Debugging

Wessen Programme laufen fehlerfrei?

Software-Fehler: ist das schlimm?

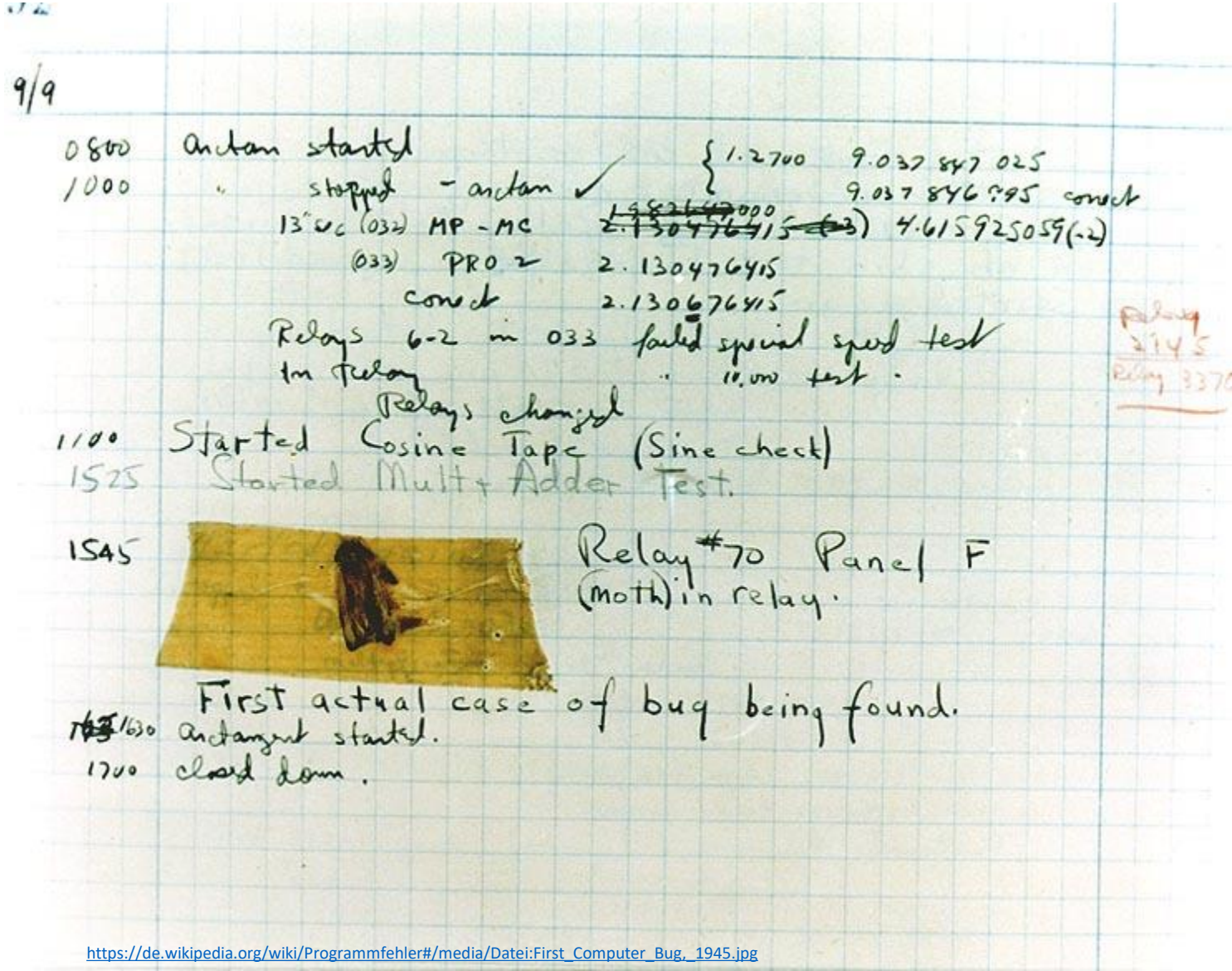
- Wenn wir einen Fehler finden, dann fixen wir den halt einfach

Was „Debugging“ bedeutet

- Allgemein: Code von Fehlern befreien
- Ein „Bug“ ist ein Softwarefehler
- Bug ist ein Insekt, eine Motte, eine Wanze, ein Käfer, etc.
- → Entwanzen

Der wohl erste „wirkliche“ Bug

- 1945 oder 1947
- Mark II (Relais)
- Grace Hopper



https://de.wikipedia.org/wiki/Programmfehler#/media/Datei:First_Computer_Bug,_1945.jpg

Debugging im engeren Sinn

- Debugging im weiteren Sinn: Code von Fehlern befreien

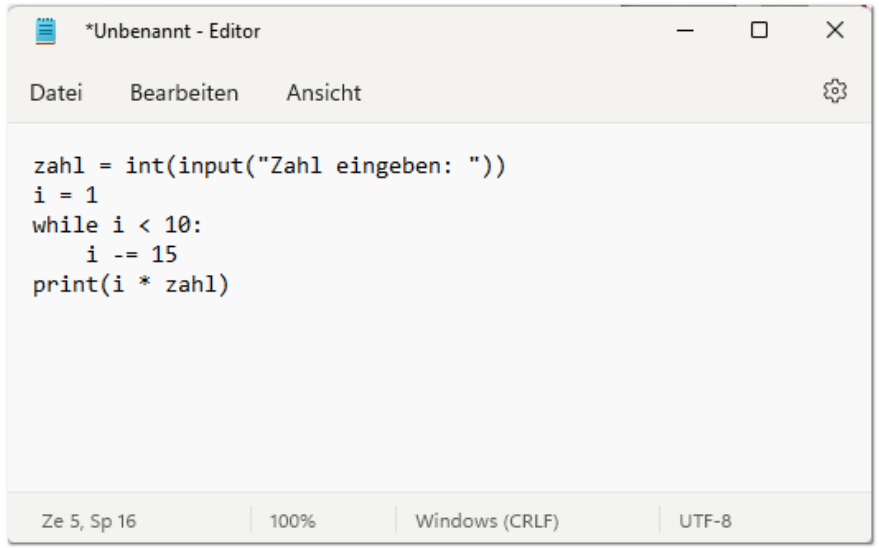
- Debugging im engeren Sinn: das Programm Zeile für Zeile abarbeiten
  - Und bei jeder Codezeile Überlegen, was das Resultat dieser Zeile sein soll

- Um einen Fehler in Ihrem Programm zu finden,
  - müssen Sie als erstes Ihr Hirn zu gebrauchen

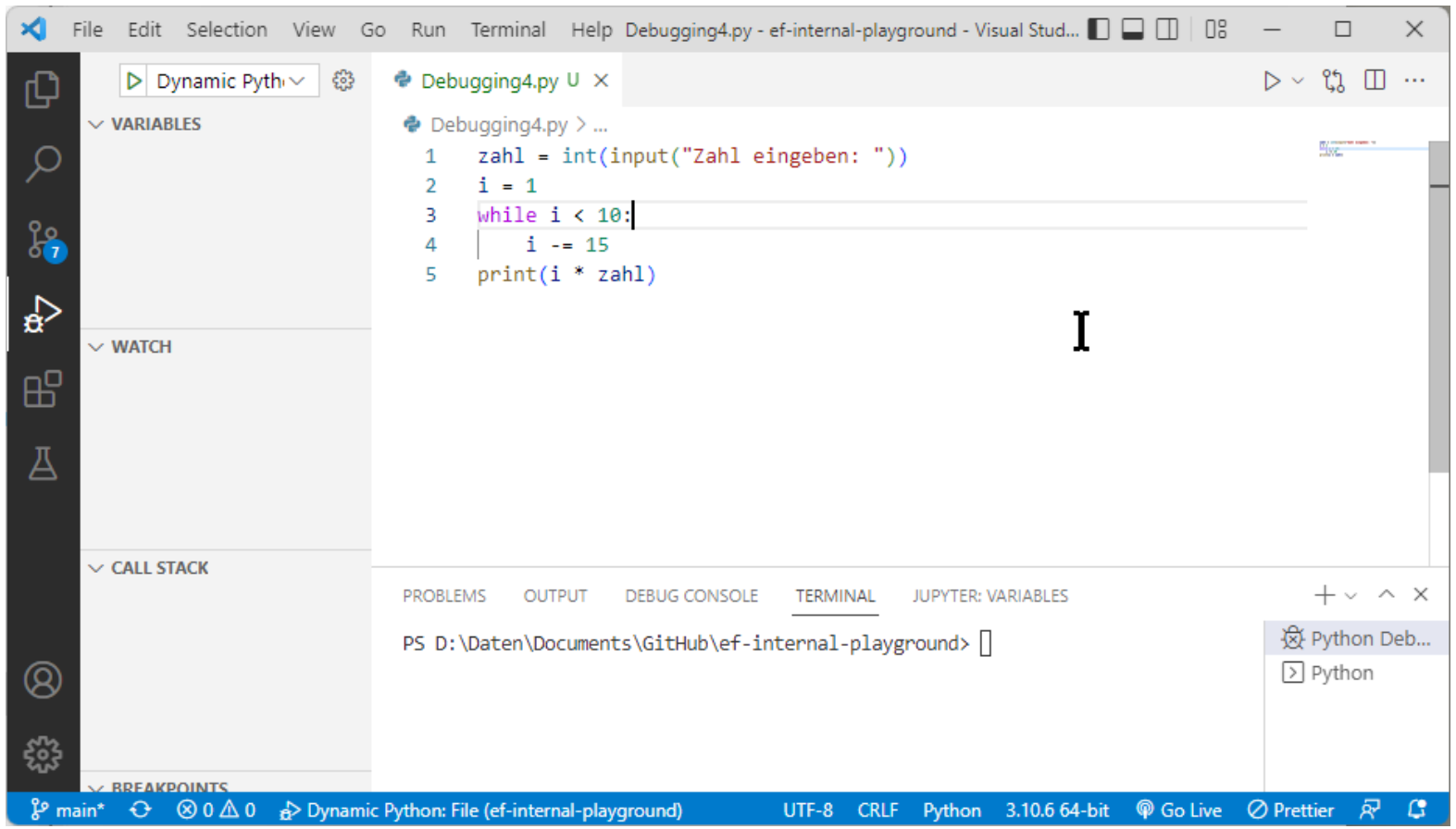
Debugging

- Dazu brauchen wir eine IDE (Visual Studio, Idle, Eclipse, IntelliJ, etc.)
- Oder einen Texteditor mit Erweiterung (VS Code, Atom, etc.)

- Notepad, WordPad, Word, Editor, etc. bringen nix



- Keine Debugging-Funktionen
- Kein Syntax-Highlighting
- Kein IntelliSense
- Keine Auswertung von Variablen



Fehlertypen

- Syntaxfehler und Semantikfehler

Ein guter Editor meldet das bereits beim Tippen

```
prinnt("Hallo")

print("Hallo"

for i in range(5):
print(i)

for i in range(5)
    print(i)
```

Fehlertypen

- Laufzeitfehler

Erst, wenn das Programm läuft, merkt es Fehler

```
zahl = input("Zahl eingeben: ")
print(54/zahl)
```

```
zahl = int(input("Zahl eingeben: "))
print(54/zahl)
```

Exception has occurred: ZeroDivisionError ×
division by zero

```
  File "C:\Users\chapp\Documents\GitHub\ef-internal-playground\zero.py", line 2, in <module>
    print(54/zahl)
```

Fehlertypen

- Logische Fehler

Da kommt häufig keine Fehlermeldung, sondern ein Resultat.

Aber stimmt es auch?

```
zahl = int(input("Zahl eingeben: "))
i = 10
while i < 10:
    print(i * zahl)
```

```
zahl = int(input("Zahl eingeben: "))
i = 1
while i < 10:
    print(i * zahl)
```

```
zahl = int(input("Zahl eingeben: "))
i = 1
while i < 10:
    print(i * zahl)
    i += 1
```

```
zahl = int(input("Zahl eingeben: "))
i = 1
while i < 10:
    i -= 1
print(i * zahl)
```

Fehlertypen

- Und hier?

- Sprachspezifisches Problem: dynamische Typisierung von Python

```
zahl = int(input("Zahl eingeben: "))
i = 1
while i < 10:
    print(i * zahl)
    j += 1
```

Was kommt da raus?

- `k = 5`
- `l = 7`
- `n = k * 1`
- `print(n)`

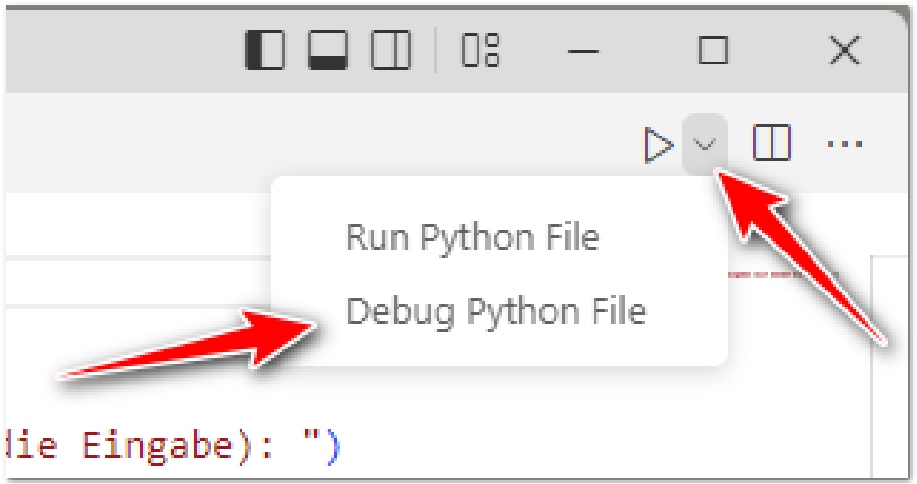
https://is.gd/utahew

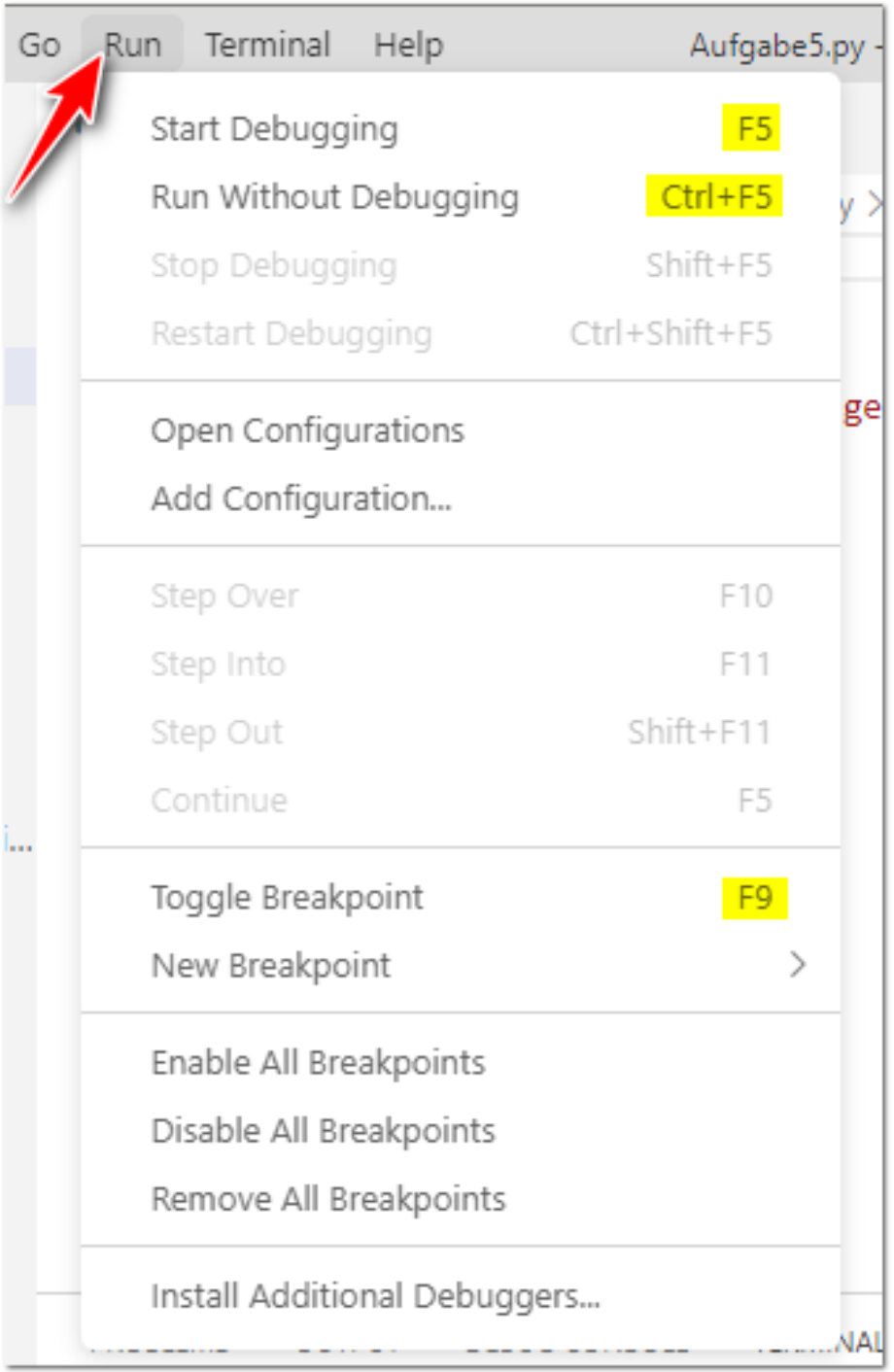
Kurze Demo in VS Code

Achtung, Thonny reagiert anders!

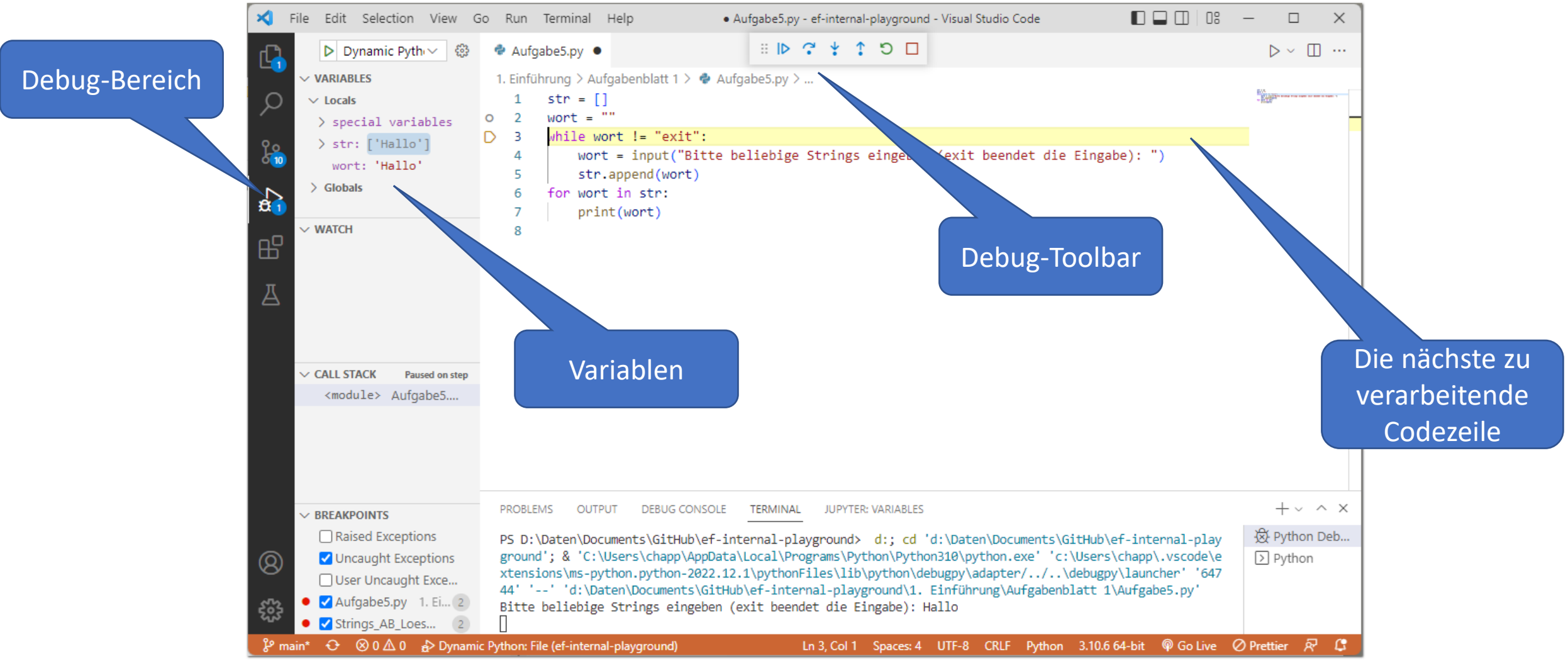
Debug Mode

- Nicht einfach auf den Play-Button klicken!





Das VS Code Fenster



Aufgaben

- Debbuging1.py

bis

- Debugging6.py

- Danach mit den Aufgaben zu Strings weiterfahren

- Die **Aufgabe 8** zu den Strings können Sie mit Debugging einfach lösen