

Manager Hackathon

How to code in Python



**Basics zu Arduino
und Programmierung**

Python

Weit verbreitete Programmiersprache. Verwendet für:

- Webentwicklung
- Wissenschaft (u.a. Mathematik, Ingenieurwissenschaften)
- Data Science und Analytics
- uvm

How to code

Arduino (C) vs. Python

C:

```
int LED_PIN = 12;

void setup() {
  pinMode(LED_PIN, OUTPUT);
}

void loop() {
  digitalWrite(LED_PIN, HIGH);
  delay(1000);
  digitalWrite(LED_PIN, LOW);
  delay(1000);
}
```

Python:

Definiert eine Funktion main – main wird zum Start Des Programms aufgerufen

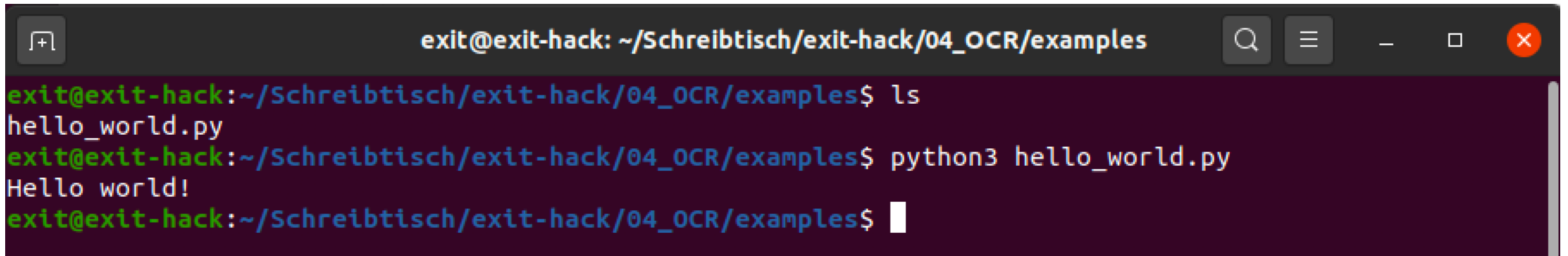
```
def main():
    print("Hello world!")

if __name__ == "__main__":
    main()
```

Alle eingerückten Befehle nach der Definition gehören zur Funktion, es gibt keine geschweiften Klammern!

How to code

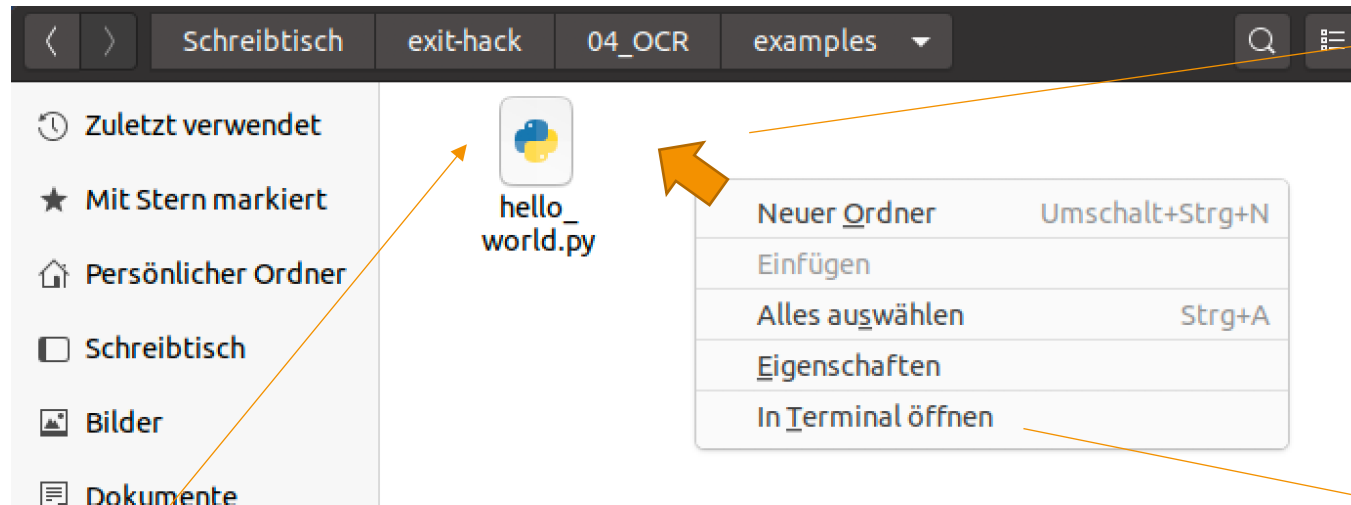
Pythonprogramme werden von der Kommandozeile aus gestartet

A terminal window with a dark background and light-colored text. The window title bar shows the path 'exit@exit-hack: ~/Schreibtisch/exit-hack/04_OCR/examples' and standard window controls. The terminal content shows a user running 'ls' to list files, then 'python3 hello_world.py' to execute a script, which outputs 'Hello world!'.

```
exit@exit-hack: ~/Schreibtisch/exit-hack/04_OCR/examples
exit@exit-hack:~/Schreibtisch/exit-hack/04_OCR/examples$ ls
hello_world.py
exit@exit-hack:~/Schreibtisch/exit-hack/04_OCR/examples$ python3 hello_world.py
Hello world!
exit@exit-hack:~/Schreibtisch/exit-hack/04_OCR/examples$
```

How to code

Python – Kommandozeile starten



1. Rechtsklick mit der Maus

2. Öffnet „Kommandozeile“
bzw. Terminal

Datei, in der das Python-Programm
gespeichert ist.

How to code

Python – Programm von der Kommandozeile ausführen

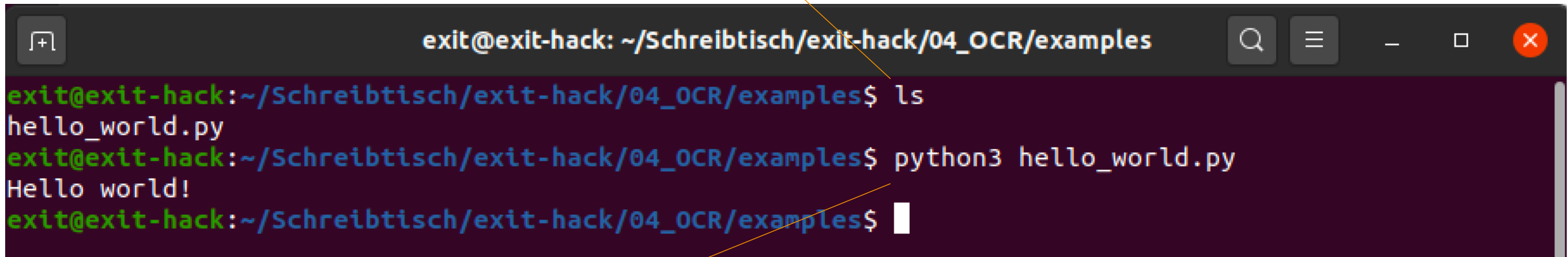
Tippen:

ls

Enter-Taste

Zeigt die Dateien im aktuellen Ordner –

Hier: hello_world_python.py



A terminal window with a dark background. The title bar shows the user 'exit' on a machine named 'exit-hack', with the current directory being '~/.Schreibtisch/exit-hack/04_OCR/examples'. The terminal content shows the following sequence of commands and output: a prompt followed by 'ls' which outputs 'hello_world.py'; a prompt followed by 'python3 hello_world.py' which outputs 'Hello world!'; and a final prompt with a cursor. An orange line points from the text 'Hier: hello_world_python.py' to the file name in the terminal output.

```
exit@exit-hack: ~/.Schreibtisch/exit-hack/04_OCR/examples
exit@exit-hack:~/.Schreibtisch/exit-hack/04_OCR/examples$ ls
hello_world.py
exit@exit-hack:~/.Schreibtisch/exit-hack/04_OCR/examples$ python3 hello_world.py
Hello world!
exit@exit-hack:~/.Schreibtisch/exit-hack/04_OCR/examples$
```

Tippen:

Python3 hello_world.py

Enter-Taste

Startet das in hello_world.py geschriebene Programm

Hier: Nur ausführen von main() und dort Ausgabe von "Hello

World!"

Daten an Funktionen übergeben

```
def main():  
    message = "Hello World!"  
    printMessage(message)  
  
def printMessage(msg):  
    print(msg)  
  
if __name__ == "__main__":  
    main()
```

Definiert die Variable message
und speichert dort den Text "Hello World!"

Variable message
(und Inhalt "Hello World!") werden an
printMessage übergeben

printMessage erhält den Wert von message
in der Variable msg.

Print (in Standard-Python enthalten)
gibt den Wert von message
Aus.

Funktionen können Werte zurückgeben

```
def main():  
    message = getMessage()  
    printMessage(message)  
  
def getMessage():  
    msg = "Hello World!"  
    return msg  
  
def printMessage(msg):  
    print(msg)  
  
if __name__ == "__main__":  
    main()
```

In message steht jetzt "Hello World!"

Definiert die Variable msg und speichert dort den Text "Hello World!"

Gibt den Wert, der in msg gespeichert ist zurück.

Bibliotheken enthalten Funktionen, die man als Programmierer verwenden kann

```
import time
```

Importieren der Bibliothek time.
Jetzt lassen sich Funktionen aus dieser Bibliothek verwenden.
D.h. man kann als Entwickler auf der Arbeit anderer Entwickler
Aufbauen.

```
def main():  
    now = time.ctime()  
    print("Current Time: ")  
    print(now)
```

Aufrufen der Funktion ctime()
Aus der Bibliothek time

```
if __name__ == "__main__":  
    main()
```

How to code

Aufgabe 3 - Texterkennung

Auf Ihrem Rechner finden Sie im Ordner

03_SOS/instructions/

die Datei `encrypted_message.txt`

Öffnen Sie die Datei und sehen sich diese an. Offenbar enthält sie eine verschlüsselte Botschaft.

In dem selben Ordner gibt es ein Programm `decrypt.py`

Entschlüsseln Sie die Nachricht durch Ausführen

von `decrypt.py`

Offensichtlich enthält dieses Programm einen Fehler (= Bug).

Finden Sie diesen Bug und entschlüsseln Sie die Nachricht mit dem korrigierten Programm.