

Introduction to programming

Python for beginners

Piotr Kardaś, 2021

Piotr Kardaś

Software Engineer

- 3 years of professional experience
- AirHelper since 2019
- (almost) Master in Computer Science
- Interests: Machine Learning, Programming books, Gym



Plan for today

Syllabus:

- functions
- basic data types
- objects and methods
- collections
- loops



Plan

1. Setting up Google Colab
2. 4 Tasks
3. Break
4. 4 Tasks
5. Further learning
6. Q&A



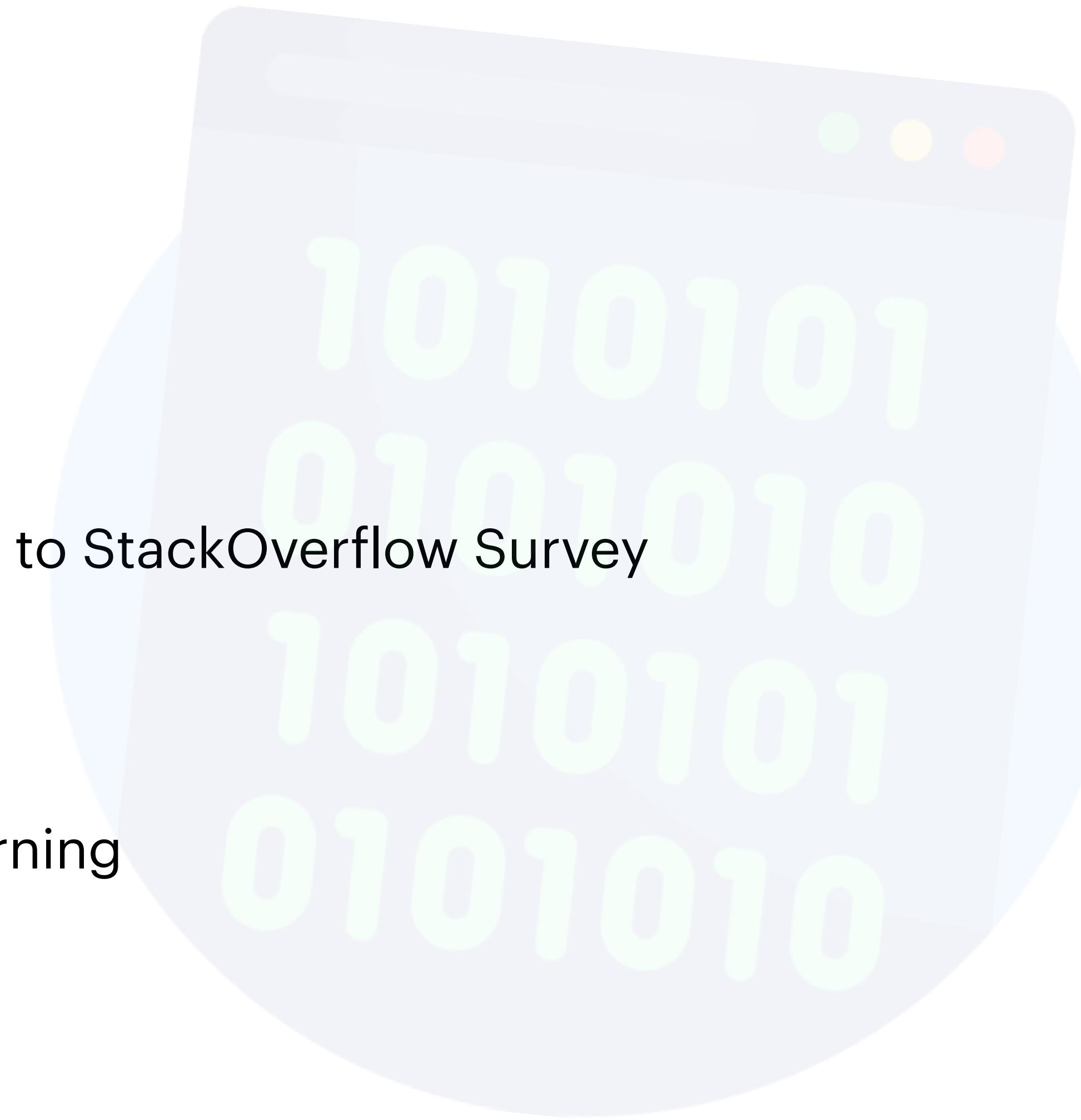
Why programming?

- 1 lots of fun and complicated problems
- 2 the joy of creation
- 3 helps the brain develop



Why Python?

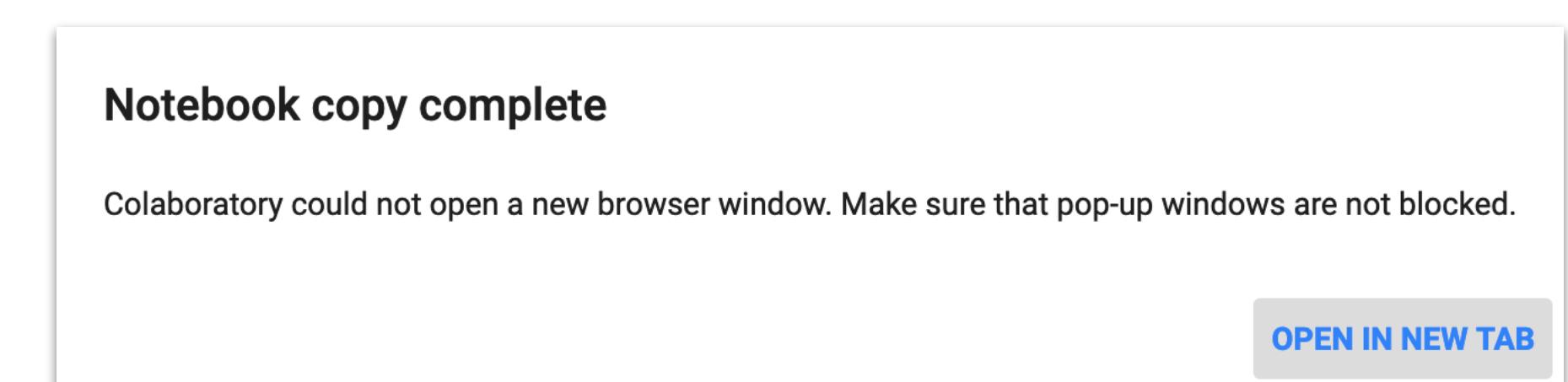
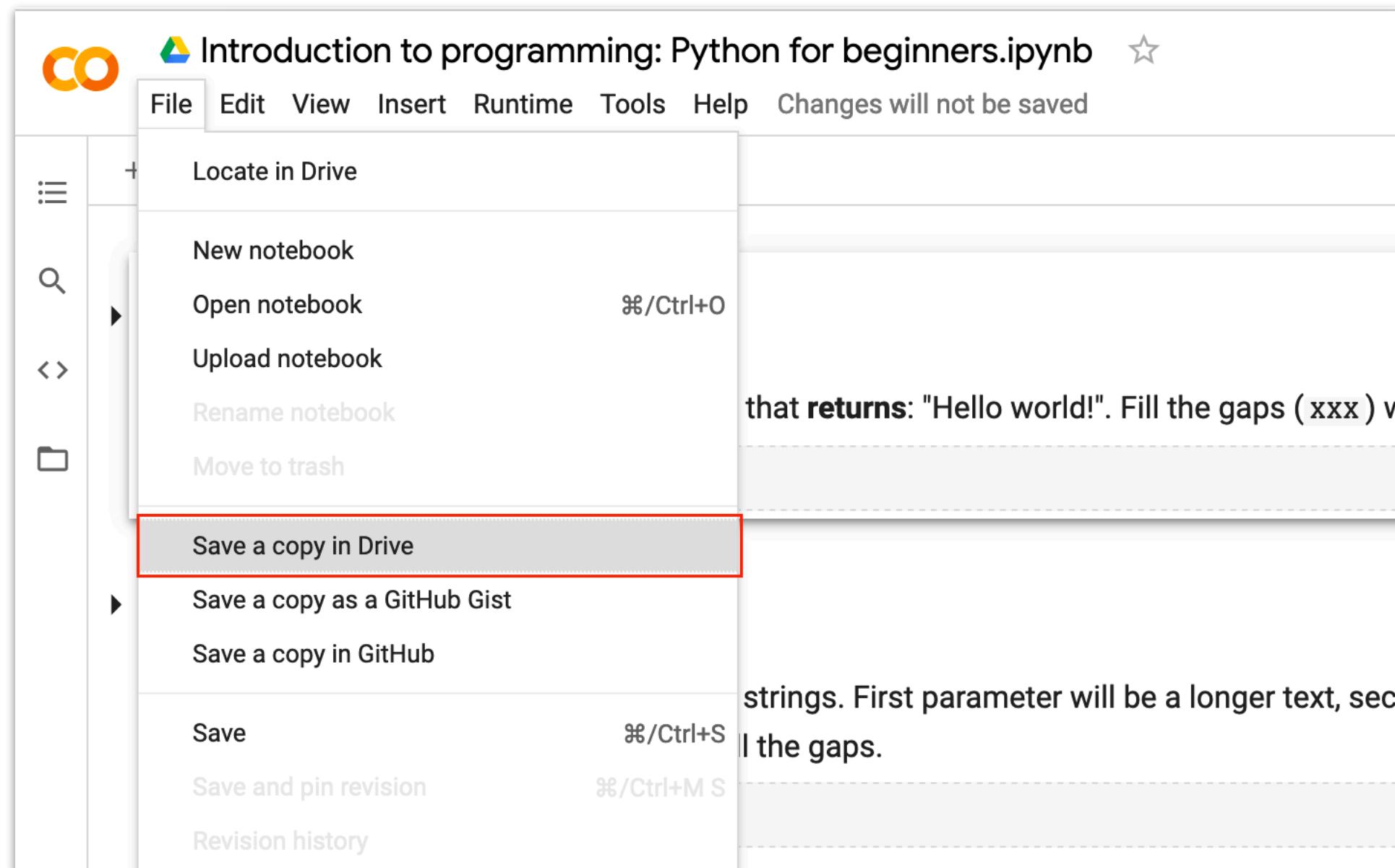
- 👉 Super-readable, English-like, great for beginners
 - 👉 Very popular in Machine Learning
 - 👉 Also used in backend applications
-
- 🥉 - among most loved programming language according to StackOverflow Survey
 - 🥉 - among most popular programming languages
 - 🥇 - most popular programming language in Machine Learning



colabo

Open Google Colab

https://colab.research.google.com/drive/1_R4nMujfly3bukp3nBgDVdikTOKycZGh?usp=sharing
(will be sent on the chat)





TASK 1

What is a function?

```
def foo() -> str:  
    return "bar"
```

```
foo()
```

```
def leaving_house() -> None:  
    """  
    Close / turn off everything after leaving the house.  
    """  
  
    main_door.close()  
  
    for light in all_lights:  
        light.turn_off()  
  
    for window in all_windows:  
        window.close()  
  
    return None
```

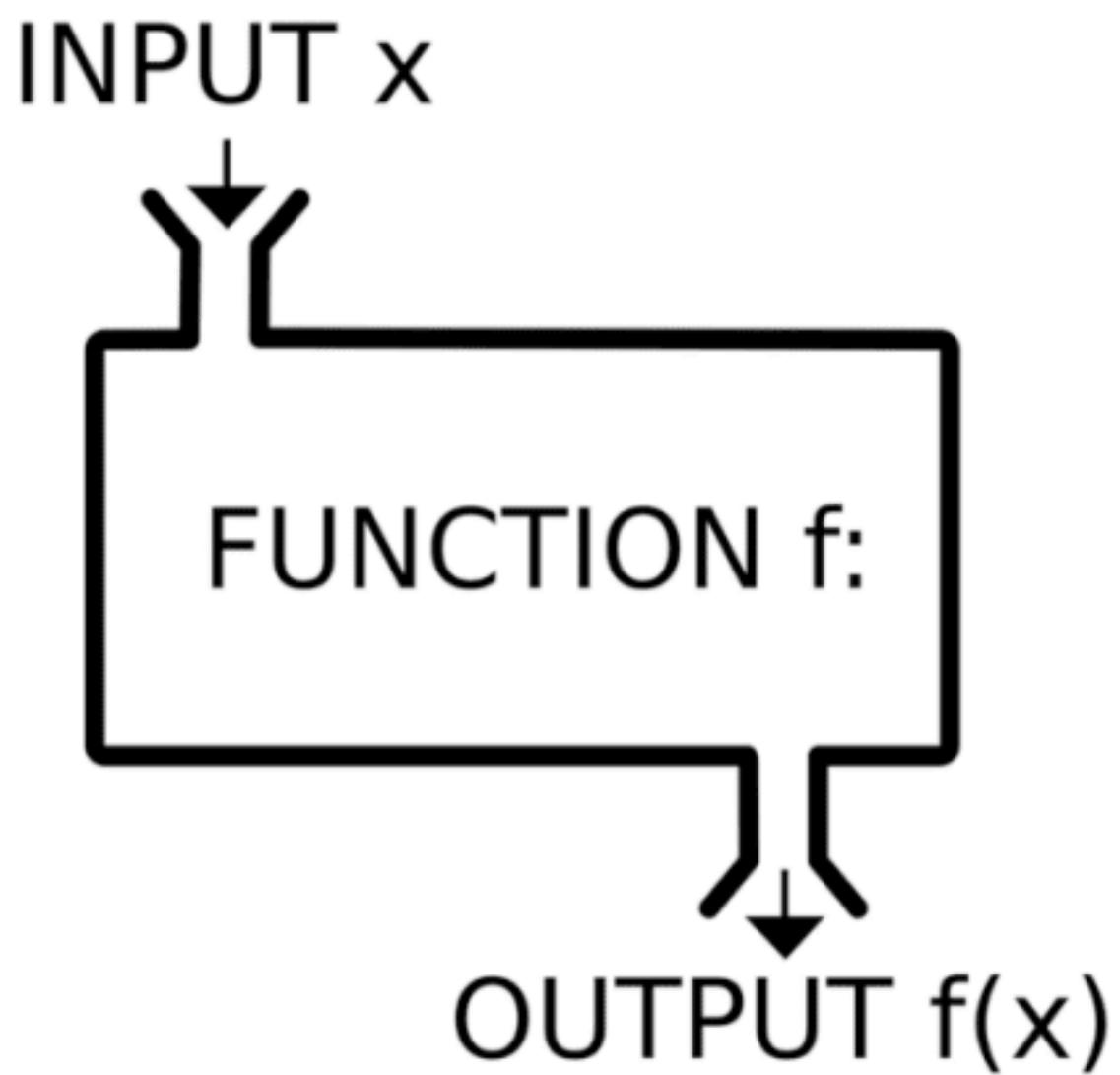


Experience



TASK 2

More advanced functions



$$f(x) = 2x$$

—

$$f(10) = 20$$

$$f(20) = 40$$

$$g(x) = x^2$$

—

$$g(10) = 100$$

$$g(20) = 400$$

$$h(x, y) = x^2 + y^2$$

—

$$h(1, 1) = 2$$

$$h(1, 5) = 5$$

```
def release_vacuum(vacuum: Vacuum) -> float:  
    """  
    Releases Roomba and returns estimated work time.  
    """  
  
    vacuum.turn_on()  
  
    return vacuum.room.area / vacuum.efficiency  
  
release_vacuum(vacuum)
```



f-string

f"Hello, {name} 🙌"

f"Claim {claim.id} is in {claim.state} state"

f"Your account balance is: {account.balance}"

Experience



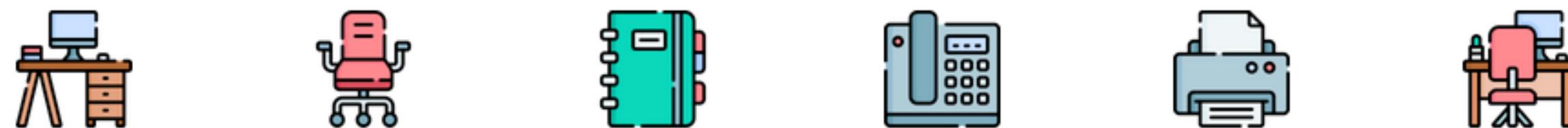
TASK 3

What is an object?

Object: Food



Object: Furniture



Object: Animal



```
class Vacuum:  
    def __init__(self, room: Room) -> None:  
        self.room = room
```

```
vacuum0 = Vacuum(bedroom)  
vacuum0.room
```

```
vacuum1 = Vacuum(living_room)  
vacuum1.room
```



Experience



TASK 4

You don't have to write everything



NumPy



django



```
$ pip install flask  
$ pip install tensorflow  
$ pip install numpy  
$ pip install langdetect  
$ pip install pydantic  
$ pip install pytest  
$ pip install mypy
```



```
class Vacuum:  
    def __init__(self, room: Room) -> None:  
        self.room = room
```

```
    def turn_on(self) -> None:  
        """
```

**Communicates with Vacuum Cleaner through WiFi
and turns on the device.**

```
        """
```

```
    ...
```

```
vacuum = Vacuum(bedroom)  
vacuum.turn_on()
```

Experience

break



TASK 5

There is a way to group objects

```
all_lights = [Light(bedroom), Light(living_room), Light(garden)]  
  
for light in all_lights:  
    light.turn_off()
```

—

```
numbers = [1, 1, 2, 3, 5, 8, 13]  
  
room_names = ["Living Room", "Bedroom", "Garden"]
```

It is possible to access specific cells

```
room_names = ["Living Room", "Bedroom", "Garden"]
```

```
room_names[0]  
# Living Room
```

```
room_names[1]  
# Bedroom
```

```
room_names[2]  
# Garden
```

```
room_names[2021]  
# ERROR!!!
```

```
room_names[-1]  
# Garden
```

Looping through list

```
room_names = ["Living Room", "Bedroom", "Garden"]

for room_name in room_names:
    print(room_name)

# Living Room
# Bedroom
# Garden
```

Experience



TASK 6

Looping through list

```
# Get rooms with name longer than 6 characters
```

```
# Awful 🤢:
```

```
room_names = ["Living Room", "Bedroom", "Garden"]
filtered_rooms = []
for room_name in room_names:
    if len(room_name) > 6:
        filtered_rooms.append(room_name)
```

```
# Nice ✌:
```

```
room_names = ["Living Room", "Bedroom", "Garden"]
filtered_rooms = [room_name for room_name in room_names if len(room_name) > 6]
```

Experience



TASK 7

More collections

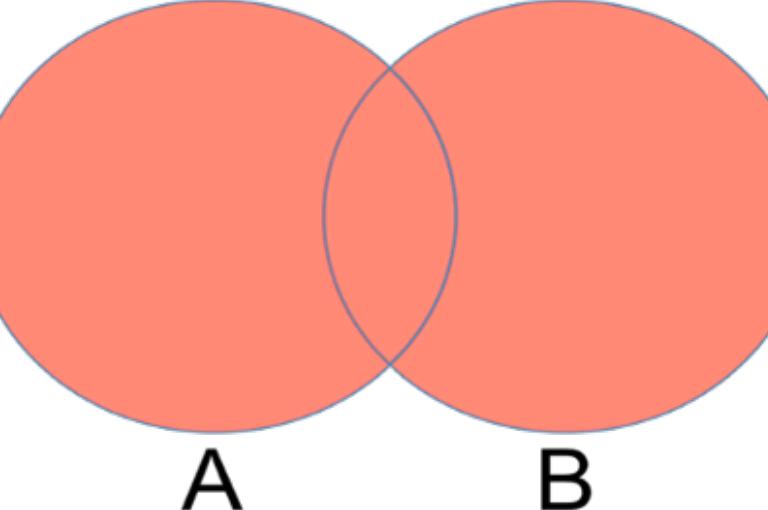
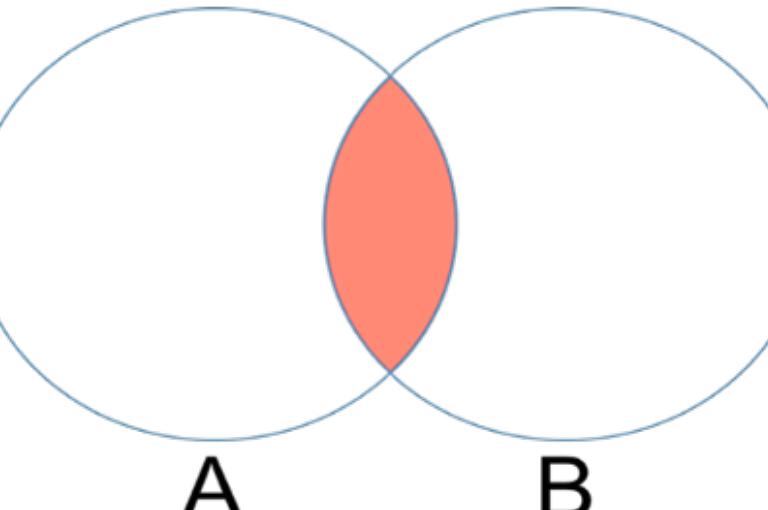
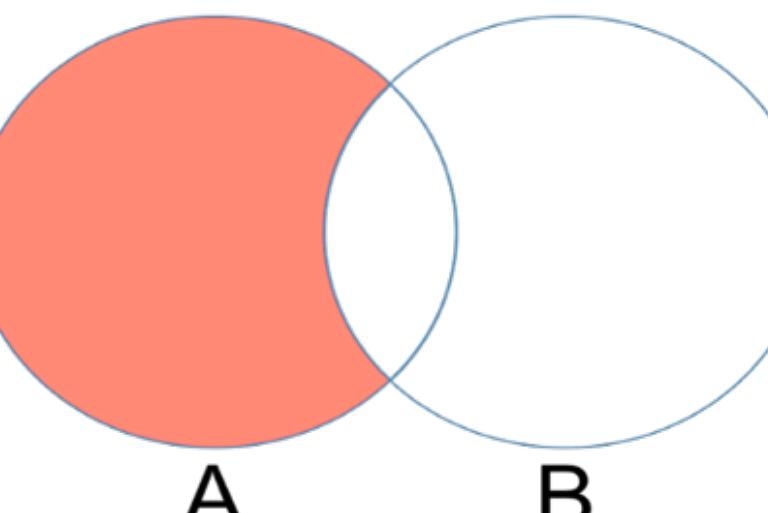
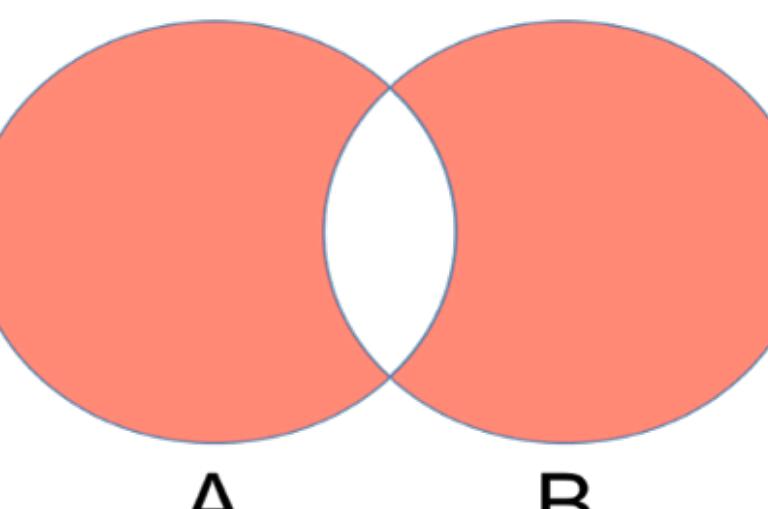
```
room_names = ["Living Room", "Bedroom", "Bedroom", "Bedroom", "Kitchen"]  
# Living Room, Bedroom, Bedroom, Bedroom, Garden
```

```
room_names = ("Living Room", "Bedroom", "Bedroom", "Bedroom", "Kitchen")  
# Living Room, Bedroom, Bedroom, Bedroom, Garden
```

```
room_names = {"Living Room", "Bedroom", "Bedroom", "Bedroom", "Kitchen"}  
# Living Room, Bedroom, Garden
```

```
rooms = {"Bedroom": Room(...), "Living Room": LivingRoom(...)}  
rooms["Bedroom"]  
# Room(...)
```

Theory

Set Operation	Venn Diagram	Interpretation
Union		$A \cup B$, is the set of all values that are a member of A , or B , or both.
Intersection		$A \cap B$, is the set of all values that are members of both A and B .
Difference		$A \setminus B$, is the set of all values of A that are not members of B
Symmetric Difference		$A \triangle B$, is the set of all values which are in one of the sets, but not both.

More on collections

```
rooms = ["Kitchen", "Bedroom", "Living Room"]
# Replace Kitchen with Garden
rooms[0] = "Garden"
```

```
rooms = ("Kitchen", "Bedroom", "Living Room")
# Replace Kitchen with Garden
rooms[0] = "Garden"
# ERROR!!! Not possible!
```

```
"Bedroom" in {"Living Room", "Bedroom", "Garden"}
# True
```

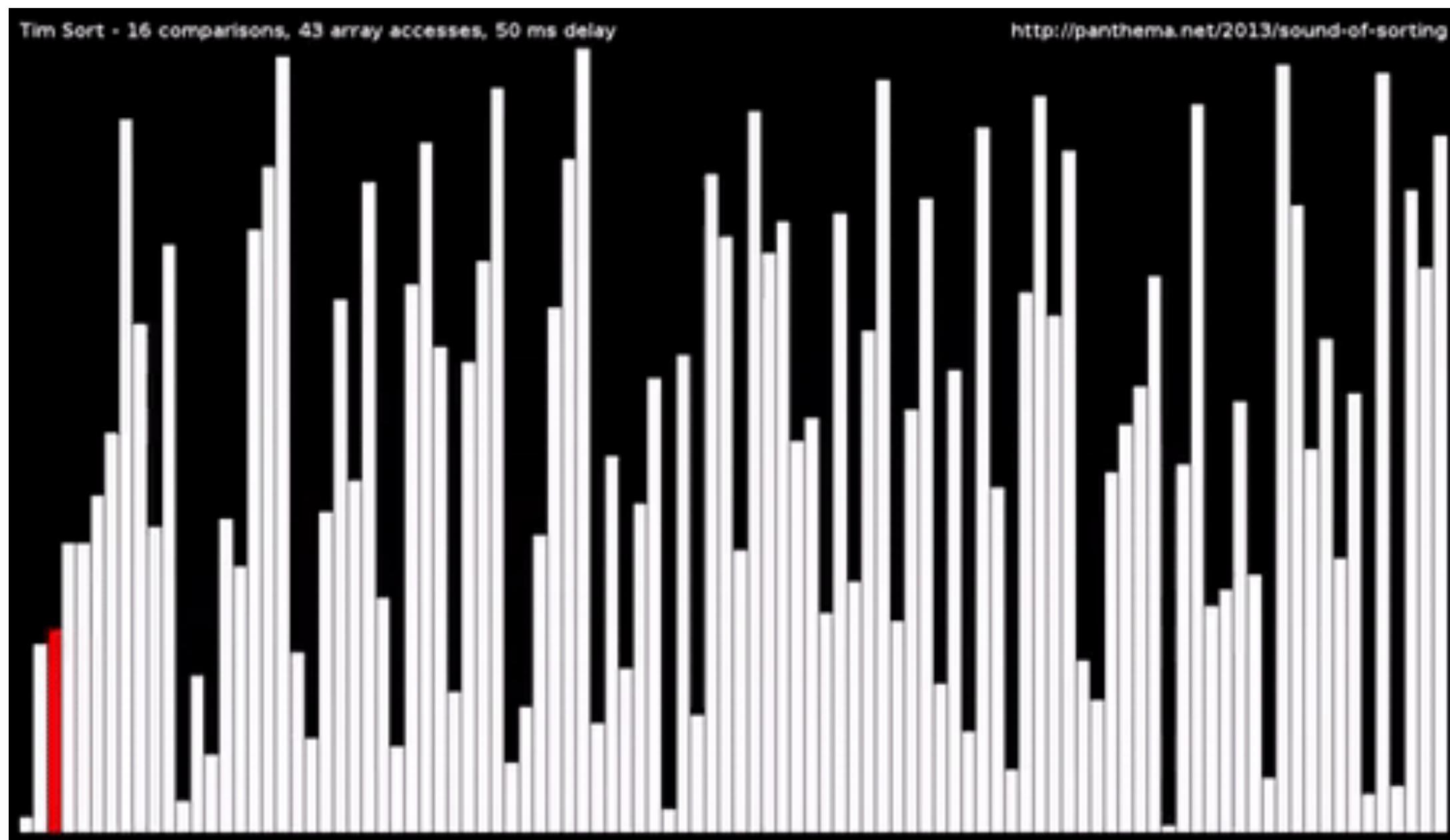
```
temperatures = {"Bedroom": 23, "Living Room": 22, "Kitchen": 24}
# What is the temperature in the Kitchen?
temperatures["Kitchen"]
# 24
```

Experience



TASK 8

Sorting



Sorting

```
temperatures = {"Bedroom": 23, "Living Room": 22, "Kitchen": 24}
```

```
warmest_room = sorted(temperatures, key=temperatures.get)[-1]
```

```
warmest_room  
# Kitchen
```

Experience

That's all.

Next steps



- 1 Install PyCham
- 2 Think about some simple problem
- 3 Solve your problem with Python application

If don't have any idea for your own project



HackerRank

Say "Hello, World!" With Python

Easy, Max Score: 5, Success Rate: 97.59%

Get started with Python by printing to stdout.

[Solve Challenge](#)

Python If-Else

Easy, Python (Basic), Max Score: 10, Success Rate: 91.27%

[Solve Challenge](#)

Arithmetic Operators

Easy, Python (Basic), Max Score: 10, Success Rate: 98.37%

[Solve Challenge](#)

Python: Division

Easy, Python (Basic), Max Score: 10, Success Rate: 98.86%

[Solve Challenge](#)

Stand out from the crowd

Take the HackerRank Skills Certification Test and make your profile stand out to peers

[Get Certified](#)

Q & A

Submit feedback

<https://forms.gle/z6D4CvrtxA4boYiU9>