

Exit Coding Game

How to code in 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.

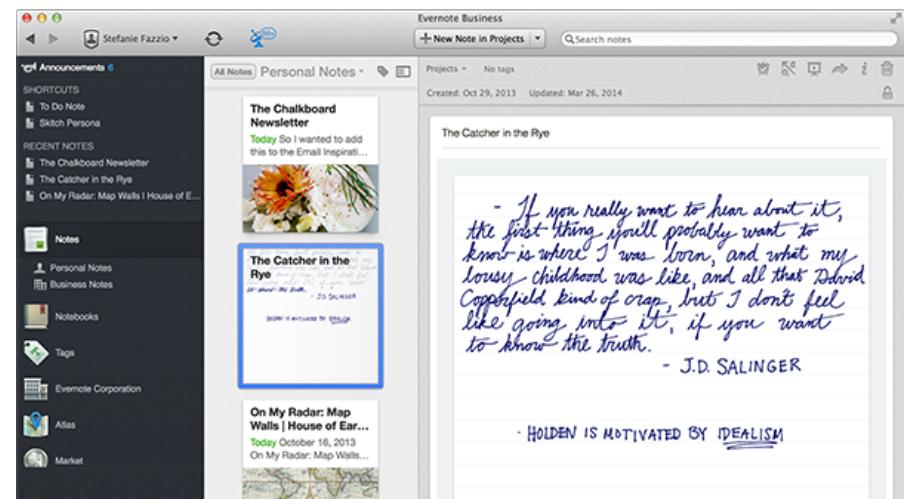


Bildquelle: <https://www.informatik-aktuell.de/betrieb/kuenstliche-intelligenz/data-science-buzzword-mit-gehaltvollem-kern.html>

How to code

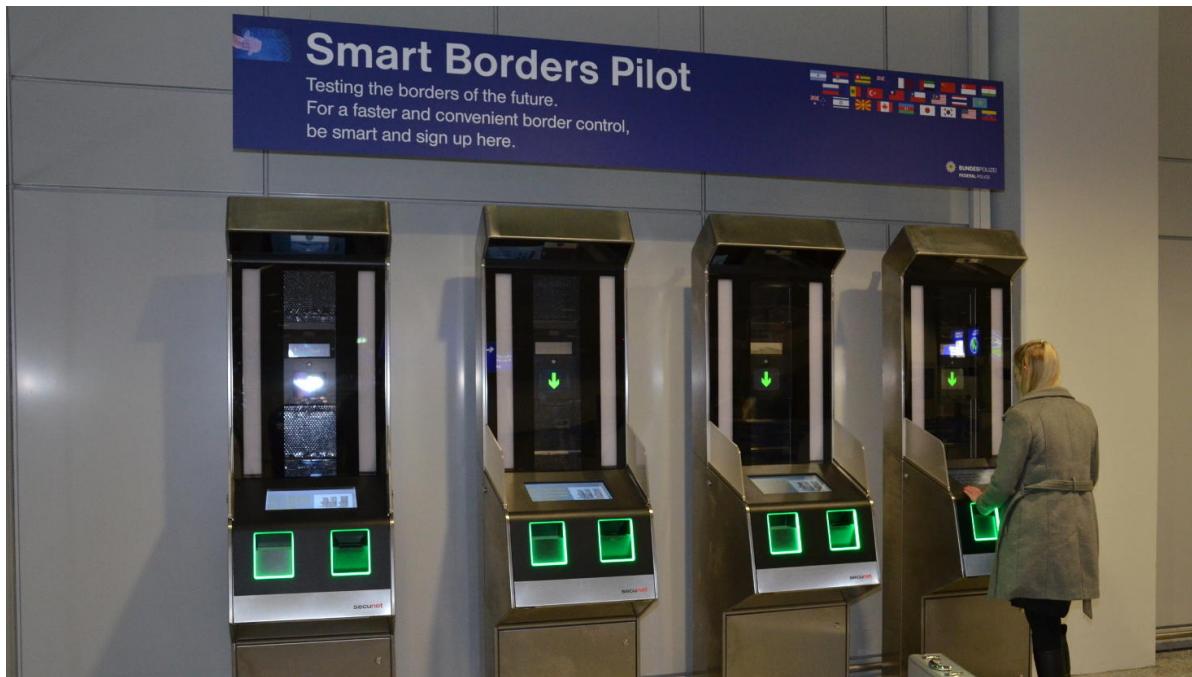
Anwendungen für Texterkennung

- 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/>

17.03.21

8

 infineon

 DGO
Digitale Gründerinitiative OBERPFALZ

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

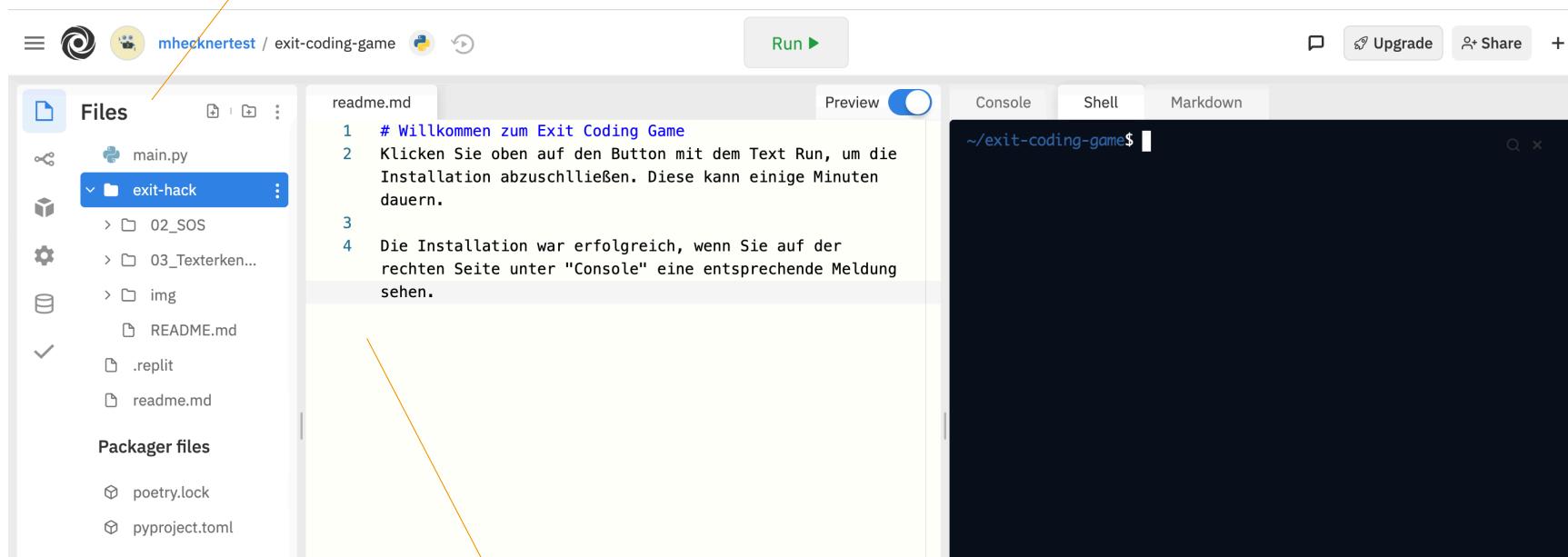


- In den frühen 80er Jahren haben Computer nur getippte Befehle verstanden
 - Die Bedienung erfolgte über eine "Kommandozeile"
- Windows kam erst 1985 auf den Markt
- In der Softwareentwicklung und Systemadministration haben Kommandozeilen teilweise überlebt...

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

How to code Repl – Code Editor und Kommandozeile über den Browser in der Cloud

Datei-Explorer (wie in Windows)



Text-Editor (hier schreibt man den Code)

Demo: Datei hello_world.py im Ordner 03_Texterkennung/examples/ im Editor öffnen



How to code Python – Programme von der Kommandozeile ausführen (1/2)

Datei hello_world.py angeklickt

The screenshot shows a Python code editor interface. The top bar displays the path 'mhecknertest / exit-coding-game' and includes a 'Run ▶' button, an 'Upgrade' button, a 'Share' button, and a '+' button. The left sidebar, titled 'Files', shows a project structure with files like 'main.py', 'exit-hack', '02_SOS', '03_Texterkenn...', 'examples', and several 'hello_...' files. The 'hello_world.py' file is highlighted with a blue selection bar at the bottom. The main editor area shows the following code:

```
1 def main():
2     print("Hello world!")
3
4 if __name__ == "__main__":
5     main()
```

The right side of the interface features a 'Console' tab, which is currently active, showing the command prompt ' ~/exit-coding-game\$'. An orange line points from the text 'Datei hello_world.py angeklickt' to the 'hello_world.py' file in the file tree. Another orange line points from the text 'Programmcode in hello_world.py' to the code in the editor.

Programmcode in hello_world.py

Demo: Datei hello_world.py im Ordner 03_Texterkennung/examples/ im Editor öffnen



How to code Python – Programme von der Kommandozeile ausführen (2/2)

Datei hello_world.py angeklickt

```
exit-hack/03_Texterkennung/examples/hello_world.py
1 def main():
2     print("Hello world!")
3
4 if __name__ == "__main__":
5     main()
```

Programmcode in hello_world.py

Demo: Datei hello_world.py im Ordner 03_Texterkennung/examples/ im Editor öffnen



**Auch auf der Kommandozeile muss man sich im
richtigen Ordner befinden...**

How to code

Ordner wechseln auf der Kommandozeile

The screenshot shows a terminal window with the following session:

```
~/exit-coding-game$ python3 hello_world.py
python3: can't open file 'hello_world.py': [Errno 2]
No such file or directory
~/exit-coding-game$ cd exit-hack
~/exit-coding-game$ cd 03_Texterkennung
~/exit-hack/03_Texterkennung$ cd examples
~/exit-hack/03_Texterkennung/examples$ python3 hello_world.py
Hello world!
~/exit-hack/03_Texterkennung/examples$
```

The terminal shows the user navigating through directory levels to find the file. The path taken is: `~/exit-coding-game` → `exit-hack` → `03_Texterkennung` → `examples`. The final command executed is `python3 hello_world.py`, which outputs "Hello world!".

Demo: Datei `hello_world.py` im Ordner `03_Texterkennung/examples/` im Editor öffnen

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
```

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()
```

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"

