

Unser **Kunde** „Leselust & Wissen GmbH“ aus dem **Buchhandel** plant einen „**Tag der offenen Buchhandlung**“. Jeder Mitarbeiter übernimmt an diesem Tag eine **Funktion**:

- Mitarbeiter 1: Kunden begrüßen
- Mitarbeiter 2: Kassieren
- Mitarbeiter 3: Kunden zur Lesung anmelden



Ralf Meier -
Abteilungsleiter
Software Code
Solutions

Um den **Ablauf** für den „Tag der offenen Buchhandlung“ zu **simulieren**, wurden wir beauftragt, ein **Programm** zu erstellen.



KUNDEN BEGRÜSSEN



Das sieht schon mal **gut** aus! 😊 Unser **Azubi** aus der Entwicklungsabteilung hat außerdem bereits einen **Vorschlag** für die **Kassierfunktion** gemacht:



Ralf Meier -
Abteilungsleiter
Software Code
Solutions



```
anzahl_buecher = 24
```

```
# 2. Kassieren
```

```
# > Es werden 3 Kunden abkassiert und
```

```
if anzahl_buecher > 0:
```

```
    anzahl_buecher -= 1
```

```
    print("Vielen Dank für Ihren Einkauf bei Leselust & Wissen GmbH!")
```

```
else:
```

```
    print("Es sind leider keine Bücher mehr vorrätig.")
```

```
if anzahl_buecher > 0:
```

```
    anzahl_buecher += 1
```

```
    print("Vielen Dank für Ihren Einkauf bei Leselust & Wissen GmbH!")
```

```
else:
```

```
    print("Es sind leider keine Bücher mehr vorrätig.")
```

```
if anzahl_buecher > 0:
```

```
    anzahl_buecher -= 1
```

```
    print("Vielen Dank für Ihren Einkauf bei Leselust & Wissen GmbH!")
```

```
else:
```

```
    print("Es sind leider keine Bücher mehr vorrätig.")
```

Doppelten Programmcode
zu einer Funktion
zusammenfassen.



!

!

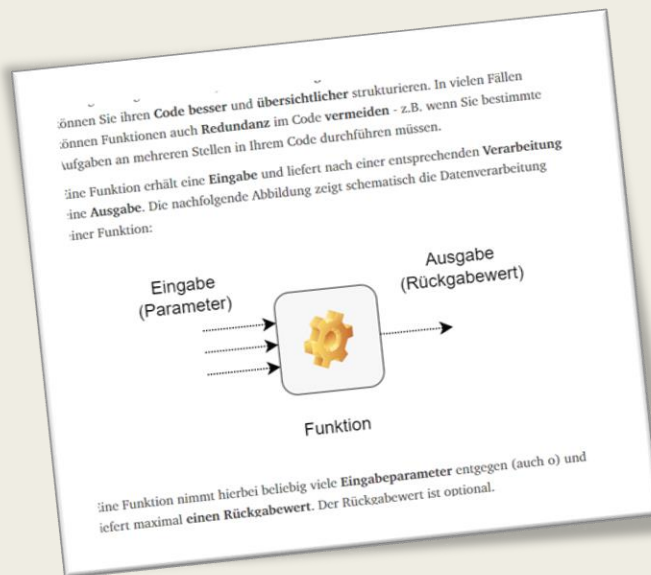


Ralf Meier -
Abteilungsleiter
Software Code
Solutions



Informieren zu Funktionen

- Informieren Sie sich in Einzelarbeit zu Funktionen mithilfe des Online-Infomaterials:



[telegra.ph/
EF-12-11-2](https://telegra.ph/EF-12-11-2)

or




Einzelarbeit

 10 Minuten

Erklärpuzzle in Partnergruppe

- Tauschen Sie sich in Ihrer **Partnergruppe** über Ihre gewonnenen Erkenntnisse zum Thema **Funktionen** aus, indem Sie das **Erklärpuzzle** vervollständigen.

Definition und Aufruf einer Funktion:

```
# Funktionsdefinition:
# Funktionsname: 1
# Eingabeparameter: 2
def Maximum(zahl1, zahl2):
    max = zahl1
    if zahl2 > max:
        max = zahl2
    return max 3
# Start des Hauptprogramms ab hier:
# Aufruf der Funktion: 2
ergebnis = Maximum(5, 3) 3
print(ergebnis) 1
```

Ordnen Sie die verschiedenen **Kategorien** und **Aussagen** den Nummern 1 bis 4 zu:

Kategorie	Nummer
Funktionsname	1
Eingabeparameter	2
Funktionsrumpf	3
Rückgabewert	4

Folge von Anweisungen, die festlegen, welche Aufgabe die Funktion bei ihrem Aufruf erfüllt.

Beschreibt die Daten, die beim Funktionsaufruf an die Funktion übertragen werden.

Jeder Parameter ist durch einen Namen zu definieren.

Legt fest, welches Ergebnis der Funktionsaufruf hat.

Ein frei wählbarer, aussagekräftiger Name.

Das Ergebnis wird mit der return-Anweisung zurückgeliefert und die Funktion wird verlassen.

Der Rückgabewert der Funktion ist optional. Eine Funktion muss nicht grundsätzlich ein Ergebnis liefern.

Beim Funktionsaufruf werden die Eingabeparameter in die Variablen an der gleichen Position in der Funktion kopiert.

- Laden Sie sich hierfür das **Erklärpuzzle** (PowerPoint) von **github** herunter:
Funktionen > Lernsituationen > Erklärpuzzle Funktionen

Erklärpuzzle in Partnergruppe



Definition und Aufruf einer **Funktion**:



Ordnen Sie die verschiedenen **Kategorien** und **Aussagen** den Nummern 1 bis 4 zu:

Funktionsdefinition:

Funktions-
name Eingabeparameter

1 2
`def Maximum(zahl1, zahl2):`

`max = zahl1`

`if zahl2 > max:`
`max = zahl2`

`return max` 3 Rückgabe-
wert

4
Funktions-
rumpf

Start des Hauptprogramms ab hier:

Aufruf der Funktion: 2

3 `ergebnis = Maximum(5, 3)`
`print(ergebnis)` 1

1	<div></div> <div></div>
2	<div></div> <div></div> <div></div> <div></div>
3	<div></div> <div></div> <div></div> <div></div>
4	<div></div> <div></div>

Funktionsname

Eingabeparameter

Funktionsrumpf

Rückgabewert

Folge von Anweisungen, die festlegen, welche Aufgabe die Funktion bei ihrem Aufruf erfüllt.

Beschreibt die Daten, die beim Funktionsaufruf an die Funktion übertragen werden.

Jeder Parameter ist durch einen Namen zu definieren.

Legt fest, welches Ergebnis der Funktionsaufruf hat.

Ein frei wählbarer, aussagekräftiger Name.

Das Ergebnis wird mit der return-Anweisung zurückgeliefert und die Funktion wird verlassen.

Der Rückgabewert der Funktion ist optional. Eine Funktion muss nicht grundsätzlich ein Ergebnis liefern.

Beim Funktionsaufruf werden die Eingabeparameter in die Variablen an der gleichen Position in der Funktion kopiert.

Erklärpuzzle in Partnergruppe

- Tauschen Sie sich in Ihrer **Partnergruppe** über Ihre gewonnenen Erkenntnisse zum Thema **Funktionen** aus, indem Sie das **Erklärpuzzle** vervollständigen.

Definition und Aufruf einer Funktion:

```
# Funktionsdefinition:
# Funktionsname: 1, Eingabeparameter: 2
def Maximum(zahl1, zahl2):
    max = zahl1
    if zahl2 > max:
        max = zahl2
    return max 3
# Start des Hauptprogramms ab hier:
# Aufruf der Funktion: 2
ergebnis = Maximum(5, 3) 3
print(ergebnis) 1
```

Ordnen Sie die verschiedenen **Kategorien** und **Aussagen** den Nummern 1 bis 4 zu:

-
-
-
-

Funktionsname	Eingabeparameter
Funktionsrumpf	Rückgabewert

Folge von Anweisungen, die festlegen, welche Aufgabe die Funktion bei ihrem Aufruf erfüllt.

Beschreibt die Daten, die beim Funktionsaufruf an die Funktion übertragen werden.

Jeder Parameter ist durch einen Namen zu definieren.

Legt fest, welches Ergebnis der Funktionsaufruf hat.

Ein frei wählbarer, aussagekräftiger Name.

Das Ergebnis wird mit der return-Anweisung zurückgeliefert und die Funktion wird verlassen.

Der Rückgabewert der Funktion ist optional. Eine Funktion muss nicht grundsätzlich ein Ergebnis liefern.

Beim Funktionsaufruf werden die Eingabeparameter in die Variablen an der gleichen Position in der Funktion kopiert.

- Laden Sie sich hierfür das **Erklärpuzzle** (PowerPoint) von **github** herunter:
Funktionen > Lernsituationen > Erklärpuzzle Funktionen

Überprüfen Sie Ihre Ergebnisse



Starten der
Bildschirmpräsentation
mit



Definition und Aufruf einer **Funktion**:



Ordnen Sie die verschiedenen **Kategorien**
und **Aussagen** den Nummern 1 bis 4 zu:

Funktionsdefinition:

Funktions-
name Eingabeparameter

def Maximum(zahl1, zahl2):

max = zahl1

if zahl2 > max:
 max = zahl2

return max

4
Funktions-
rumpf

Start des Hauptprogramms ab hier:

Aufruf der Funktion:

ergebnis = Maximum(5, 3)

print(ergebnis)

1	<input type="text"/>
2	<input type="text"/>
3	<input type="text"/>
4	<input type="text"/>



DEFINITION UND AUFRUF EINER FUNKTION



PLANUNG



PROGRAMM ERSTELLEN!

1. KUNDEN BEGRUESSEN
2. KASSIEREN
3. KUNDEN ZUR LESUNG ANMELDEN

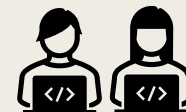


Programmcode mit Funktionen in Partnerarbeit verbessern

- Verbessern Sie den **Programmcode** Ihres Azubi-Kollegen, indem Sie den Copy-Paste-Programmcode in **Funktionen** auslagern.
- Definieren Sie hierfür im **oberen Programmteil** geeignete Funktionen und rufen Sie diese mehrfach im **unteren Programmteil** auf.

Programmcode-
Vorlage

 
[pastebin.com/
RXmCh1vp](https://pastebin.com/RXmCh1vp)



Partnerarbeit



15 Minuten

VORTEILE VON FUNKTIONEN

