

Exit Coding Game

How to code in Python



**Programmierung mit
Python**

How to code Python

Weit verbreitete und einfach zu erlernende Programmiersprache.

Verwendet u.a. für:

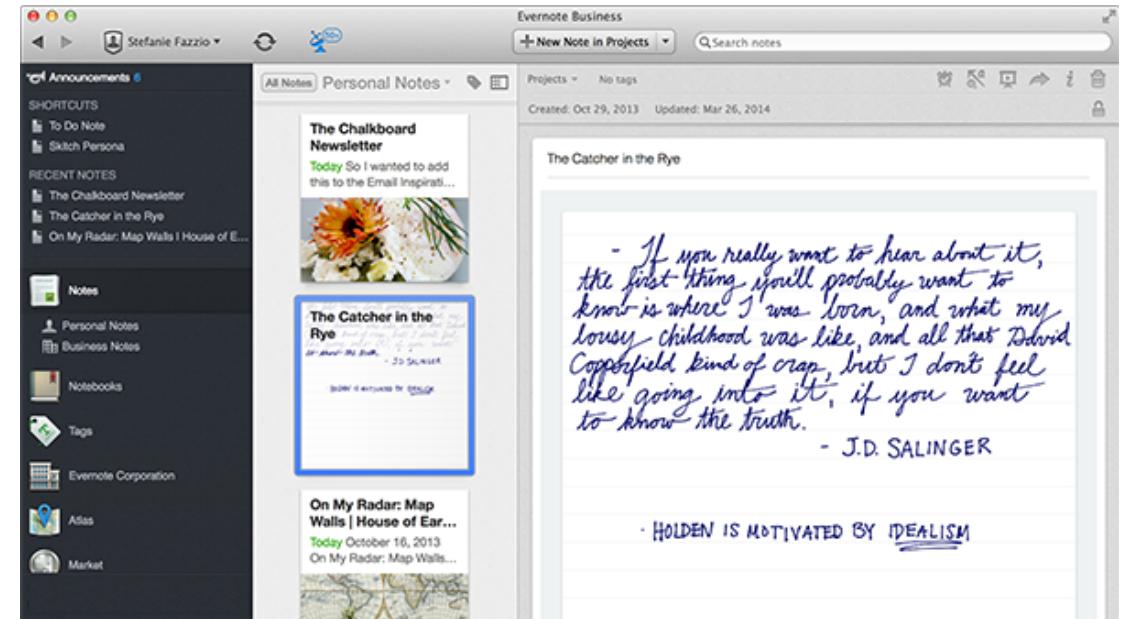
- Webentwicklung
- Wissenschaft (u.a. Mathematik, Ingenieurwissenschaften)
- Data Science und Analytics
- uvm.



Bildquelle: <https://www.informatik-aktuell.de/betrieb/kuenstliche-intelligenz/data-science-buzzword-mit-gehaltvollem-kern.html>

Anwendungen für Texterkennung

- Digitalisierung handschriftlicher Notizen, um diese durchsuchbar und weiterverarbeitbar zu machen
- Digitalisierung von Formularen und Anträgen
- Digitalisierung und automatisierte Auswertung von Kassenzetteln als Basis für "Machine Learning"
- uvm.



How to code

Ausblick automatisierte Analyse von Bilddaten



Quelle: <https://netzpolitik.org/2019/deutsche-grossflughafen-gesichtserkennung-jetzt-auch-fuer-kinder/>

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 – diese wird zum Start des Programms aufgerufen

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

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

Alle eingerückten (Tab-Taste) Anweisungen nach der Definition gehören zur Funktion, es gibt keine geschweiften Klammern!

How To Code

Back in time – Computer kann man auch ohne Fenster bedienen



- In den frühen 80er Jahren haben Computer nur getippte Befehle verstanden
 - Die Bedienung erfolgte über eine “Kommandozeile”
- Windows kam erst 1985 auf den Markt
- In der Softwareentwicklung und Systemadministration haben Kommandozeilen teilweise überlebt...

Bildquelle: <https://www.worthpoint.com/worthopedia/sanyo-crt-30-monochrome-monitor-xt-1811071050>

How to code

Python – Programm von der Kommandozeile ausführen

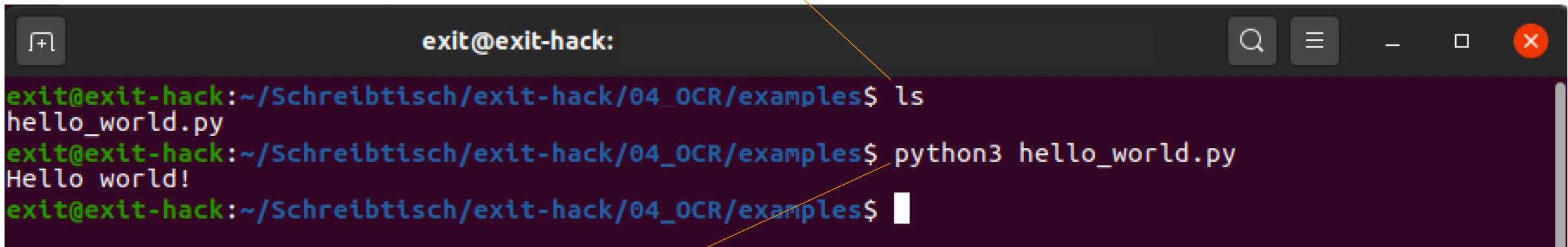
Tippen:

ls

Enter-Taste

Zeigt die Dateien im aktuellen Ordner –

Hier: hello_world_python.py



The screenshot shows a terminal window with a dark background and light-colored text. The title bar says "exit@exit-hack:". The terminal prompt is "exit@exit-hack:~/Schreibtisch/exit-hack/04_OCR/examples\$". The user has run two commands: "ls" which lists the files in the directory, showing "hello_world.py", and "python3 hello_world.py" which executes the script and prints "Hello world!" to the console. The terminal ends with the prompt "exit@exit-hack:~/Schreibtisch/exit-hack/04_OCR/examples\$". Two orange lines from the text above point to the "ls" command and the "python3" command respectively.

```
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

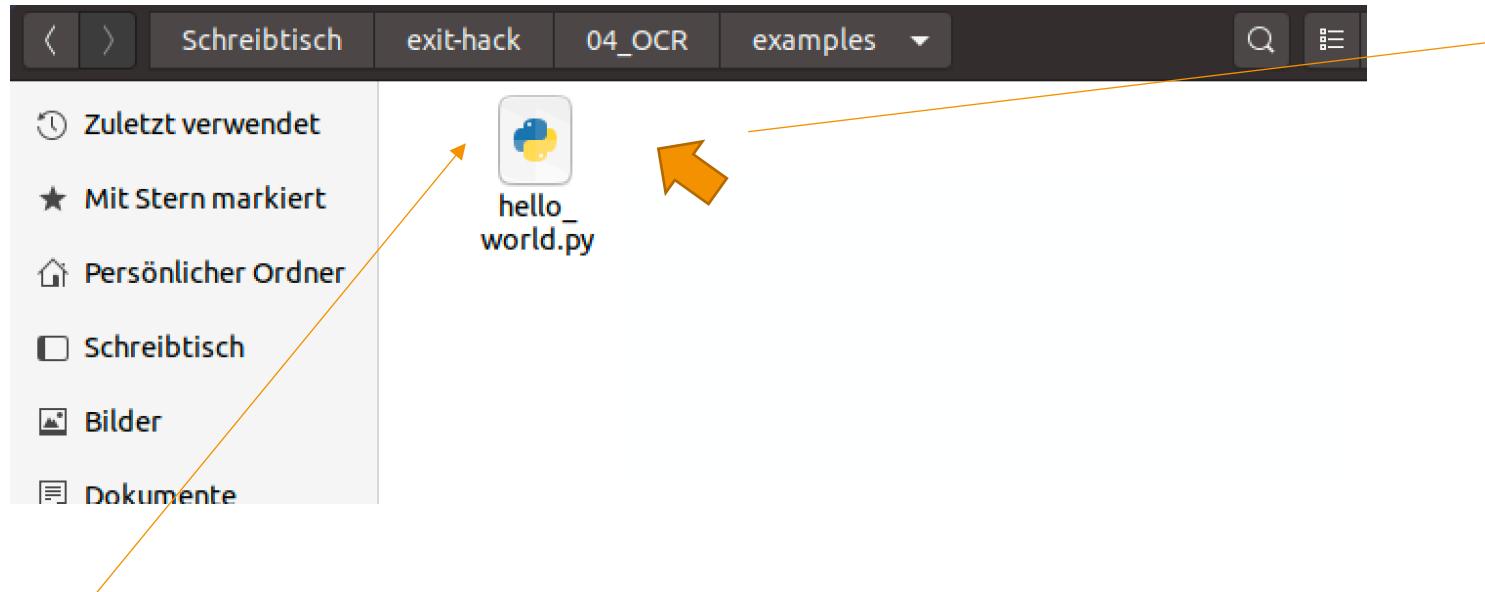
Enter-Taste

Startet das in hello_world.py geschriebene Programm

Hier: Nur Ausführen von main() und dort Ausgabe von *Hello world!*

How to code

Python – Kommandozeile starten

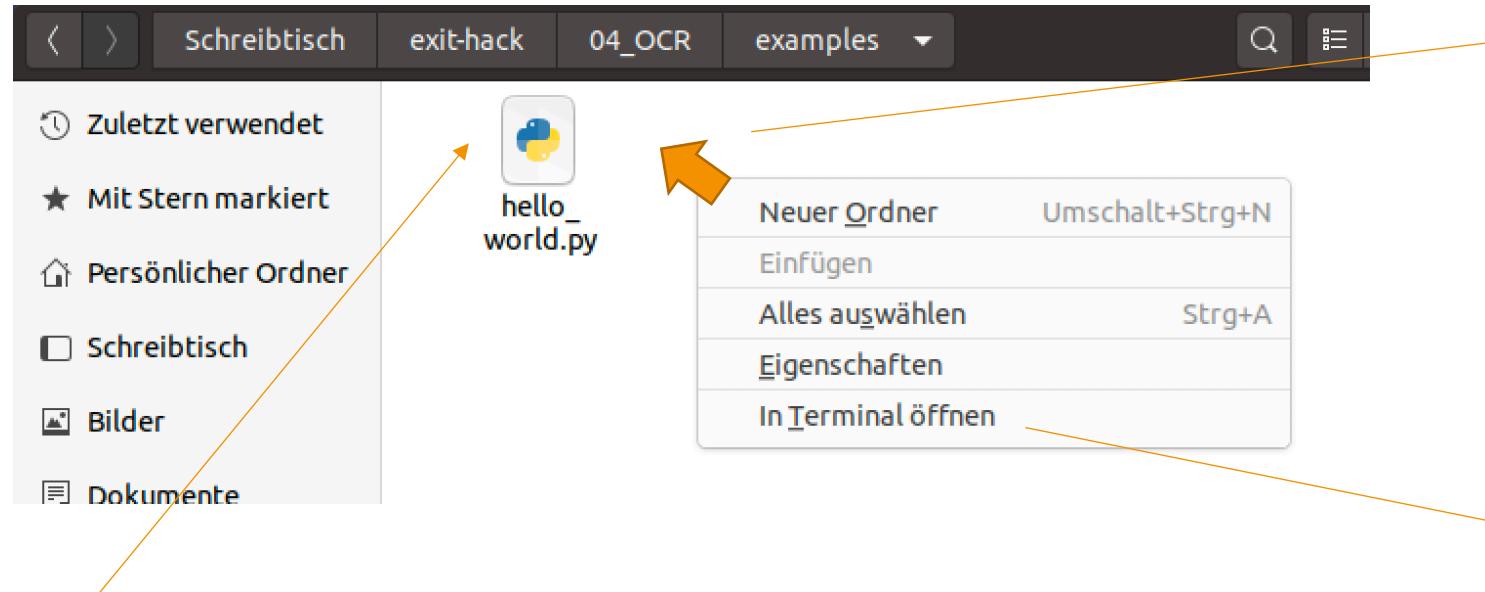


Datei, in der das Python-Programm gespeichert ist.

1. Rechtsklick mit der Maus auf die freie Fläche **neben** der Datei öffnet Kontextmenü

How to code

Python – Kommandozeile starten



Datei, in der das Python-Programm gespeichert ist.

1. Rechtsklick mit der Maus auf die freie Fläche **neben** der Datei öffnet Kontextmenü

2. Klick auf „In Terminal öffnen“ öffnet „Kommandozeile“ bzw. Terminal im Ordner der Datei

Demo: Hello World ausführen

How to code

Daten an Funktionen übergeben

```
def main():
    message = "Hello World!"
    printMessage(message)

def printMessage(message):
    print(message)

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

Definiert die Variable `message` und speichert dort den Text "Hello World!"
In Python gibt es keine Datentypen wie `int`

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

`printMessage` erhält den Wert von `message` (aus der
Funktion `main`) in der neuen Variable `message`.

`print` (in Standard-Python enthalten) gibt
den Wert von `message` aus.

How to code

Funktionen können Werte zurückgeben

```
def main():
    geld = 2
    brotzeit = getLeberkaeseSemmel(geld)
    printBrotzeit(brotzeit)

def getLeberkaeseSemmel(geld):
    print("Kaufe ein für €")
    print(geld)
    return "Leberkäsesemmel"

def printBrotzeit(brotzeit):
    print("Deine Brotzeit: ")
    print(brotzeit)

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

In brotzeit steht jetzt der zurückgegebene Wert "Leberkäsesemmel"

Variable geld (und deren Inhalt 2) werden an getLeberkaeseSemmel übergeben (kennen wir schon)

Variable brotzeit
Wird an Funktion übergeben (kennen wir auch schon)

Funktion gibt einen Wert zurück (z.B. Ergebnis einer Berechnung, oder Daten, die geholt werden, ...)
Hier: Ergebnis ist die "Leberkäsesemmel"

Funktion gibt den Inhalt von brotzeit aus, hier:
"Leberkäsesemmel"

How to code

Bibliotheken enthalten Funktionen, die man als Programmierer verwenden kann

```
import time

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

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

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

Aufrufen der Funktion ctime()
aus der Bibliothek time

Legt die Variable now an und speichert das Ergebnis der
Funktion ctime aus der Bibliothek time in der Variable
now. D.h. die Funktion ctime "gibt Informationen bzw.
Daten zurück"