

Mathematics and Computer Science for Modeling Programming Session

Stephan Sehring
using materials by Jan Tekülve and Daniel Sabinasz

Institut für Neuroinformatik, Ruhr-Universität Bochum

September 18, 2024

Setting Up

- ▶ Open the *Anaconda Navigator*
- ▶ Launch the *Spyder* IDE (Integrated Development Environment)
- ▶ Create your first python script file
 - ▶ Close the default temporary file
 - ▶ Go to *File* → *Save as ...*
 - ▶ (*Recommended*) Create a new folder for your python projects
 - ▶ Choose the name *helloworld.py*
- ▶ You are set up to write a python script



Print

- ▶ Write the following line into the file:

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

- ▶ Press the green *Play* button in the toolbar to execute the script
- ▶ Observe the output in the console on the right

User Input

- ▶ Use input to prompt the user

```
person = input('Enter your name: ')\n#whatever the user types is stored in person\nprint('Hello ' + person)
```

User Input

- ▶ Use input to prompt the user

```
person = input('Enter your name: ')\n#whatever the user types is stored in person\nprint('Hello ' + person)
```

- ▶ Invalid Data Types

```
inputValue = input('Please enter a number: ')\nresult = 5 + inputValue # This results in an error!
```

User Input

- ▶ Use input to prompt the user

```
person = input('Enter your name: ')\n#whatever the user types is stored in person\nprint('Hello ' + person)
```

- ▶ Invalid Data Types

```
inputValue = input('Please enter a number: ')\nresult = 5 + inputValue # This results in an error!
```

- ▶ Variables might need to be *type casted*

```
result = 5 + float(inputValue)\n#This works if an actual number was typed
```

Type Casting

► Implicit Typecast

```
a = 1.0 #float
```

```
b = 2 #int
```

```
c = a + b #3.0 float
```

Type Casting

► Implicit Typecast

```
a = 1.0 #float
b = 2 #int
c = a + b #3.0 float
```

► Explicit Typecasts

```
d = float(b) #2.0
e = 3.7
f = int(3.7) #3 Any floating point is cut off
g = str(e) #String '3.7'
h = int(g) # This results in an error!
i = float(g) # 3.7
print('Variable i is: ' +str(i)) #Print expects strings
```

Useful built-in Functions

► Rounding and Absolute Value

```
a = 3.898987897897
b = round(a,3) #3.899
c = abs(-3.2) #|-3.2| = 3.2
t = type(c) #t is <class 'float'>
test = t is float # True
```

► The math module

```
import math #Import makes a module available
squareTwo = math.sqrt(2) # $\sqrt{2}$ 
power = math.pow(3,4) #  $3^4$ 
exponential = math.exp(4) # $e^4$ 
piNumber = math.pi #3.14159265359
```

Tasks: Control Statements

1. Write a Guessing Game, where the script chooses a random integer between 0 and 20 and the user has to guess it. With each guess the user gets told if his guess was higher or lower than the desired number.
 - ▶ Import the python module “random” using the command “import random”
 - ▶ Assign a random integer to a variable using `random.randint(0,20)`
 - ▶ Create a while-loop in which the user is asked for a number using the ‘input()’ function
 - ▶ Depending on the number input tell the user whether his guess was smaller than, larger than or equal to the desired value
 - ▶ Think about how to end the while-loop

Tasks: Lists

2. Write a script that returns the biggest element in a list
 - ▶ Create a list with arbitrary numbers of your choice
 - ▶ Loop through the list with a for loop
 - ▶ In each loop compare the current list element with your current estimate of the highest number
3. Write a script that looks for a specific element in the list and deletes it
 - ▶ Loop through the list with a for-loop and store the elements position in a variable
 - ▶ After the for loop remove the element at that position with the *del* command
- 4*. Write a script that takes a list and transfers its elements to a second list in sorted order.
 - ▶ Look for the smallest element in the first list. Write it to the second list. Delete it in the first list. Repeat.