# PROGRAMMING IN PYTHON I

## Unit 02: Conditions & loops

Michael Widrich & Van Quoc Phuong Huynh
Institute for Machine Learning

JOHANNES KEPLER
UNIVERSITY LINZ

# Outline

1. **Control flow tools**

2. **If, elif, else**

3. **Loops**

# CONTROL FLOW TOOLS

# Motivation

- To solve tasks, we often have to make decisions
  - □ E.g.: *Should I order food, yes or no?*
- These decisions can be conditional
  - □ E.g.:
    *Should I order food, yes or no?*
    *If I'm hungry and don't want to cook: Yes.*
    *Otherwise: No.*
- We can also create loops using such conditions
  - □ E.g.:

(1) *Should I order food now, yes or no?*
(2) *I'm hungry and don't want to cook: Yes.*
(3) *I'm not hungry and don't want to cook: Go to (1).*
(4) *Otherwise: No.*

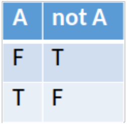
# Evaluating conditions

- We already heard about the boolean datatype
  - □ Can be True or False
- To make decisions, we need an expression that results in a boolean value
  - □ E.g.: *Yes or no?*, *Is 'a' equal to 'b'?*, *Is the value within a range?*, …
- Often we use logical operations or comparisons[1] as such expressions
  - □ E.g.:

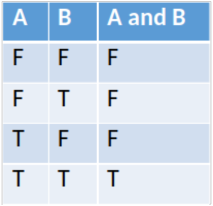
```
4 > 5 # -> False
3 < 4 < 5 # -> True
True or False # -> True
```

[1] https://docs.python.org/3.7/reference/expressions.html?#comparisons

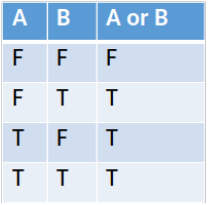
# Logical operations



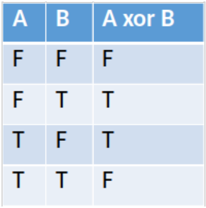
| A | not A |
|---|---|
| F | T |
| T | F |



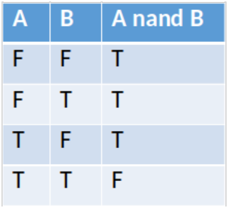
| A | B | A and B |
|---|---|---|
| F | F | F |
| F | T | F |
| T | F | F |
| T | T | T |



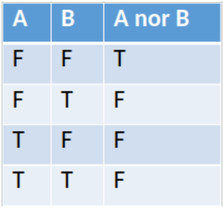
| A | B | A or B |
|---|---|---|
| F | F | F |
| F | T | T |
| T | F | T |
| T | T | T |



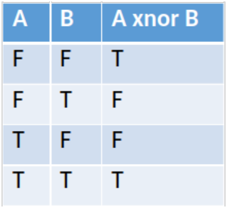
| A | B | A xor B |
|---|---|---|
| F | F | F |
| F | T | T |
| T | F | T |
| T | T | F |



| A | B | A nand B |
|---|---|---|
| F | F | T |
| F | T | T |
| T | F | T |
| T | T | F |



| A | B | A nor B |
|---|---|---|
| F | F | T |
| F | T | F |
| T | F | F |
| T | T | F |



| A | B | A xnor B |
|---|---|---|
| F | F | T |
| F | T | F |
| T | F | F |
| T | T | T |

# IF, ELIF, ELSE

# If, elif, else

- Depending on some condition, we may want to (not) execute different parts of our code
- If, elif, else statements allow to implement such a decision making
  - □ If (=if) `a` is True, then do `s`,
    otherwise if (=elif) `b` is True, then do `t`,
    otherwise if (=elif) `c` is True, then do `u`,
    otherwise (=else) do `v`
- More details in file `02_code.py`

LOOPS

# Background: goto

- Depending on some condition, we may want to (not) execute different parts of our code
- Sometimes, this also includes repeating the execution of code that was already executed
- The goto statement in many languages allows you to jump to a selected line in your code
    - You can jump around freely within your code during the execution time
    - This makes the code hard to read (*spaghetti code*)
- To avoid this bad readability, other tools such as loops are provided
    - The goto statement is not available in Python and other languages

# While loop

- The while loop in Python will repeat a part of code as long as an expression is True

  E.g.:

  - □ Ask user for password until they enter the correct password
  - □ Run some main routine of a micro controller until power is gone (e.g. keep driving around a small robot)
  - □ Keep optimizing network parameters until the output is close enough to the target

- Danger: This can (and often does) lead to endless loops if expression is never False!

# For loop

- The for loop in Python will repeat a part of code for each element in an iterable

  E.g.:

  - □ For each element in list `['a', 'b', 'c']` compute the upper-case letter
  - □ For a given number of updates `range(n_updates)`, update the weights of a neural networks

- More details on loops in file `02_code.py`
- More details on control flow tools in Python: `https://docs.python.org/3.7/tutorial/controlflow.html`