']
active: True
```

Python

- Unpack tuples/lists



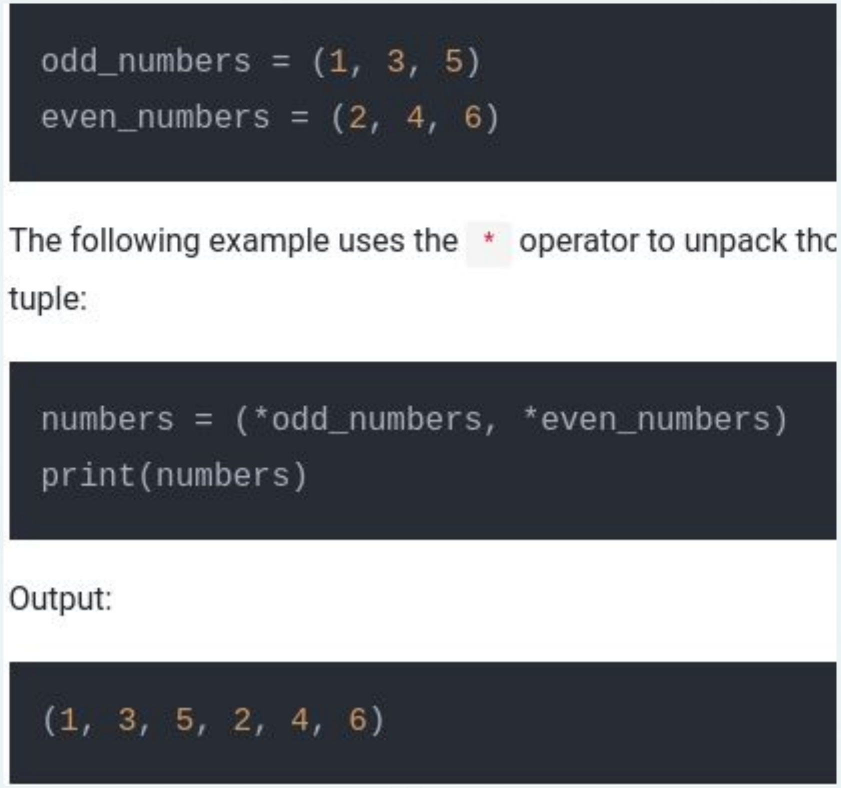
```
x, y = (1, 2)
```

```
x, y, _ = 10, 20, 30
```

```
r, g, *other = (192, 210, 100, 0.5)
```

Output:

```
192
210
[100, 0.5]
```



```
odd_numbers = (1, 3, 5)
even_numbers = (2, 4, 6)
```

The following example uses the `*` operator to unpack thc tuple:

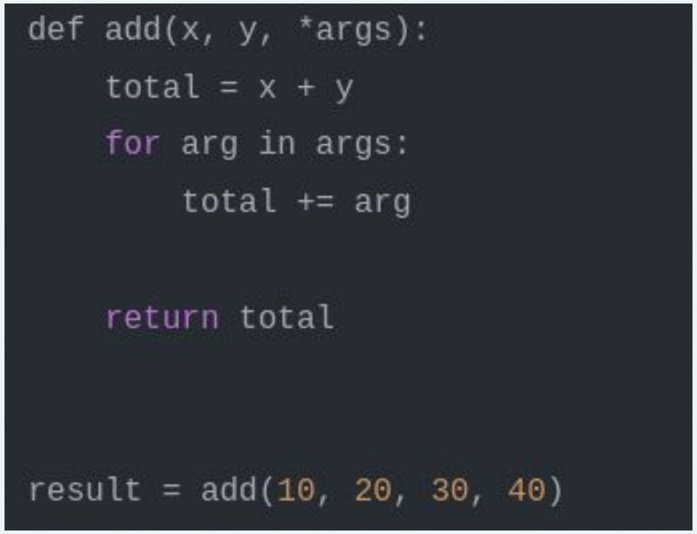
```
numbers = (*odd_numbers, *even_numbers)
print(numbers)
```

Output:

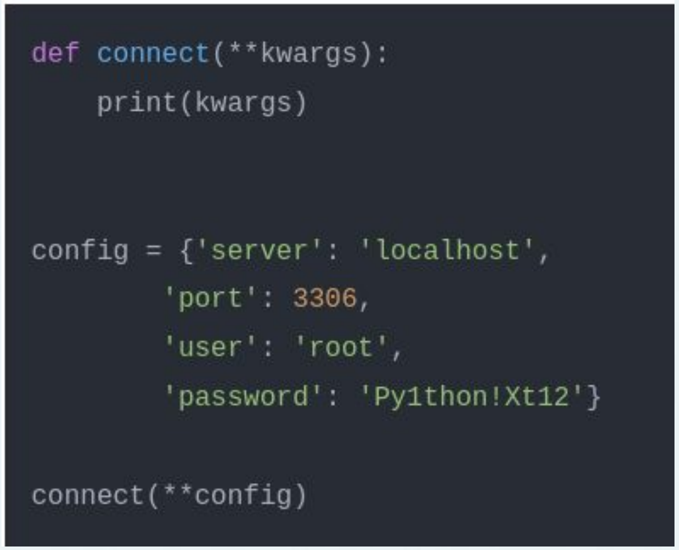
```
(1, 3, 5, 2, 4, 6)
```

Python

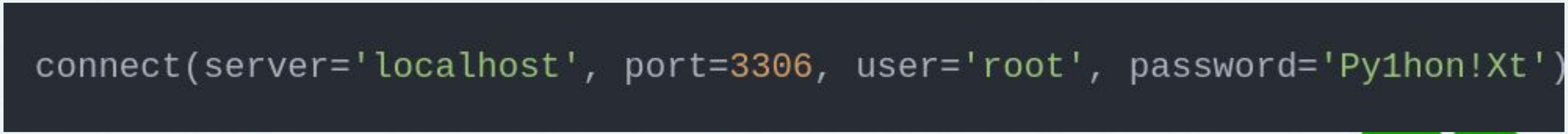
- *args/ **kwargs



```python
def add(x, y, *args):
    total = x + y
    for arg in args:
        total += arg

    return total


result = add(10, 20, 30, 40)
```



```python
def connect(**kwargs):
    print(kwargs)


config = {'server': 'localhost',
          'port': 3306,
          'user': 'root',
          'password': 'Py1thon!Xt12'}

connect(**config)
```



```python
connect(server='localhost', port=3306, user='root', password='Py1hon!Xt')
```

Feladat

Programmers then:
Programmers now:
I just coded for Apollo mission with 50KB storage.
My code womt compile because of imdentation error :(
Looking at you python !

Ajánlott irodalom

- https://www.pythontutorial.net/python-basics/
- https://www.notion.so/zarkom/Introduction-to-Git-ac396a06977704709a12b6a0e545db049
- https://realpython.com/jupyter-notebook-introduction/
- https://realpython.com/python-modules-packages/
- https://www.programmingcube.com/how-to-create-a-remote-git-repository-from-a-local-one/