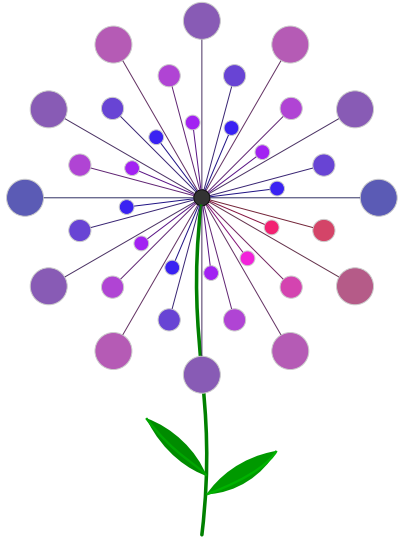


FreeFlower



Informatik: Programmieren

Kapitel 04: **Parameter**

Weshalb braucht es Parameter?

```
import turtle as t

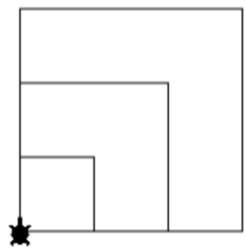
def quadrat_50():
    for _ in range(4):
        t.fd(50)
        t.rt(90)

def quadrat_100():
    for _ in range(4):
        t.fd(100)
        t.rt(90)

def quadrat_150():
    for _ in range(4):
        t.fd(150)
        t.rt(90)

quadrat_50()
quadrat_100()
quadrat_150()

t.done()
```



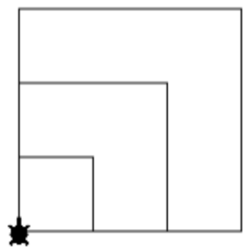
→ **Wunsch**: Ein einziger Befehl mit variabler (dynamischer) Länge!

Ein Befehl für viele Quadrate

```
import turtle as t

def quadrat(laenge):
    for _ in range(4):
        t.fd(laenge)
        t.rt(90)

quadrat(50)
quadrat(100)
quadrat(150)
```



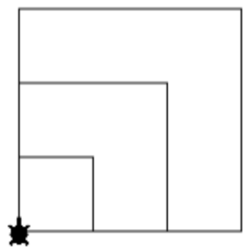
Ein Befehl für viele Quadrate

```
import turtle as t

def quadrat(laenge):
    for _ in range(4):
        t.fd(laenge)
        t.rt(90)

quadrat(50)
quadrat(100)
quadrat(150)
```

Übergabe



FreeFlower

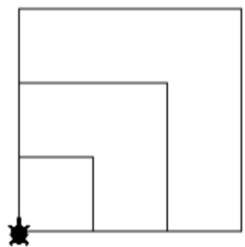
Ein Befehl für viele Quadrate

```
import turtle as t

def quadrat(laenge):
    for _ in range(4):
        t.fd(laenge)
        t.rt(90)

quadrat(50)
quadrat(100)
quadrat(150)
```

Übergabe



FreeFlower

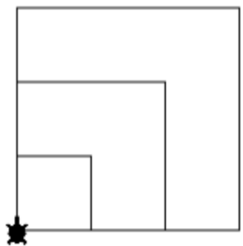
Ein Befehl für viele Quadrate

```
import turtle as t

def quadrat(laenge):
    for _ in range(4):
        t.fd(laenge)
        t.rt(90)

quadrat(50)
quadrat(100)
quadrat(150)
```

Übergabe



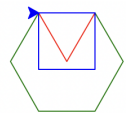
Mehrere Parameter

```
import turtle as t

def farb_vieleck(ecken, farbe):
    t.color(farbe)
    for _ in range(ecken):
        t.fd(50)
        t.rt(360 / ecken)

farb_vieleck(3,"red")
farb_vieleck(6,"green")
farb_vieleck(4,"blue")

t.done()
```



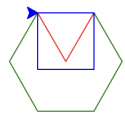
Mehrere Parameter

```
import turtle as t

def farb_vieleck(ecken, farbe):
    t.color(farbe)
    for _ in range(ecken):
        t.fd(50)
        t.rt(360 / ecken)

farb_vieleck(3,"red")
farb_vieleck(6,"green")
farb_vieleck(4,"blue")

t.done()
```



→ Man kann beliebig viele Parameter aneinanderreihen!



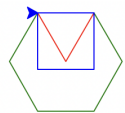
Mehrere Parameter

```
import turtle as t

def farb_vieleck(ecken, farbe):
    t.color(farbe)
    for _ in range(ecken):
        t.fd(50)
        t.rt(360 / ecken)

farb_vieleck(3,"red")
farb_vieleck(6,"green")
farb_vieleck(4,"blue")

t.done()
```



→ Man kann beliebig viele Parameter aneinanderreihen!

```
def meine_funktion(p_1, p_2, p_3, ...)
```



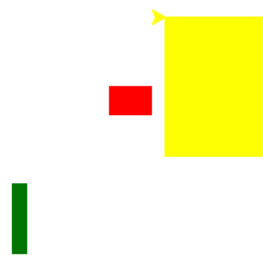
Mehrere Parameter

```
import turtle as t

def rechteck_farbig(...):
    ...

rechteck_farbig(...)
rechteck_farbig(...)
rechteck_farbig(...)
```

Welche Parameter werden für diese Definition verwendet?





Mehrere Parameter

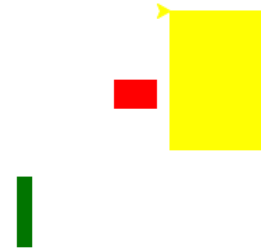
```
import turtle as t


def rechteck_farbig(x, y, farbe, hoehe, breite):
    # Start-Position setzen
    t.teleport(x, y)

    # Farbiges Rechteck zeichnen
    t.color(farbe)
    t.fillcolor(farbe)
    t.begin_fill()
    for _ in range(2):
        t.fd(breite)
        t.rt(90)
        t.fd(hoehe)
        t.rt(90)
    t.end_fill()


rechteck_farbig(50, 100, "red", 20, 30)
rechteck_farbig(-20, 30, "green", 50, 10)
rechteck_farbig(90, 150, "yellow", 100, 70)

t.done()
```



Lebensdauer von Variablen (Scope)

- **Parameter** und lokal definierte Variablen existieren nur **innerhalb** der Funktion

```
1 def meine_funktion(x):
2     y = x + 10
3     print(y)
4
5 meine_funktion(5)
6 print(y)  # Fehler
```

NameError: name 'y' is not defined



Lebensdauer von Variablen (Scope)

```python
def meine_funktion(x):
    y = x + 10
    print(y)

meine_funktion(5)
print(y)  # Fehler
```

NameError: name 'y' is not defined

- **Parameter** und lokal definierte Variablen existieren nur **innerhalb** der Funktion
- Ausserhalb der Funktion sind sie **nicht sichtbar**



Lebensdauer von Variablen (Scope)

```python
def meine_funktion(x):
    y = x + 10
    print(y)

meine_funktion(5)
print(y)  # Fehler
```

NameError: name 'y' is not defined

- **Parameter** und lokal definierte Variablen existieren nur **innerhalb** der Funktion
- Ausserhalb der Funktion sind sie **nicht sichtbar**
- **Vorteil**: Variablennamen können mehrfach verwendet werden



Lebensdauer von Variablen (Scope)

```python
def meine_funktion(x):
    y = x + 10
    print(y)

meine_funktion(5)
print(y)  # Fehler
```

NameError: name 'y' is not defined

- **Parameter** und lokal definierte Variablen existieren nur **innerhalb** der Funktion
- Ausserhalb der Funktion sind sie **nicht sichtbar**
- **Vorteil**: Variablennamen können mehrfach verwendet werden
- Dies nennt man ***Scope*** (Gültigkeitsbereich)



Lebensdauer von Variablen (Scope)

```
1 def meine_funktion(x):
2     y = x + 10
3     print(y)
4
5 meine_funktion(5)
6 print(y)  # Fehler
```

NameError: name 'y' is not defined

- **Parameter** und lokal definierte Variablen existieren nur **innerhalb** der Funktion
- Ausserhalb der Funktion sind sie **nicht sichtbar**
- **Vorteil**: Variablennamen können mehrfach verwendet werden
- Dies nennt man ***Scope*** (Gültigkeitsbereich)
- **Lebensdauer** muss beachtet werden: Sobald die Funktion verlassen wird, existieren die Parameter und lokalen Variablen nicht mehr



Variablen

Was sind **Parameter**?

- **Parameter** sind Variablen



Variablen

Was sind **Parameter**?

- **Parameter** sind Variablen
- Ein Wort, das einen ***veränderbaren*** (variablen) Wert enthält.



Variablen

Was sind **Parameter**?

- **Parameter** sind Variablen
- Ein Wort, das einen ***veränderbaren*** (variablen) Wert enthält.
- Variablenname darf (fast) frei gewählt werden



Variablen

Was sind **Parameter**?

- **Parameter** sind Variablen
- Ein Wort, das einen ***veränderbaren*** (variablen) Wert enthält.
- Variablenname darf (fast) frei gewählt werden
  - Name muss mit einem Buchstaben oder Unterstrich beginnen (nicht: `1meinname`)



FreeFlower

Variablen

Was sind **Parameter**?

- **Parameter** sind Variablen
- Ein Wort, das einen ***veränderbaren*** (variablen) Wert enthält.
- Variablenname darf (fast) frei gewählt werden
  - Name muss mit einem Buchstaben oder Unterstrich beginnen (nicht: `1meinname`)
  - Kein von Python reserviertes Wort (`def`, `if`, `for` etc.)



Variablen

Was sind **Parameter**?

- **Parameter** sind Variablen
- Ein Wort, das einen ***veränderbaren*** (variablen) Wert enthält.
- Variablenname darf (fast) frei gewählt werden
  - Name muss mit einem Buchstaben oder Unterstrich beginnen (nicht: `1meinname`)
  - Kein von Python reserviertes Wort (`def`, `if`, `for` etc.)
  - Falls zusammengesetzter Name: Trennen Sie die Teile durch Unterstriche (z.B.: `mein_name`)



Variablen

Was sind **Parameter**?

- **Parameter** sind Variablen
- Ein Wort, das einen ***veränderbaren*** (variablen) Wert enthält.
- Variablenname darf (fast) frei gewählt werden
  - Name muss mit einem Buchstaben oder Unterstrich beginnen (nicht: `1meinname`)
  - Kein von Python reserviertes Wort (`def`, `if`, `for` etc.)
  - Falls zusammengesetzter Name: Trennen Sie die Teile durch Unterstriche (z.B.: `mein_name`)
  - **Sinnvoller** und **beschreibender** Name!



Auftrag

- Empfehlung: Erstellen und testen Sie den Code direkt in den `CodeRunner-Aufgaben`
- „Aufgaben Funktionen 2“ auf Moodle
- Zeit: 30 Minuten
- Danach: Gemeinsame Besprechung der Aufgabe mit der Diagonale (Rautezeichen)



Wissen überprüfen

Was wird vom folgenden Programm ausgegeben?

```
def f(x):
    x = x + 10
    print(x)

a = 5
f(a)
print(a)
```



Wissen überprüfen

Was wird vom folgenden Programm ausgegeben?

```
def f(x):
    x = x + 10
    print(x)

a = 5
f(a)
print(a)
```

Ausgabe:

```
15
 5
```

Die Änderung an `x` (lokal) beeinflusst die Variable `a` (global) nicht.

