Week 1: Python Fundamentals

# Unit 1: First Steps in Python

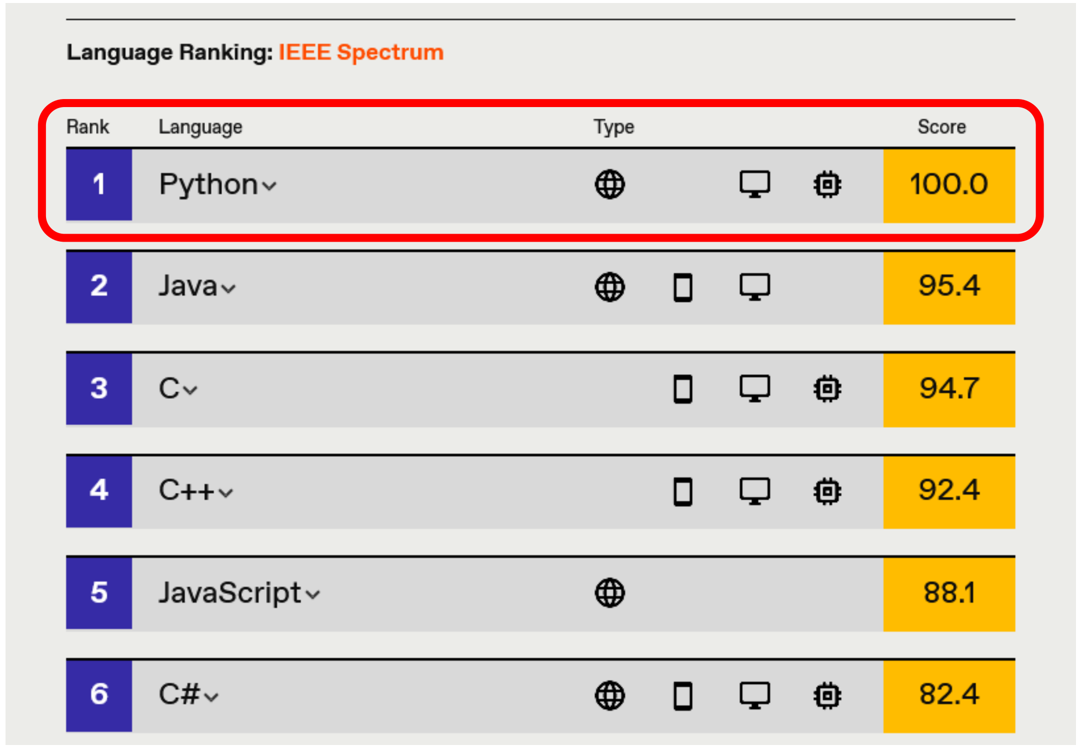
hosted by openSAP
open.sap.com

FH AACHEN
UNIVERSITY OF APPLIED SCIENCES

First steps in Python

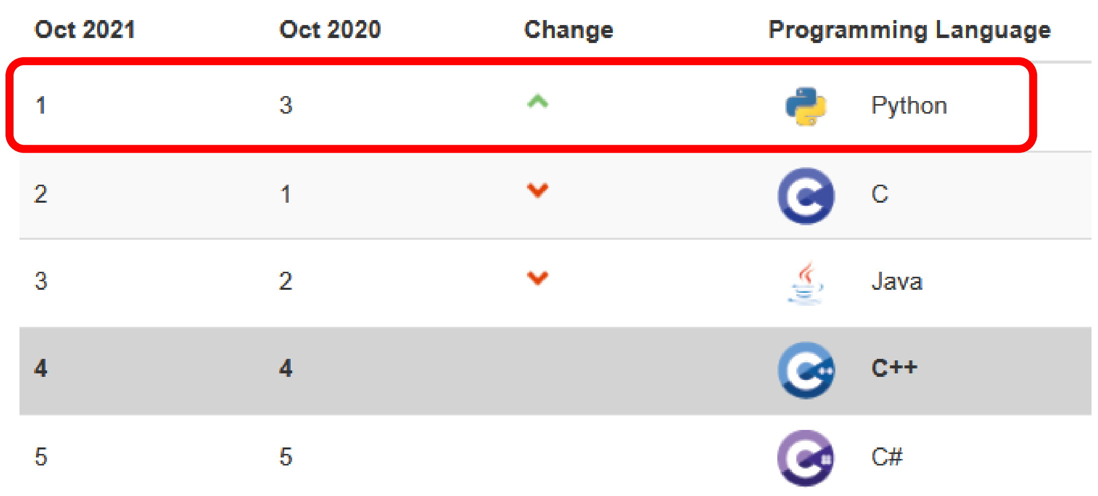
## Why Python? Python is popular!

- 1st place on IEEE popularity index
- 1st place in TIOBE index
- Popularity means:
  - As a learner, you'll find lot of tutorials, learning material, examples, and tips & tricks in the Internet.
  - As a programmer, you'll find many libraries and tools, which make your programming life easier.
  - If you run into problems, you'll quickly find answers in Google, YouTube, and other platforms.



**Language Ranking:** IEEE Spectrum

| Rank | Language | Type | Score |
|---|---|---|---|
| 1 | Python | 🌐 🖥 ⚙ | 100.0 |
| 2 | Java | 🌐 📱 🖥 | 95.4 |
| 3 | C | 📱 🖥 ⚙ | 94.7 |
| 4 | C++ | 📱 🖥 ⚙ | 92.4 |
| 5 | JavaScript | 🌐 | 88.1 |
| 6 | C# | 🌐 📱 🖥 ⚙ | 82.4 |

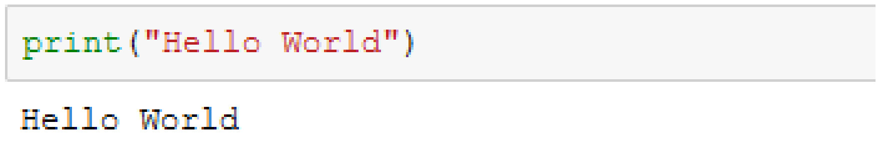


| Oct 2021 | Oct 2020 | Change | Programming Language |
|---|---|---|---|
| 1 | 3 | ▲ | Python |
| 2 | 1 | ▼ | C |
| 3 | 2 | ▼ | Java |
| 4 | 4 | | C++ |
| 5 | 5 | | C# |

First steps in Python

## Why Python? Python is easy to learn!

- In comparison to other programming languages, Python is easy to learn.
  - The program on the right is already a complete program and can be executed on your computer.
  - Python's design emphasizes *readability* e.g. by using just a few keywords or by forcing indentation.
- This simplicity is reflected in the guiding principles for the development of Python known as the *Zen of Python*.
  - Have a look for these principles in Wikipedia.



```
print("Hello World")
```

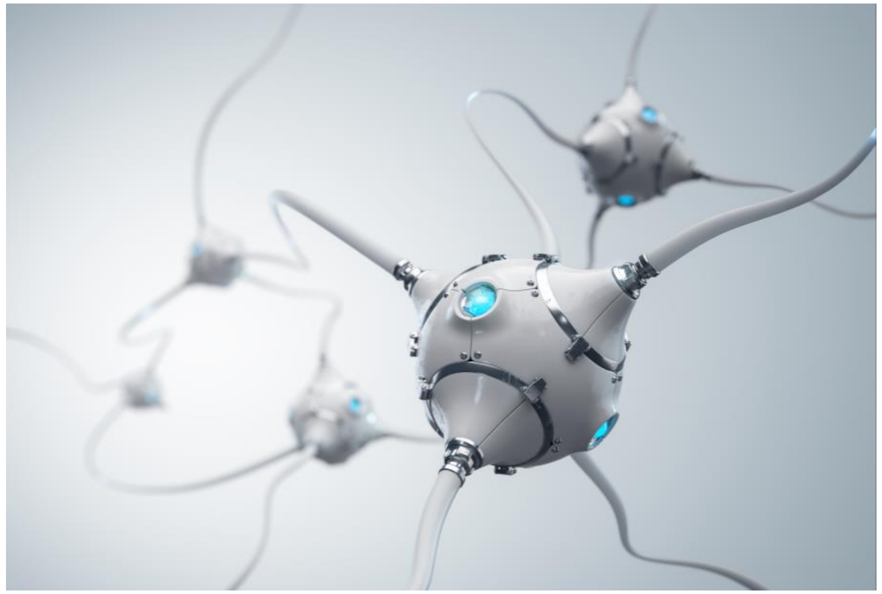
```
Hello World
```

First steps in Python

## Why Python? Python is used in industry!

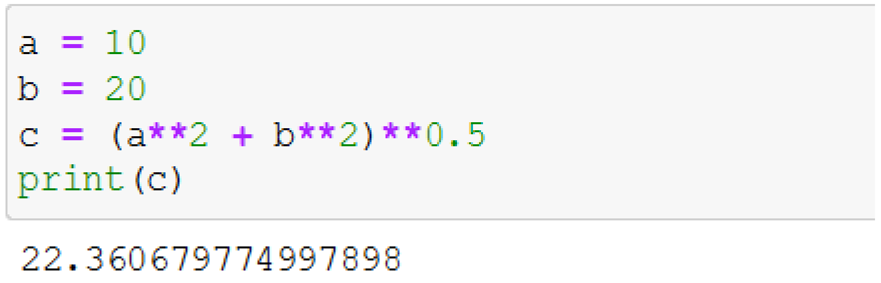
- Although Python is easy to learn and easy to use, it is very powerful.
- Thus, it is used not only in beginners' courses in programming but also in real projects in industry.

- Python supports programming paradigms like *object orientation* or *functional programming* and can thus be used for different scenarios.
- Especially in the field of artificial intelligence and machine learning, Python is dominant.



First steps in Python
## What is a program?

- All Python programs consist of *statements* or *instructions*.
- These statements are typically written one below the other and are executed one by one.
  - (The sequence of execution can be changed by *control structures*, which we will talk about later.)
- A statement in a Python program can be for example a mathematical expression like
  - 5 + 3
  - 123 + 234
  - The figure on the right shows a program consisting of four statements. The meaning of each statement is not of interest right now.



```
a = 10
b = 20
c = (a**2 + b**2)**0.5
print(c)
```

```
22.360679774997898
```

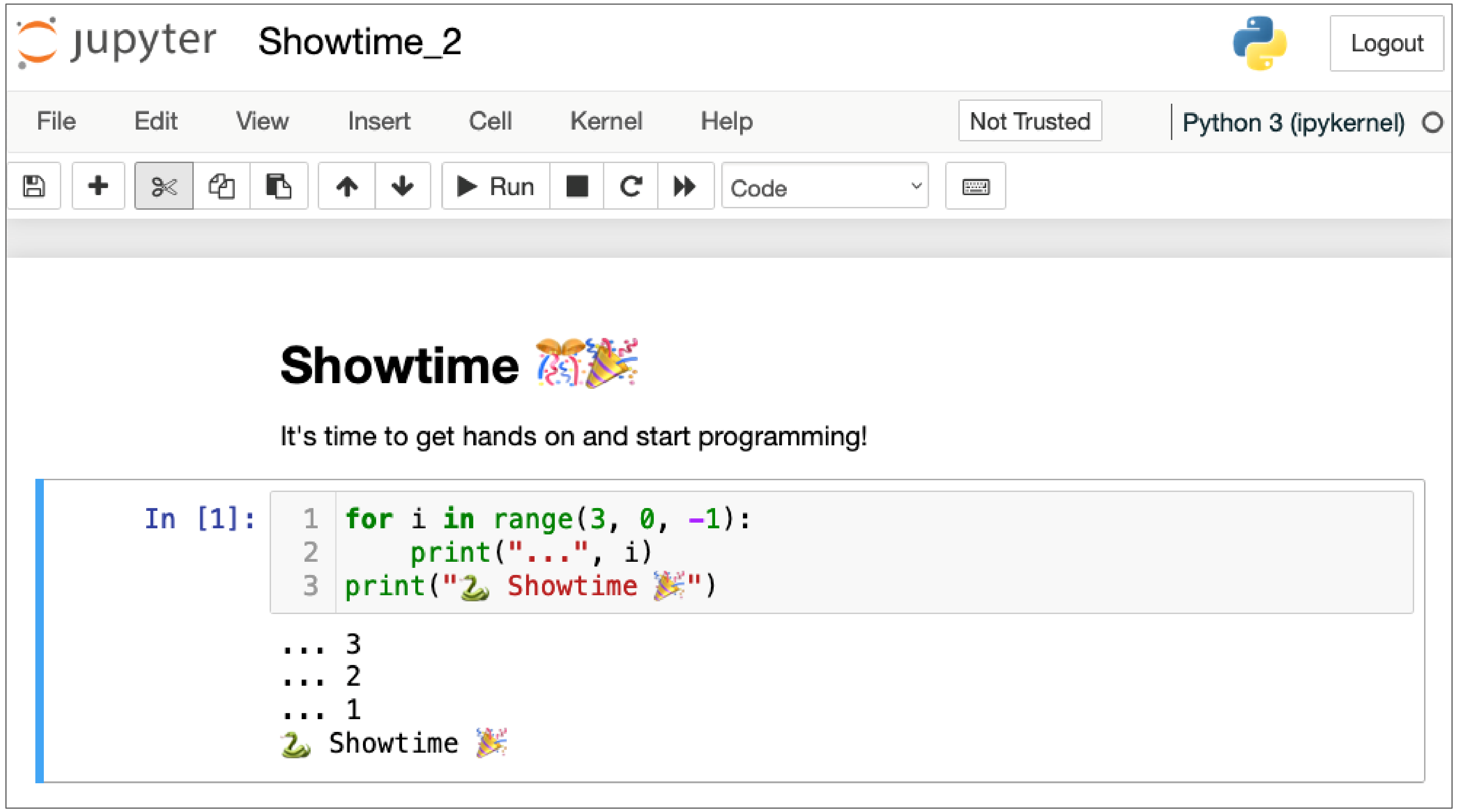
First steps in Python
## Showtime

Now it’s time to get hands on and start programming!

If you like, you can open the Jupyter Notebook instructions in parallel to the demo.

If you haven’t done so yet:

- Download the Notebook
- Start the Jupyter Server
- Open the Notebook



# Showtime 🎊🎉

It's time to get hands on and start programming!

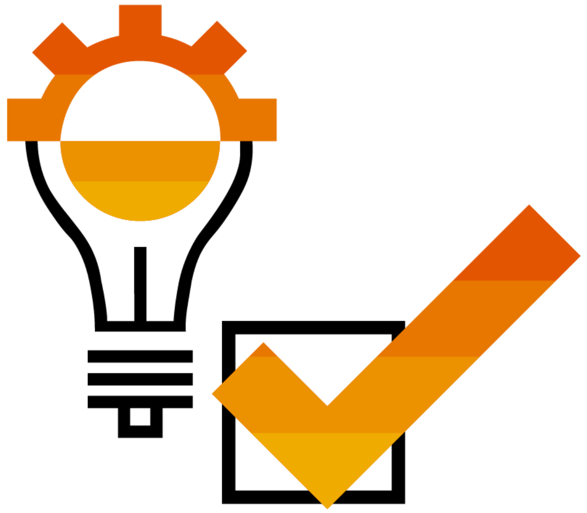
In [1]:
```
for i in range(3, 0, -1):
    print("...", i)
print("🐍 Showtime 🎉")
```

```
... 3
... 2
... 1
🐍 Showtime 🎉
```

First steps in Python
## Summary / key takeaways

In this unit you learned ...

- … that Python is popular, easy to learn, and nevertheless used in industry
- … that programs consist of statements, which are executed one by one



## Thank You!

**Contact Information:**
open@sap.com

hosted by openSAP
open.sap.com

FH AACHEN
UNIVERSITY OF APPLIED SCIENCES

Week 1: Python Fundamentals

## Unit 2: Using Variables

Using variables

## Variables are used in almost any computer program

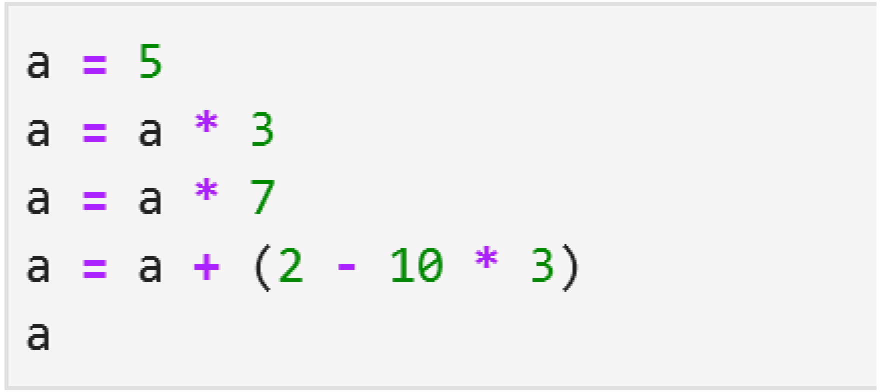
- Computer programs handle data. And like real world objects, the data must be accessible.
  - In the real world you must handle real world objects. You must place these objects somewhere, and you must be able to find and access these objects later on.
- In computer programs the data is placed in, and accessed from *variables*.
- Important:
  - A variable can store just one value at a given time.
  - If the variable is read, the value is not consumed and taken away. This is like a book: If you read it, the words are still available later on.

```
name = "David"
surname = "Bowie"
account_balance = -2000
_new_balance = 1000
```

Using variables

## Basic operations with variables

- Variables have names like `x` or `length` or `name_of_person`.
  - There are a few rules for variable names.
- Variables can handle data in two ways:
  - Data can be assigned to a variable *(write access)*
  - Data can be read from a variable *(read access)*
- Data is assigned to a variable by the `=` sign
  - Example: `x = 42`
  - The value (data) `42` on the right side of the = is assigned to `x`
- Data can be read by simply calling the variable
  - Example `x = 2 * y`
  - The current value of `y` is read, it is multiplied with `2`, and the result is assigned to variable `x`



```
a = 5
a = a * 3
a = a * 7
a = a + (2 - 10 * 3)
a
```

Using variables

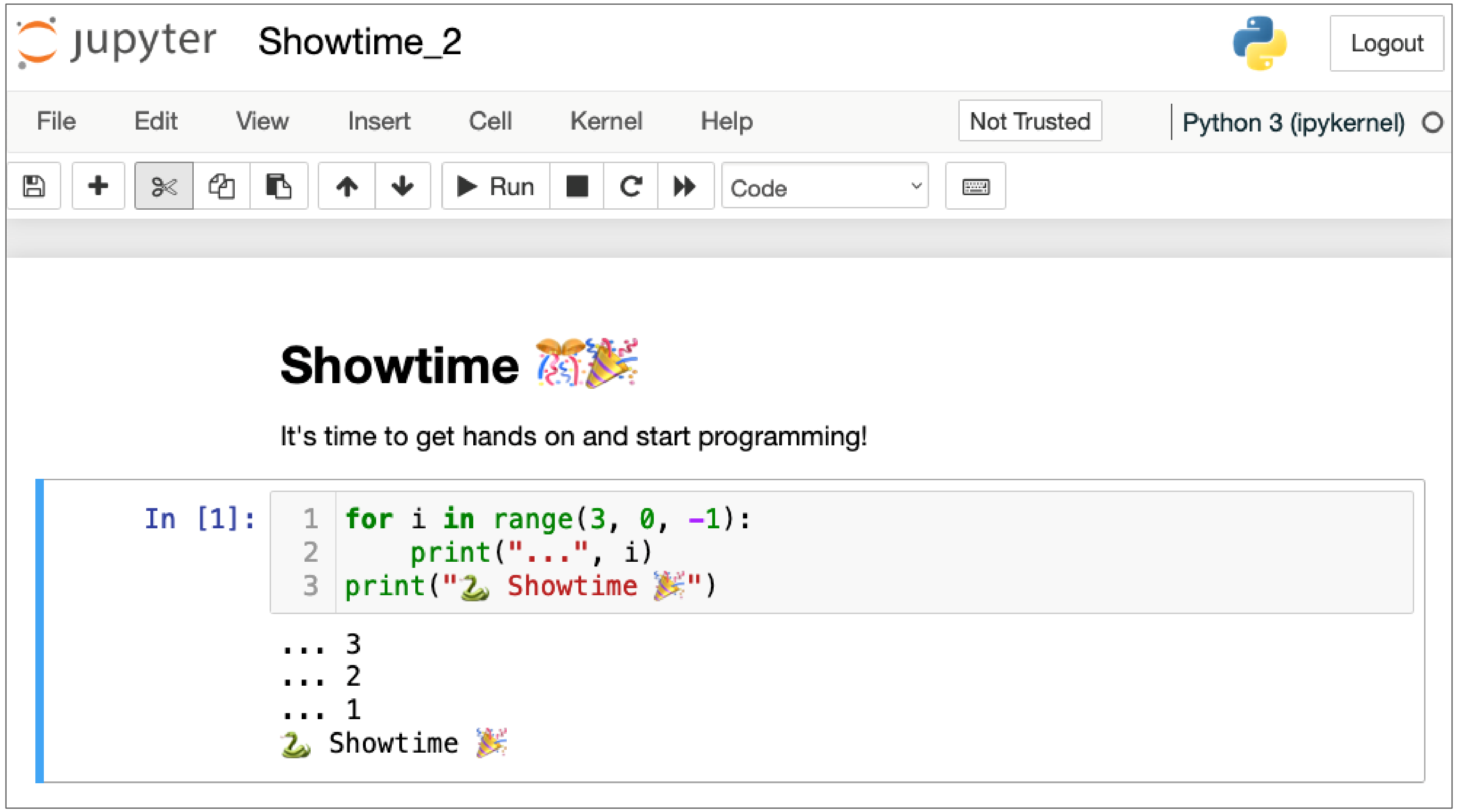
## Showtime

Now it's time to get hands on and start programming!

If you like, you can open the Jupyter Notebook instructions in parallel to the demo.

If you haven't done so yet:

- Download the Notebook
- Start the Jupyter Server
- Open the Notebook



Showtime 🎊🎉

It's time to get hands on and start programming!

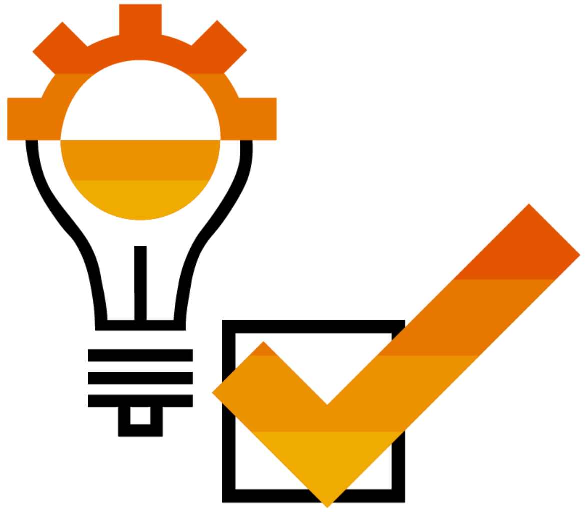
In [1]:
```
for i in range(3, 0, -1):
    print("...", i)
print("🐍 Showtime 🎉")
```

```
... 3
... 2
... 1
🐍 Showtime 🎉
```

Using variables
## Summary

In this unit you learned ...

- … that programs make use of variables
- … that data can be written into and read from variables
- … that assignments in programming and assertions in mathematics are not the same
- … that there are a few weird looking statements, which are used regularly



## Thank You!

**Contact Information:**
open@sap.com

hosted by openSAP
open.sap.com

Week 1: Python Fundamentals

## Unit 3: Performing Simple Input and Output

hosted by openSAP open.sap.com

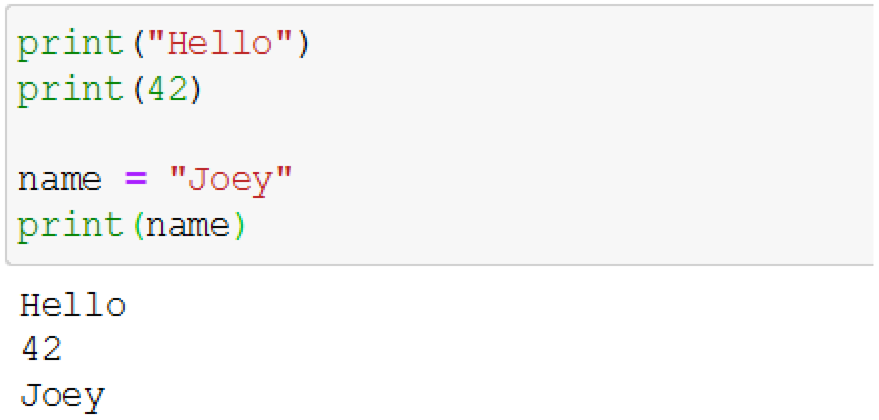
FH AACHEN UNIVERSITY OF APPLIED SCIENCES

Performing simple input and output

## Real programs must deal with flexible input and output

- So far, the output of the last statement has been printed below the cell.
- This way of working has two drawbacks:
  – Not every statement has an output.
  – You only have limited control over your output.
- A more general way to handle output is required.

- So far, the programs could only handle output. Real programs require the possibility to handle input as well.



```
print("Hello")
print(42)

name = "Joey"
print(name)
```

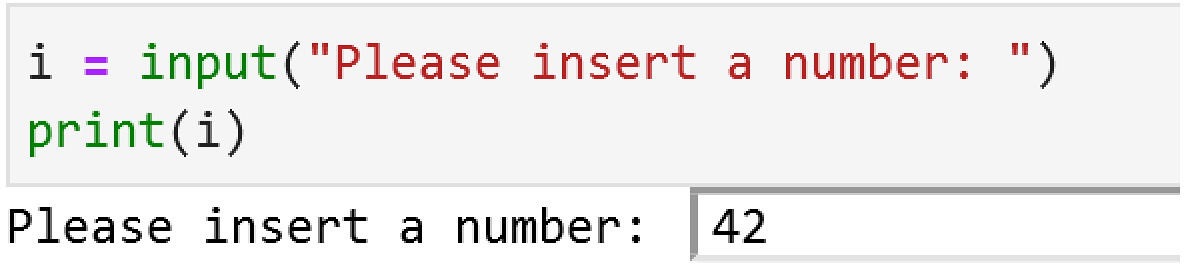
```
Hello
42
Joey
```

Performing simple input and output

## Creating output with `print()` and input with `input()`

- The function `print()` can be used to get better control of the output.
  - Not only at the end of the cell
  - Not only once per cell
  - Not only one argument

- The function `input()` enables the programmer to handle input.



```python
i = input("Please insert a number: ")
print(i)
```

```
Please insert a number:  42
```

Performing simple input and output

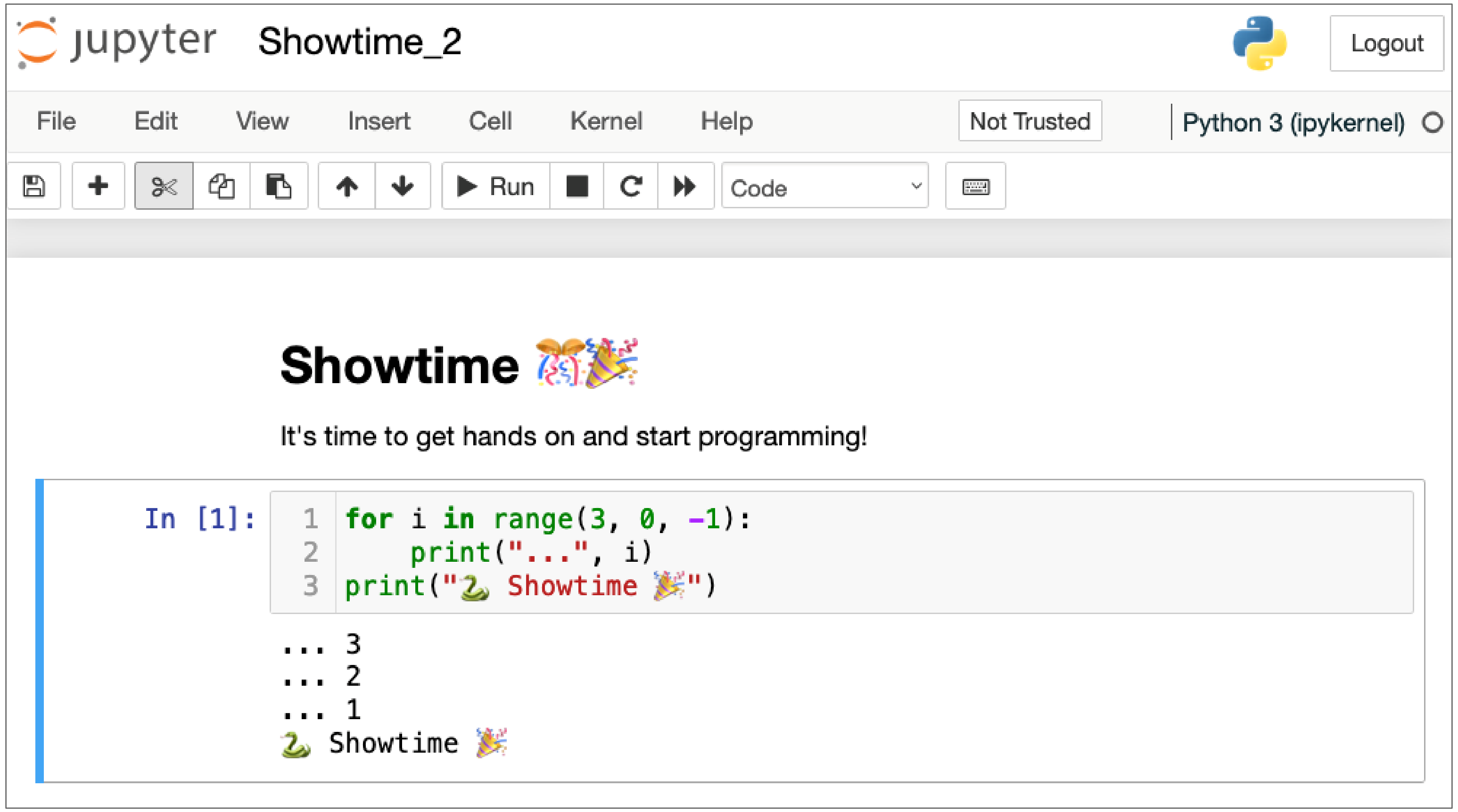
## Showtime

Now it's time to get hands on and start programming!

If you like, you can open the Jupyter Notebook instructions in parallel to the demo.

If you haven't done so yet:

- Download the Notebook
- Start the Jupyter Server
- Open the Notebook



Showtime 🎊🎉

It's time to get hands on and start programming!

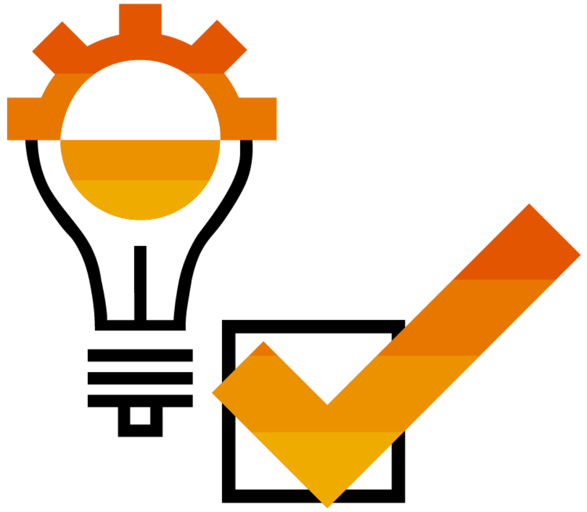
In [1]:
```
for i in range(3, 0, -1):
    print("...", i)
print("🐍 Showtime 🎉")
```

```
... 3
... 2
... 1
🐍 Showtime 🎉
```

Performing simple input and output
## Summary / key takeaways

In this unit you learned ...

- … that you can handle input with `input()`
- … that you can control the output with `print()`
- … that many programs are designed according to the IPO pattern, i.e., Input – Processing – Output.



# Thank You!

**Contact Information:**
open@sap.com

hosted by openSAP
open.sap.com

FH AACHEN UNIVERSITY OF APPLIED SCIENCES

Week 1: Python Fundamentals

## Unit 4: What Is a Data Type?

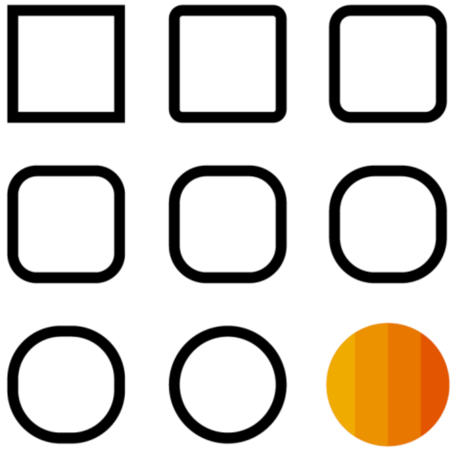
hosted by openSAP open.sap.com

FH AACHEN UNIVERSITY OF APPLIED SCIENCES

What is a data type?

## Different kinds of data

- As stated before: Programs handle data.
  However, there are different kinds of data.
  – Example: There is data like numbers `1`, `45`, `-320` and other data like text "`My name is Stephan`"
- In programming, these different kinds of data are called *data types*.
- **Important**: The way you handle data depends on its data type.
  – Example: You can multiply the number `3 * 5`
  – but you cannot multiply texts like "`My name`" * "`is Stephan`". This operation is not defined.



What is a data type?

## Python supports four different primitive data types

- Python support the following data types:
  - Integer, e.g.: `42, -100, 23, 0`
  - Float, e.g.: `2.3, -0.00012, 3.2e10`
  - String, e.g. `"This is a string", "x", "xy123"`
  - Boolean: `True, False`
- Python offers the function `type()`, to check the data type
- It is possible to convert data with casting functions

```python
print(type(42))
print(type(-3.14))
print(type(True))

s = "Hello World"
print(type(s))
```

```
<class 'int'>
<class 'float'>
<class 'bool'>
<class 'str'>
```

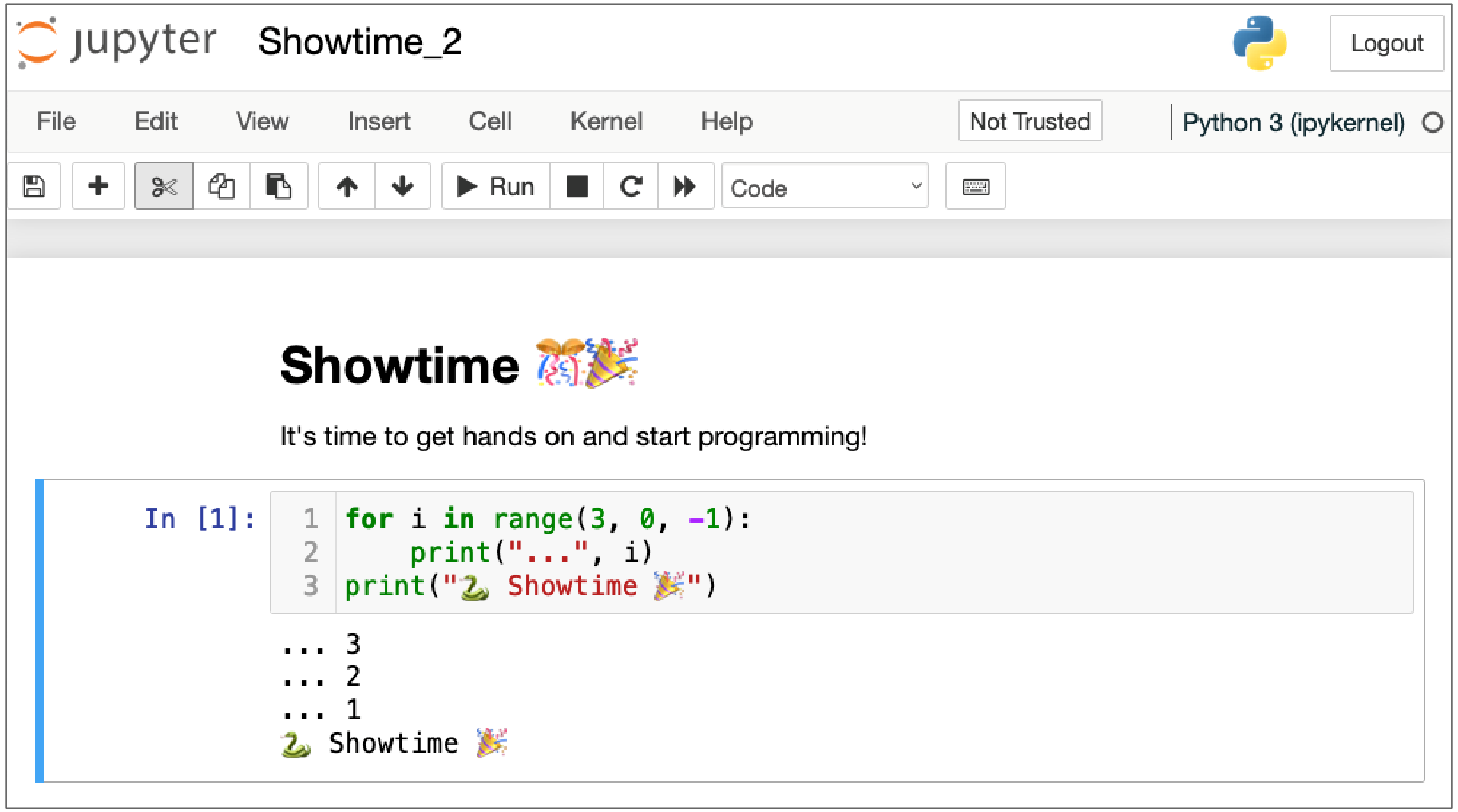
What is a data type?
## Showtime

Now it's time to get hands on and start programming!

If you like, you can open the Jupyter Notebook instructions in parallel to the demo.

If you haven't done so yet:

- Download the Notebook
- Start the Jupyter Server
- Open the Notebook



### Showtime 🎊🎉

It's time to get hands on and start programming!

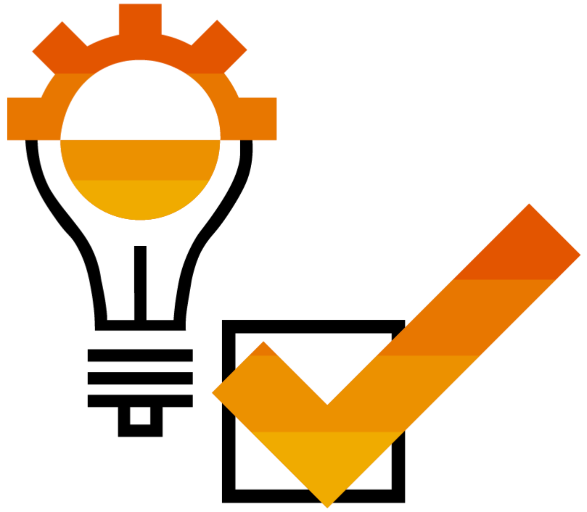
```
In [1]: for i in range(3, 0, -1):
            print("...", i)
        print("🐍 Showtime 🎉")
```

```
... 3
... 2
... 1
🐍 Showtime 🎉
```

## What is a data type?
## **Summary / key takeaways**

In this unit you learned ...

- … that there are different data types
- … that Python supports the data types integer, float, Boolean, and string
- … that operations are defined for these data types
- … that it is possible to convert (cast) these datatypes



# Thank You!

**Contact Information:**
open@sap.com

hosted by openSAP
open.sap.com

FH AACHEN
UNIVERSITY OF APPLIED SCIENCES

Week 1: Python Fundamentals

## Unit 5: Using If Statements

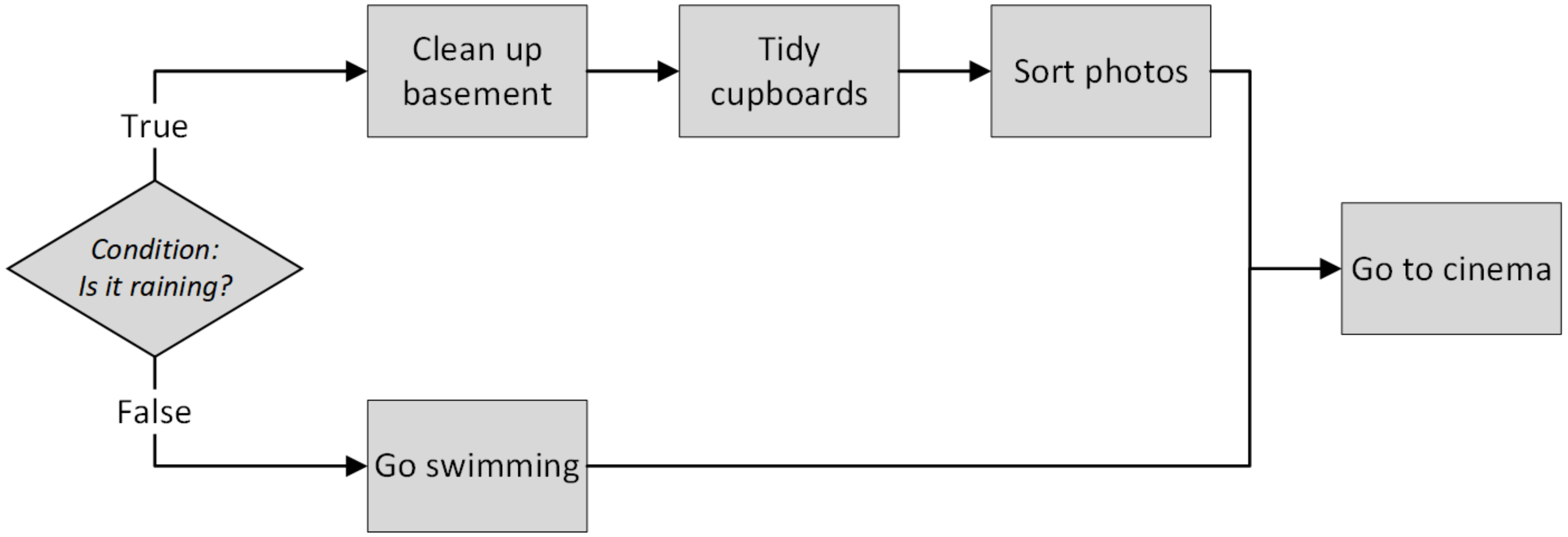
Using if statements
## Decisions in real life

*“If it is raining tomorrow, I will clean up the basement. After that, I will tidy my cupboards and sort photos. Otherwise, I will go swimming. In the evening, I am going to the cinema.”*

Meaning:

- If it is raining tomorrow, I will do as follows:
  - Clean up the basement
  - Tidy my cupboards
  - Sort photos
- Otherwise (if it is not raining):
  - Go swimming
- In the evening: go to the cinema

How to implement that in Python?



Using if statements

## If statements in Python

- Two things are required in Python to implement decisions:
  – A *control flow* to enable decisions and splitting
  – A way to formulate *conditions*
- In Python, the required control flow is enabled by the **if statement**.
- Conditions are nothing else but Boolean values, or operations which result in Boolean values.

```
if condition:
    statement_a1
    ...
    statement_an
else:
    statement_b1
    ...
    statement_bm
```

Using if statements

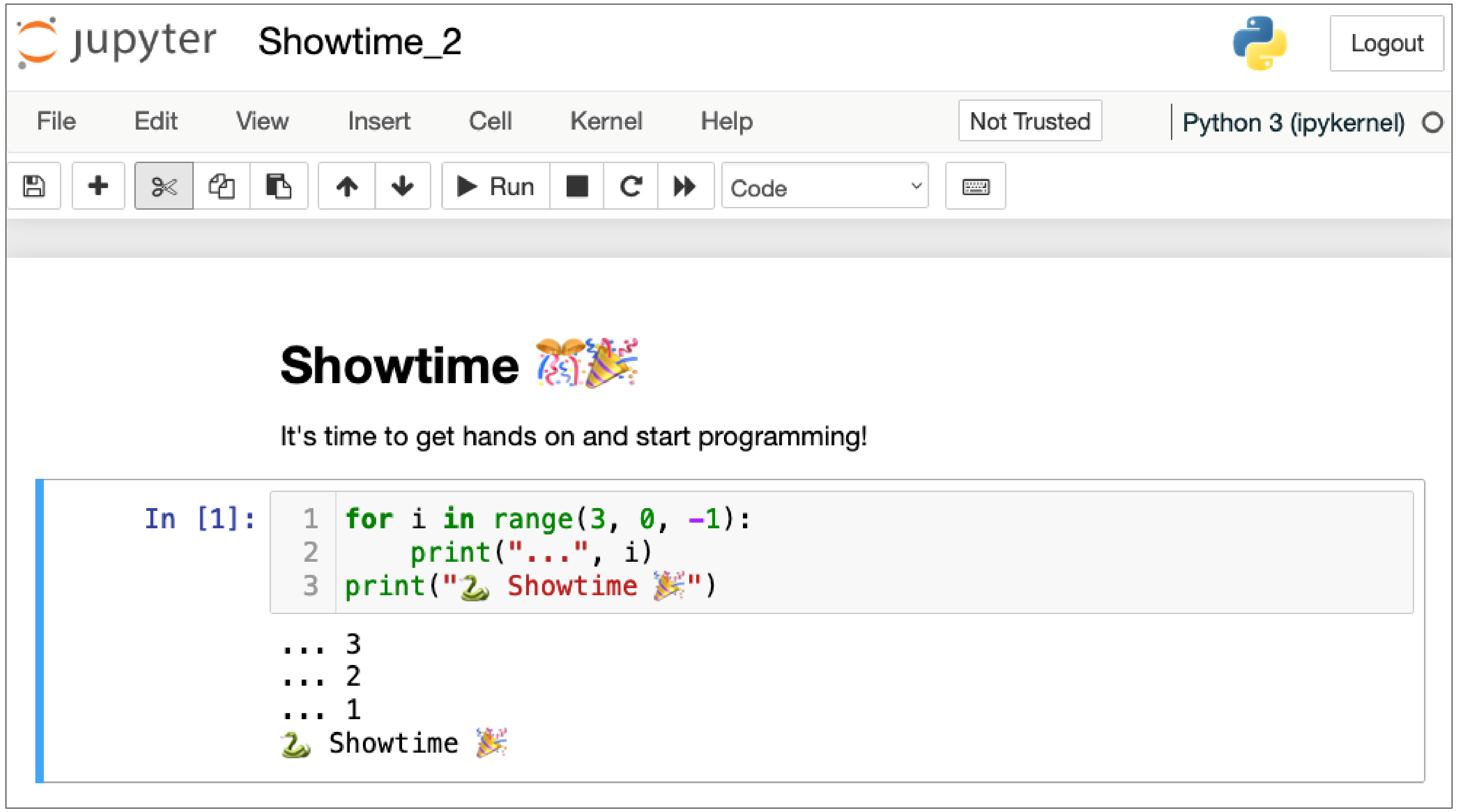
## Showtime

Now it’s time to get hands on and start programming!

If you like, you can open the Jupyter Notebook instructions in parallel to the demo.

If you haven’t done so yet:

- Download the Notebook
- Start the Jupyter Server
- Open the Notebook



Showtime 🎊🎉

It's time to get hands on and start programming!

In [1]:
```
for i in range(3, 0, -1):
    print("...", i)
print("🐍 Showtime 🎉")
```

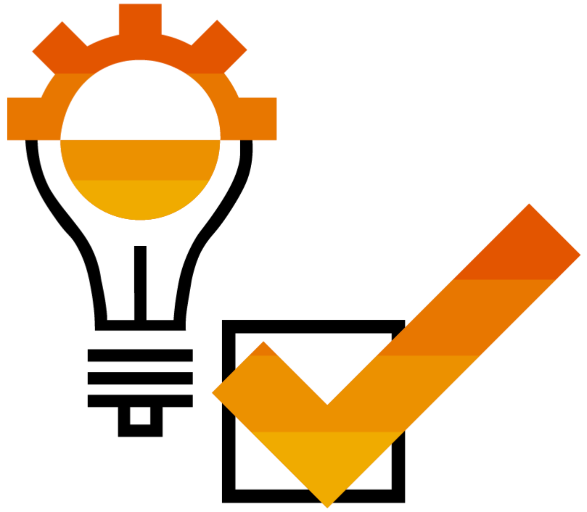
```
... 3
... 2
... 1
🐍 Showtime 🎉
```

Using if statements

## Summary / key takeaways

In this unit you learned ...

- … that decisions are implemented using if statements
- … that there is a special syntax for these if statements
- … that indentation is required (and makes the program more readable)
- … that conditions can be constructed using comparison operators



# Thank You!

**Contact Information:**

open@sap.com

hosted by openSAP open.sap.com

FH AACHEN UNIVERSITY OF APPLIED SCIENCES

Week 1: Python Fundamentals

## Unit 6: Using Multiple If Statements

hosted by openSAP
open.sap.com

FH AACHEN
UNIVERSITY OF APPLIED SCIENCES

Using multiple if statements

## Checking more than one condition

- Sometimes it is necessary to check multiple conditions consecutively
  - "If it is raining tomorrow, I will clean the bathroom" → *if condition_1…*
  - "Otherwise I will go shopping, but only if I get the money back that I lent to my friend" → *elif condition_2…*
  - "Otherwise I will work in my garden" → *else…*

- Python example to classify temperature input:

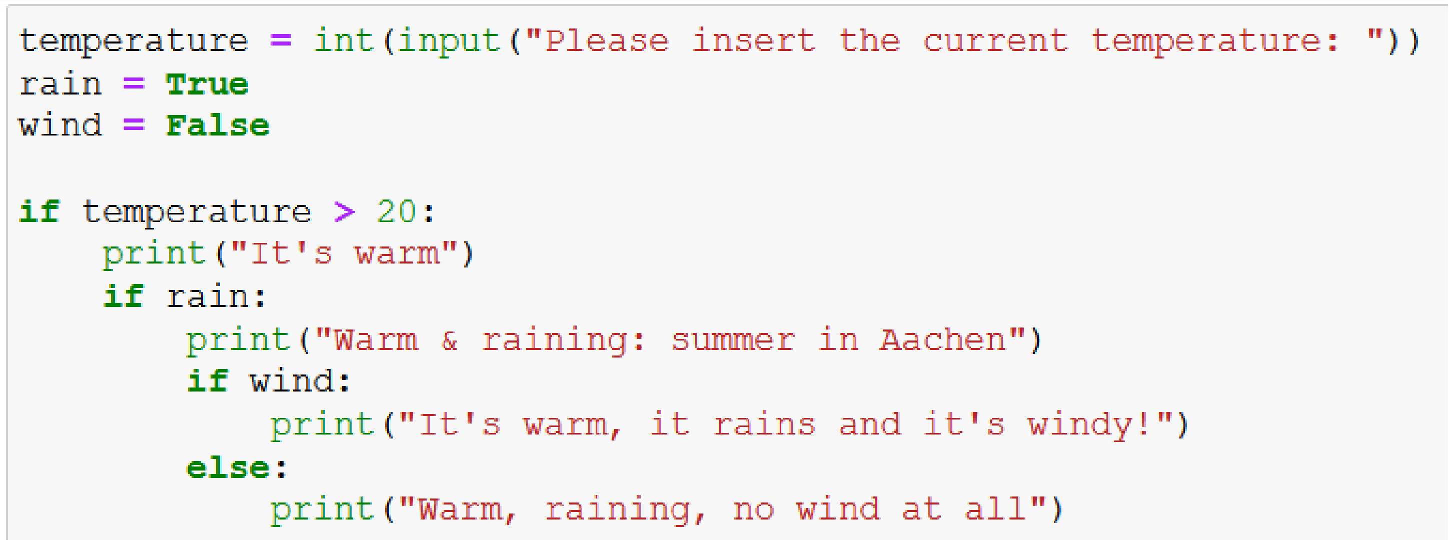
```
temperature = int(input("How many degrees (Celsius) is it? "))

if temperature > 30:
    print("hot")
elif temperature > 20:
    print("warm")
else:
    print("cold")
```

Using multiple if statements

## Nesting conditions

- Sometimes conditions need to be nested
  - “If it is raining tomorrow, I will do housework”
    - “If the kitchen needs cleaning, I will clean it”
      - “If the stove is very dirty, I will start cleaning here, since I need it for cooking”
    - “Else if there is a lot of laundry, I will wash my clothes”
- ➔ Too many nested conditions can be confusing
- Extended classification of temperatures in Python:



```python
temperature = int(input("Please insert the current temperature: "))
rain = True
wind = False

if temperature > 20:
    print("It's warm")
    if rain:
        print("Warm & raining: summer in Aachen")
        if wind:
            print("It's warm, it rains and it's windy!")
        else:
            print("Warm, raining, no wind at all")
```

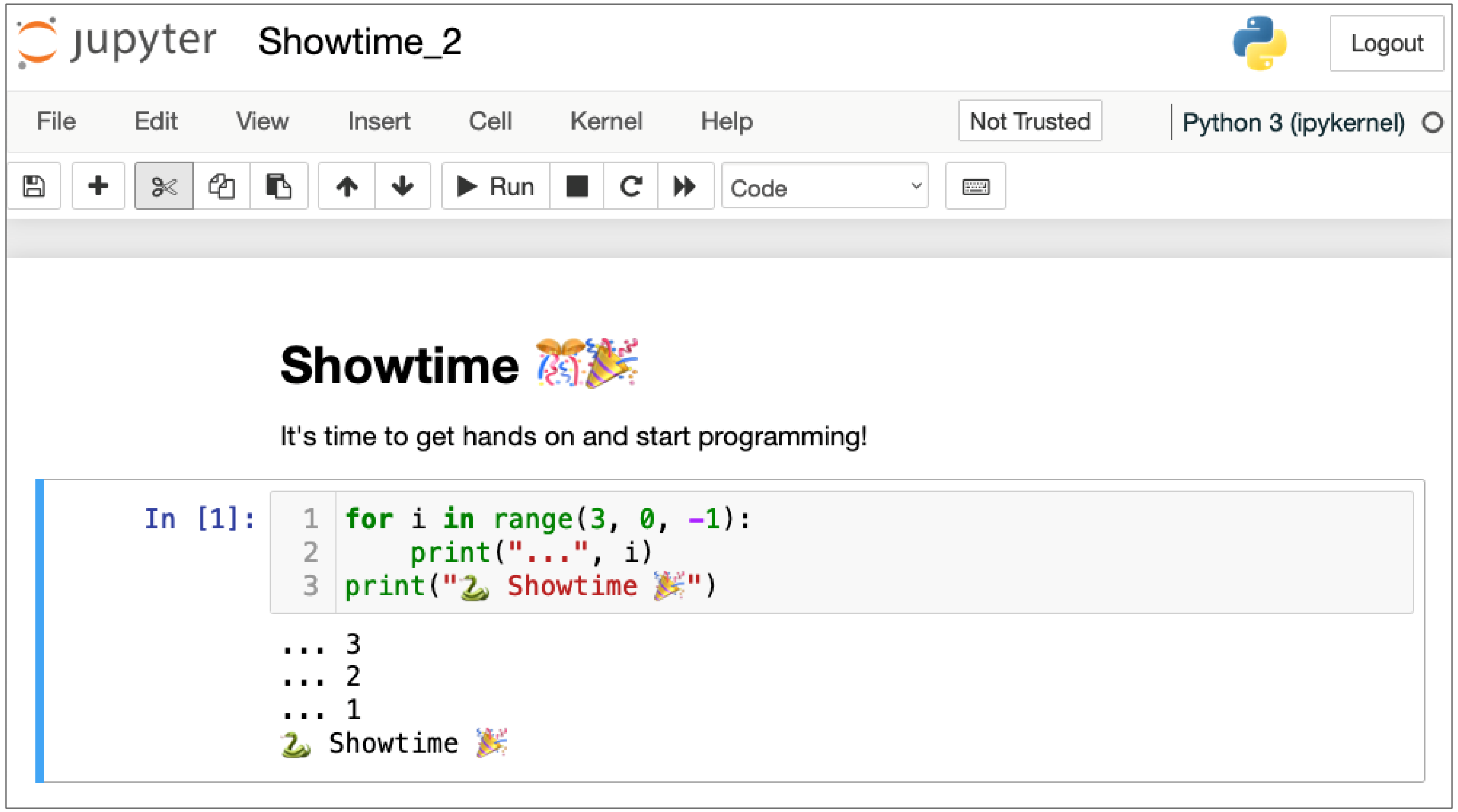
Using multiple if statements
## Showtime

Now it’s time to get hands on and start programming!

If you like, you can open the Jupyter Notebook instructions in parallel to the demo.

If you haven’t done so yet:

- Download the Notebook
- Start the Jupyter Server
- Open the Notebook



Showtime 🎊🎉

It's time to get hands on and start programming!

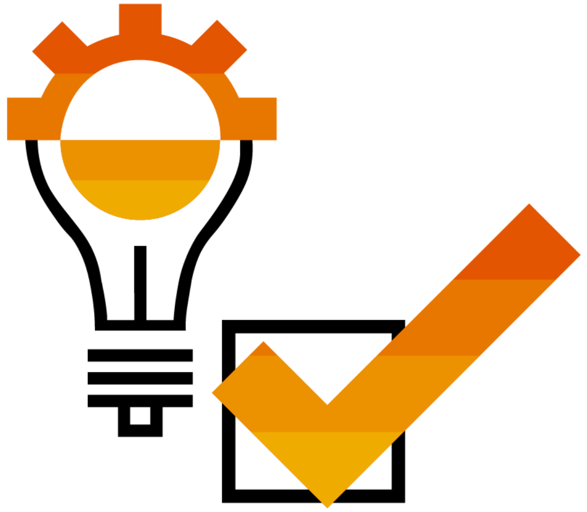
In [1]:
```
for i in range(3, 0, -1):
    print("...", i)
print("🐍 Showtime 🎉")
```

```
... 3
... 2
... 1
🐍 Showtime 🎉
```

Using multiple if statements
## Summary / key takeaways

In this unit you learned ...

- … how to implement several conditions one after the other
- … that conditions may be nested in Python
- … that many nested conditions can clutter your code



# Thank You!

**Contact Information:**
open@sap.com

hosted by openSAP
open.sap.com

FH AACHEN
UNIVERSITY OF APPLIED SCIENCES

Week 1: Python Fundamentals

## Unit 7: Creating Complex Logical Expressions

hosted by openSAP open.sap.com

FH AACHEN UNIVERSITY OF APPLIED SCIENCES

Creating complex logical expressions

## Combining conditions to complex logical expressions

- Testing multiple conditions is possible with logical operators: and, or, not
  – “I will clean the kitchen tomorrow, ***if*** it rains ***and*** I have time ***or*** I don’t know what else to do”

| Logical *and* | | |
|---|---|---|
| **a** | **b** | **a and b** |
| False | False | False |
| False | True | False |
| True | False | False |
| True | True | True |

| Logical *or* | | |
|---|---|---|
| **a** | **b** | **a or b** |
| False | False | False |
| False | True | True |
| True | False | True |
| True | True | True |

| Logical *not* | |
|---|---|
| **a** | **not a** |
| False | True |
| True | False |

**Details:** https://en.wikipedia.org/wiki/Boolean_algebra#Basic_operations

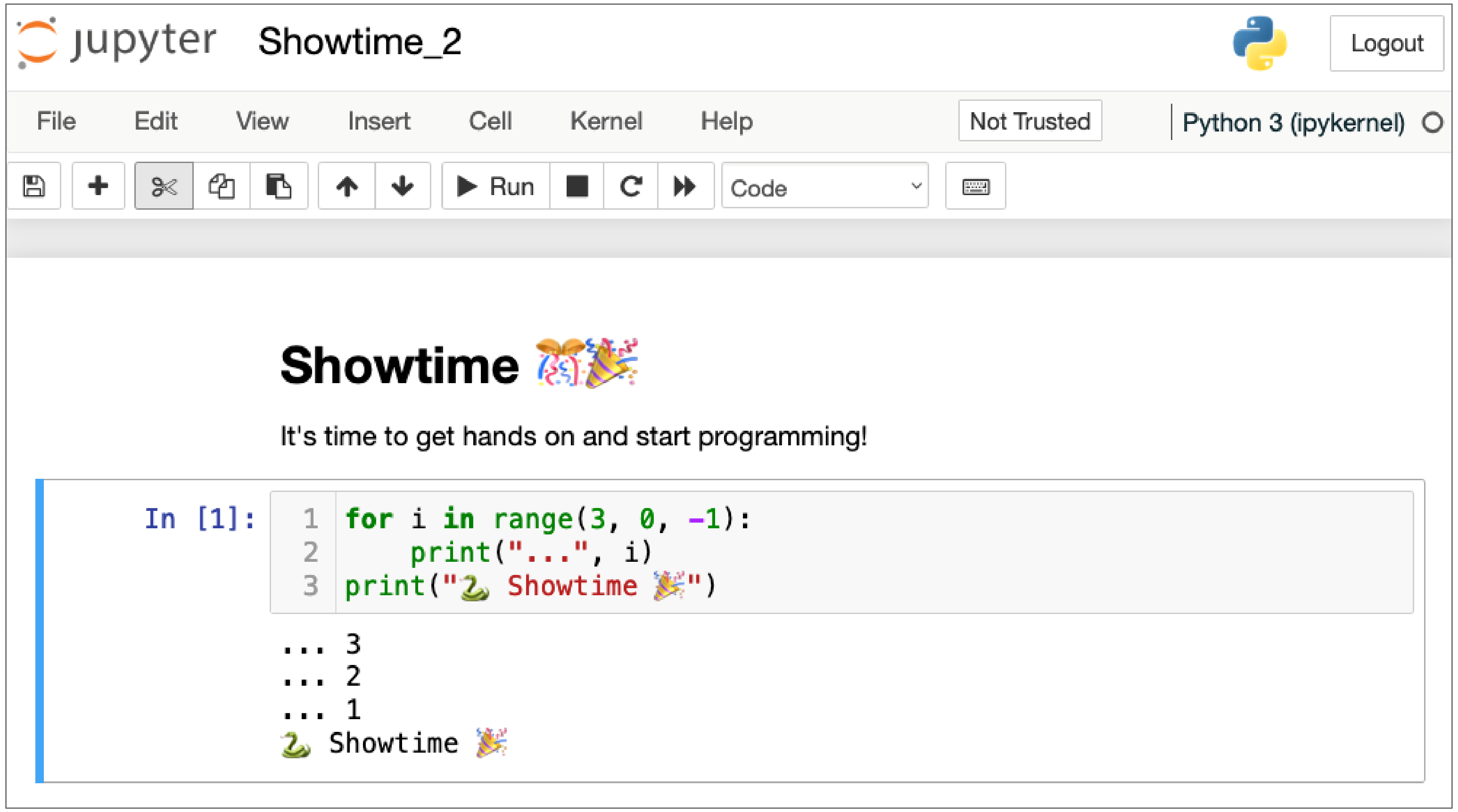
Creating complex logical expressions
## Showtime

Now it’s time to get hands on and start programming!

If you like, you can open the Jupyter Notebook instructions in parallel to the demo.

If you haven’t done so yet:

- Download the Notebook
- Start the Jupyter Server
- Open the Notebook



**Showtime** 🎊🎉

It's time to get hands on and start programming!

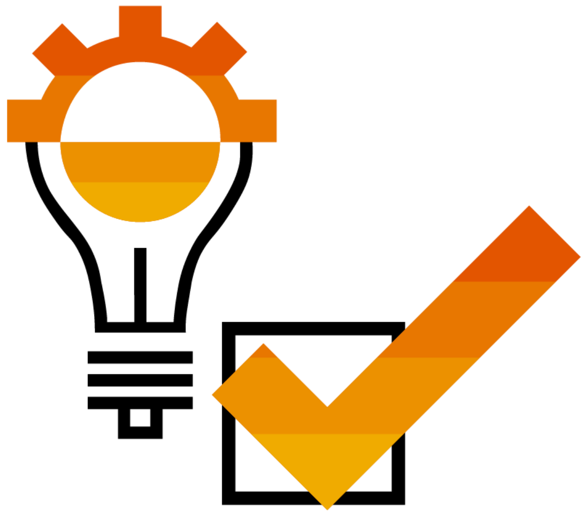
```
In [1]: for i in range(3, 0, -1):
            print("...", i)
        print("🐍 Showtime 🎉")

... 3
... 2
... 1
🐍 Showtime 🎉
```

Creating complex logical expressions
## Summary / key takeaways

In this unit you learned ...

- … how to create complex conditions combining several logical expressions



# Thank You!

**Contact Information:**
open@sap.com

hosted by openSAP
open.sap.com

FH AACHEN
UNIVERSITY OF APPLIED SCIENCES