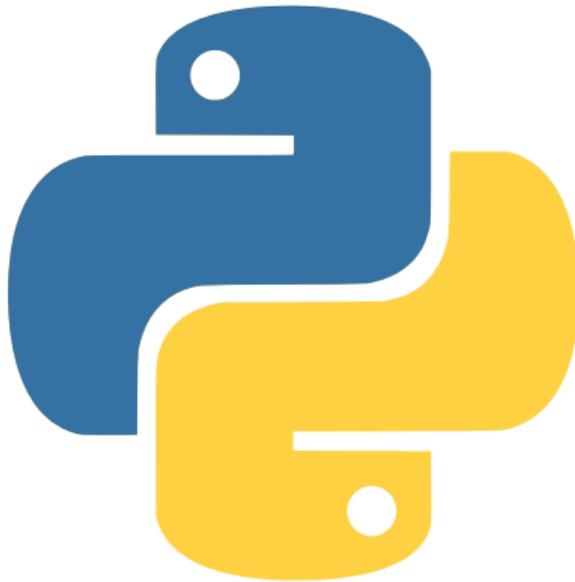




Universität Stuttgart

Projekt digit@L – BOOST. SKILLS. SUPPORT.



Dominik
Göddeke

Programmierkurs Python

Defaultwerte für
Funktionsargumente

Defaultwerte für Funktionsargumente

Motivation

- Start mit einem bereits **bekanntem Beispiel**

Motivation

- Start mit einem bereits **bekanntem Beispiel**
- Gegeben: Gravitationskonstante g , Startgeschwindigkeit v_0

Motivation

- Start mit einem bereits **bekanntem Beispiel**
- Gegeben: Gravitationskonstante g , Startgeschwindigkeit v_0
- Dann Höhe eines Objekts bei einem Wurf senkrecht nach oben zur Zeit t :

$$y(t) := v_0 t - \frac{1}{2} g t^2$$

Motivation

- Start mit einem bereits **bekanntem Beispiel**
- Gegeben: Gravitationskonstante g , Startgeschwindigkeit v_0
- Dann Höhe eines Objekts bei einem Wurf senkrecht nach oben zur Zeit t :

$$y(t) := v_0 t - \frac{1}{2} g t^2$$

- **Realisierung in Python**

```
def y(t, v0, g):  
    y = v0*t - 0.5*g*t*t  
    return y
```

```
print(y(0.01, 0, 9.81))
```

Motivation

$$y(t) := v_0 t - \frac{1}{2} g t^2$$

Motivation

$$y(t) := v_0 t - \frac{1}{2} g t^2$$

- Klar: Gravitationskonstante $g = 9.81 \frac{m}{s^2}$ faktisch konstant

Motivation

$$y(t) := v_0 t - \frac{1}{2} g t^2$$

- Klar: Gravitationskonstante $g = 9.81 \frac{m}{s^2}$ faktisch konstant
- Solange wir auf der Erde bleiben

Motivation

$$y(t) := v_0 t - \frac{1}{2} g t^2$$

- Klar: Gravitationskonstante $g = 9.81 \frac{m}{s^2}$ faktisch konstant
- Solange wir auf der Erde bleiben
- Außerdem: $v_0 = 0$ sinnvoller Standardwert, Beginn ohne Bewegung

Umsetzung in Python

- Angabe eines **Default-Werts**, mit folgender Syntax

```
def function_name(arg1, arg2, arg3=default_val3, arg4=default_val4, ...)
```

Umsetzung in Python

- Angabe eines **Default-Werts**, mit folgender Syntax

```
def function_name(arg1, arg2, arg3=default_val3, arg4=default_val4, ...)
```

- **Variablen mit Defaultwert**: beim Aufruf der Funktion nicht nötig

Umsetzung in Python

- Angabe eines **Default-Werts**, mit folgender Syntax

```
def function_name(arg1, arg2, arg3=default_val3, arg4=default_val4, ...)
```

- **Variablen mit Defaultwert**: beim Aufruf der Funktion nicht nötig
- **Überschreibung des Defaultwerts** selbstverständlich möglich

Umsetzung in Python

- Angabe eines **Default-Werts**, mit folgender Syntax

```
def function_name(arg1, arg2, arg3=default_val3, arg4=default_val4, ...)
```

- **Variablen mit Defaultwert**: beim Aufruf der Funktion nicht nötig
- **Überschreibung des Defaultwerts** selbstverständlich möglich
- Erhöht **Bequemlichkeit** und **Nutzbarkeit** bei Funktionsdefinition und -aufruf

Umsetzung in Python

- Angabe eines **Default-Werts**, mit folgender Syntax

```
def function_name(arg1, arg2, arg3=default_val3, arg4=default_val4, ...)
```

- **Variablen mit Defaultwert**: beim Aufruf der Funktion nicht nötig
- **Überschreibung des Defaultwerts** selbstverständlich möglich
- Erhöht **Bequemlichkeit** und **Nutzbarkeit** bei Funktionsdefinition und -aufruf
- Ohne Verlust an Flexibilität

Umsetzung in Python

- Angabe eines **Default-Werts**, mit folgender Syntax

```
def function_name(arg1, arg2, arg3=default_val3, arg4=default_val4, ...)
```

- **Variablen mit Defaultwert**: beim Aufruf der Funktion nicht nötig
- **Überschreibung des Defaultwerts** selbstverständlich möglich
- Erhöht **Bequemlichkeit** und **Nutzbarkeit** bei Funktionsdefinition und -aufruf
- Ohne Verlust an Flexibilität
- Typ eines Standardwertes: Festgelegt durch Definition

Umsetzung in Python

- Angabe eines **Default-Werts**, mit folgender Syntax

```
def function_name(arg1, arg2, arg3=default_val3, arg4=default_val4, ...)
```

- **Variablen mit Defaultwert**: beim Aufruf der Funktion nicht nötig
- **Überschreibung des Defaultwerts** selbstverständlich möglich
- Erhöht **Bequemlichkeit** und **Nutzbarkeit** bei Funktionsdefinition und -aufruf
- Ohne Verlust an Flexibilität
- Typ eines Standardwertes: Festgelegt durch Definition
- Wichtig: **Reihenfolge**, erst die nicht-Defaultvariablen, dann die mit Default

Umsetzung im Beispielcode

Zweites Beispiel: Näherung der e-Funktion

Zweites Beispiel: Näherung der e-Funktion

- Beispiel aus der Schulmathematik, mal wieder

Zweites Beispiel: Näherung der e-Funktion

- Beispiel aus der Schulmathematik, mal wieder
- Näherung von e^x durch Abschneiden einer Reihe, d.h. endliche statt unendliche Summe

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!} \approx \sum_{i=0}^N \frac{x^k}{k!}$$

Zweites Beispiel: Näherung der e-Funktion

- Beispiel aus der Schulmathematik, mal wieder
- Näherung von e^x durch Abschneiden einer Reihe, d.h. endliche statt unendliche Summe

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!} \approx \sum_{i=0}^N \frac{x^k}{k!}$$

- Hier als Funktion, mit Defaultwerten für Toleranz als Abbruchkriterium und Maximalwert für N

Zweites Beispiel: Näherung der e-Funktion

- Beispiel aus der Schulmathematik, mal wieder
- Näherung von e^x durch Abschneiden einer Reihe, d.h. endliche statt unendliche Summe

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!} \approx \sum_{i=0}^N \frac{x^k}{k!}$$

- Hier als Funktion, mit Defaultwerten für Toleranz als Abbruchkriterium und Maximalwert für N
- Gewissermaßen eine eigene Version von `math.exp()` mit Kontrolle über die Genauigkeit

So geht das im Code

Impressum, Danksagung und Quellen



Stiftung
Innovation in der
Hochschullehre



Gefördert durch die Stiftung Innovation in der Hochschullehre im Rahmen des Projekts digit@L, <https://stiftung-hochschullehre.de>

Gefördert mit Mitteln der Deutschen Forschungsgemeinschaft (EXC 2075 - 390740016) im Rahmen der Exzellenzstrategie

Autor: Dominik Göddeke, IANS, Universität Stuttgart



Weitere Quellen:

- Logos Universität Stuttgart, IANS, SimTech: Universität Stuttgart, alle Rechte vorbehalten
- Logo Python: <https://freesvg.org/387>, CC-0
- Logo Stiftung: Stiftung Innovation in der Hochschullehre, alle Rechte vorbehalten
- Logo ZOERR: Universität Tübingen, alle Rechte vorbehalten



Veröffentlicht auf dem Zentralen OER Repository Baden-Württemberg, <https://www.zoerr.de>