Informatik E2 Abels

Funktion

Funktion

Eine Funktion ist ein Codeblock, der erst ausgeführt wird, wenn er aufgerufen wird.

Eine Funktion kann sog. Parameter übergeben bekommen.

Eine Funktion kann einen sog. Rückgabewert zurückgeben.

Funktion

Name

```
def sageHallo():
    print("----------")
    print("Hallo Welt!")
    print("----------")

sageHallo()
sageHallo()
```

Terminal

```
----------
Hallo Welt!
----------
----------
Hallo Welt!
----------
```

Übung 1

Schreib ein Programm namens **Funktionen** mit folgenden Funktionen:

Funktionen.py

```
def schreibeMinusZeile():
  ...

def sageHallo():
  ...

def zaehleBisFuenfzig():
  for ...:
    print(..., end="")

def zaehleRueckwaertsAbZwanzig():
  ...
```

Terminal

```
--------------
--------------
Hallo
--------------
1, 2, ..., 50
--------------
20, 19, ..., 1
--------------
--------------
```

Rufe die Funktionen in der richtigen Reihenfolge (ggf. mehrfach) auf, um die gegebene Ausgabe zu erzeugen.

Übung 1

Funktionen.py

```python
def schreibeMinusZeile():
    print("--------------")

def sageHallo():
    print("Hallo")

def zaehleBisFuenfzig():
    for i in range(1, 50):
        print(str(i) + ", ", end="")
    print(50)

def zaehleRueckwaertsAbZwanzig():
    for i in range(20, 1, -1):
        print(str(i) + ", ", end="")
    print(1)
```

```python
schreibeMinusZeile()
schreibeMinusZeile()
sageHallo()
schreibeMinusZeile()
zaehleBisFuenfzig()
schreibeMinusZeile()
zaehleRueckwaertsAbZwanzig()
schreibeMinusZeile()
schreibeMinusZeile()
```

Funktion mit Parametern

Parameter

```
def sageHallo(name):
  print("Hallo, " + name + "!")

sageHallo("Adam")
sageHallo("Eva")
```

Terminal

```
Hallo, Adam!
Hallo, Eva!
```

Funktion mit Parametern

Default - Wert

```
def sageHallo(name, land = "Deutschland", alter):
  print("Hallo " + name + " (" + alter + ") aus " + land)

sageHallo("Adam", 31)
sageHallo("Eva", "USA", 42)
```

Terminal

```
Hallo Adam (31) aus Deutschland
Hallo Eva (42) aus USA
```

Übung 2

Erweitere dein Programm um folgende Funktionen:

Funktionen.py

```
def gibSternenzeileAus(n):
  ...

def gibAdditionAus(a, b):
  ...

def gibReihenfolgeAus(a, b, c):
  ...
```

Terminal

```
**********
*
**
***
****
3 + 5 = 8
0.1 + 5.2 = 5.3
**********
-5 <= 1 <= 23
2 <= 2 <= 3
**********
```

Rufe die Funktionen in der richtigen Reihenfolge (ggf. mehrfach) auf, um die gegebene Ausgabe zu erzeugen.

Übung 2

Funktionen.py

```python
def gibSternenzeileAus(n = 10):
    for _ in range(n):
        print("*", end="")
    print()

def gibAdditionAus(a, b):
    print(a, "+", b, "=", a+b)

def gibReihenfolgeAus(a, b, c):
    if a <= b:
        if b <= c:
            print(a, "<=", b, "<=", c)
        else:
            if a <= c:
                print(a, "<=", c, "<=", b)
            else:
                print(c, "<=", a, "<=", b)
    else:
        if a <= c:
            print(b, "<=", a, "<=", c)
        else:
            if b <= c:
                print(b, "<=", c, "<=", a)
            else:
                print(c, "<=", b, "<=", a)
```

```python
gibSternenzeileAus()
gibSternenzeileAus(1)
gibSternenzeileAus(2)
gibSternenzeileAus(3)
gibSternenzeileAus(4)
gibAdditionAus(3, 5)
gibAdditionAus(0.1, 5.2)
gibSternenzeileAus()
gibReihenfolgeAus(23, -5, 1)
gibReihenfolgeAus(2, 3, 2)
gibSternenzeileAus()
```

Funktion mit Rückgabewert

```python
def quadrat(n):
    return n * n

print(quadrat(3))
print(quadrat(-5))
print(quadrat(quadrat(2)))
```

Terminal

```
9
25
16
```

Übung 3

Erweitere dein Programm um folgende Funktionen, sodass es die gegebene Ausgabe erzeugt:

Funktionen.py

```python
def kreisflaeche(r):
  ...

def potenz(basis, exponent):
  ...

def ggt(a, b):
  ...

print(round(kreisflaeche(1.5), 2)
print(potenz(3, 5))
print(ggt(12, 12))
print(ggt(12, 17))
print(ggt(12, 18))
print(ggt(18, 12))
```

Terminal

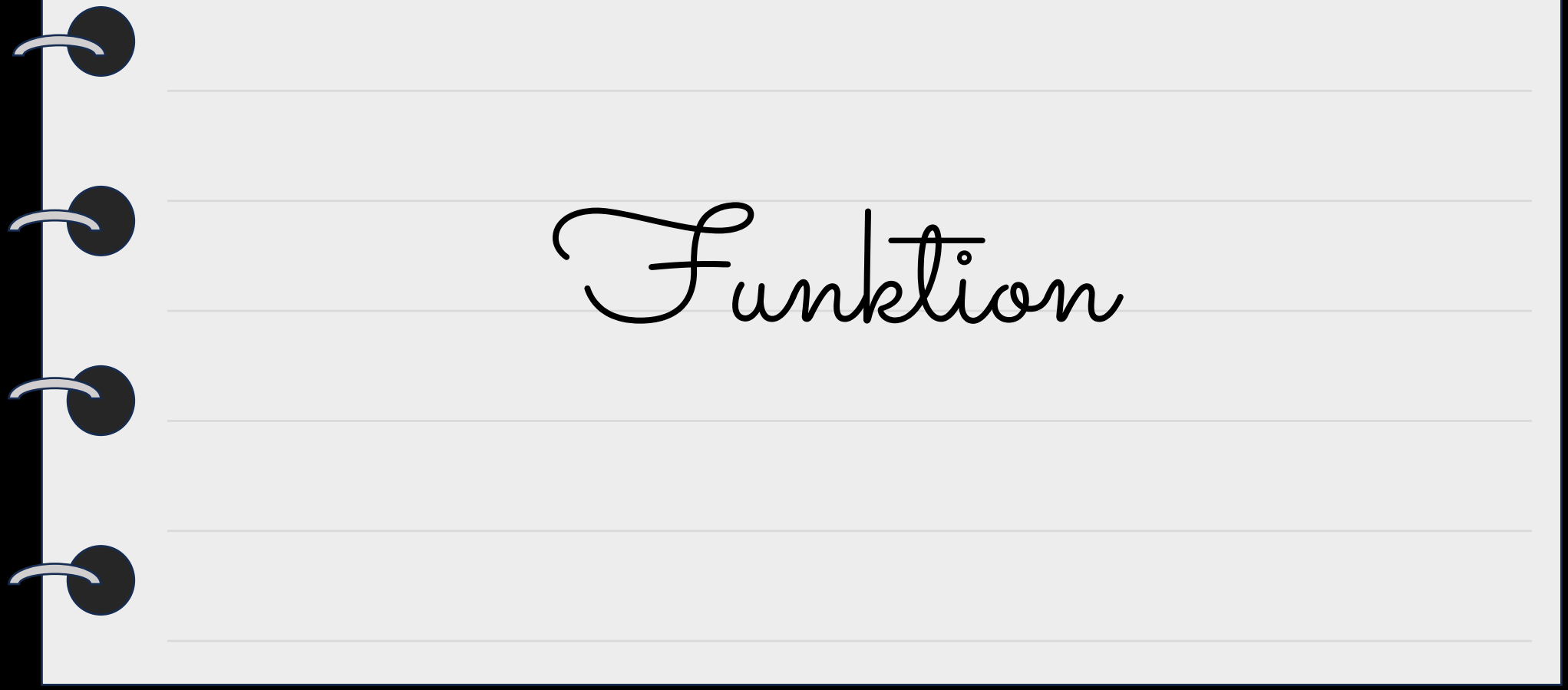
```
4.71
243
12
1
6
6
```

Übung 3

Funktionen.py

```python
import math

def kreisflaeche(r):
  return math.pi * r

def potenz(basis, exponent):
  result = 1
  for _ in range(exponent):
    result *= basis
  return result

def ggt(a, b):
  if a == b:
    return a
  if a > b:
    while b != 0:
      r = a % b
      a = b
      b = r
    return a
  else:
    return ggt(b, a)

print(round(kreisflaeche(1.5), 2))
print(potenz(3, 5))
print(ggt(12, 12))
print(ggt(12, 17))
print(ggt(12, 18))
print(ggt(18, 12))
```

Tagebucheintrag



Wochenübung

Schreibe ein Programm **Primzahl**, das testet, ob eine Zahl eine Primzahl ist.

Ist die Zahl kleiner 2, kann `False` zurückgegeben werden. Sonst werden alle Zahlen bis zur Wurzel der Zahl durchlaufen und überprüft, ob sie ein Teiler der Zahl sind. Falls ja, kann `False` zurückgegeben werden. Sonst wird am Ende `True` zurückgegeben.

Primzahl.py

```python
# Eingabe
a = ...

# Verarbeitung
def istTeiler(zahl, teiler):
  ...

def wurzel(x):
  ...

def istPrim(n):
  ...

# Ausgabe
if istPrim(a):
  print(...)
else:
  print(...)
```

Terminal

```
Zahl: 5
5 ist eine Primzahl!
```