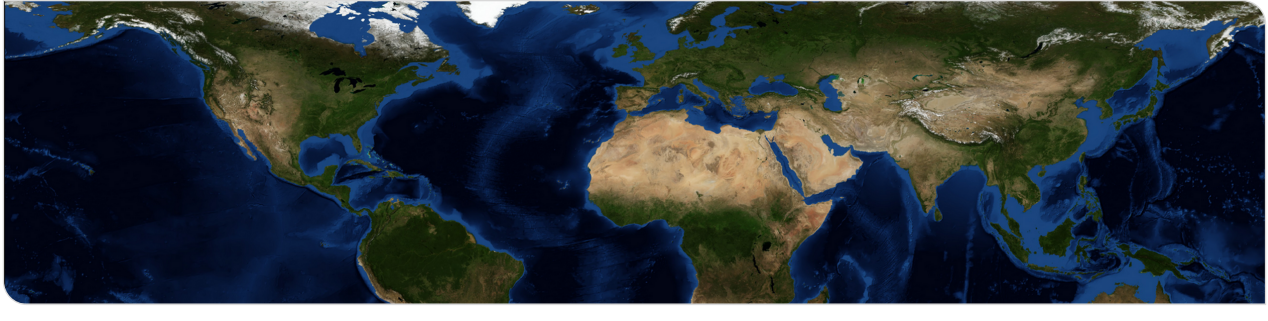
Karlsruher Institut für Technologie

Introduction to Python

3. Lecture: Loops and statements

Dr. Jutta Vüllers, Dr. Julia Fuchs, Dr. Annika Bork-Unkelbach | WS 2023/24



Overview

Recap: Important things from the first lectures

What we learned so far

- setup a working python environment
- open jupyter notebooks
- using Python as a calculator
- writing a programm
  - give some input
  - calculate something
  - give some output
  - save code and run a programm

Recap example

Calculate the mean time you need to get to university in the morning

```
#calculate time to go to university
dist = input("Give distance from home to university in km: ")
mean_velocity = 5.0
time = float(dist) / mean_velocity
time_in_minutes = time * 60
print ("Mean time to university in minutes: ",time_in_minutes)
```

Motivation

Imagine you want to choose whether to walk or go by bicycle / train ?
Or imagine you would like to ask more than one user?
How would the code look like?

Control structures

Today, we will look at:

1. Decision control using the if statement
2. Repeating sequences of code
   - while loop
   - for loop

The if statement

Calculate the mean time you need to get to university in the morning

```
#calculate time to go to university
dist = float(input("Give distance from home to university in km: "))
transport = int(input("Do you walk (press 1) or ride your bycicle (press 2)?"))
mean_vel_walk = 5.0
mean_vel_bike = 20.0
if transport == 1:
    time = dist / mean_vel_walk
if transport == 2:
    time = dist / mean_vel_bike
time_in_minutes = time * 60
print ("Mean time to university in minutes: ",time_in_minutes)
```

Comparison operators

Comparison Operator	Meaning
==	Equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
!=	Not equal to

Logigal operators

Logical operator	Description
and	True, if both statements are true
or	True, ifm one statement is true
not	reverse the result

Identity operators

Identity Operator	Description
is	Returns true, if both opertors are the same object
is not	Returns true, if both operators are not the same object

Membership operators

Membership operator	Description
in	Returns true if a sequence with the specified value is present in the object
not in	Returns true if a sequence with the specified value is not present in the object

The if elif else statement

```
#calculate time to go to university
dist = float(input("Give distance from home to university in km: "))
transport = int(input("Do you walk (press 1) or ride your bycicle (press 2)?"))
mean_vel_walk = 5.0
mean_vel_bike = 20.0

if transport == 1:
    time_in_minutes = (dist / mean_vel_walk) * 60
    print ("Mean time to university in minutes: ",time_in_minutes)
elif transport == 2:
    time_in_minutes = (dist / mean_vel_bike) * 60
    print ("Mean time to university in minutes: ",time_in_minutes)
else:
    print("Please enter a valid number of transport. ")
```

Comments on the if elif else statement

When to use which statement:

- else
  - use in cases with only 2 possible situations
  - can be omitted
  - can be helpful to catch exceptions
- elif (else if)
  - use in cases with more than 2 possible situations
  - faster than multiple if statements

Repeating taks: The while loop

```
#Print all numbers up to 10
number = 0
  while number != 10:
      number= number + 1
      print(number)
```

Exit the while loop immediately with break

```
#Print messages until the user enters 'quit'
  while True:
      message = input("Please type a message, type quit for ending: ")
      if message == 'quit':
          break
      else
          print(message)
```

With break the while loops is exited immediately without running any of the remaining code in the loop.

Continue

```
#Print even numbers
current_number = 0
while current_number < 20:
  current_number += 1
  if current_number % 2 != 0+
   continue
  print(current_number)
```

Continue is used to return to the beginning of the loop.

Looping through lists: The for loop

The for loop is used to repeat tasks on lists:

```
ilist = ["Julia", "Jutta", "Annika", "Alexandre", "Yichen"]
print("The names of the python course team are: ")
for name in list:
    print(name)
```

How to work with this exercise

The examples and exercises in todays Jupyter Notebook are intended to **WORK** with. Please always...

- ... read the instructions carefully
- ... copy an example cell and paste it below
- ... run the code and see what happens
- ... modify the code and run it again
- ... document your findings in an additional Markdown cell below