

Python

Verarbeitung - Folge



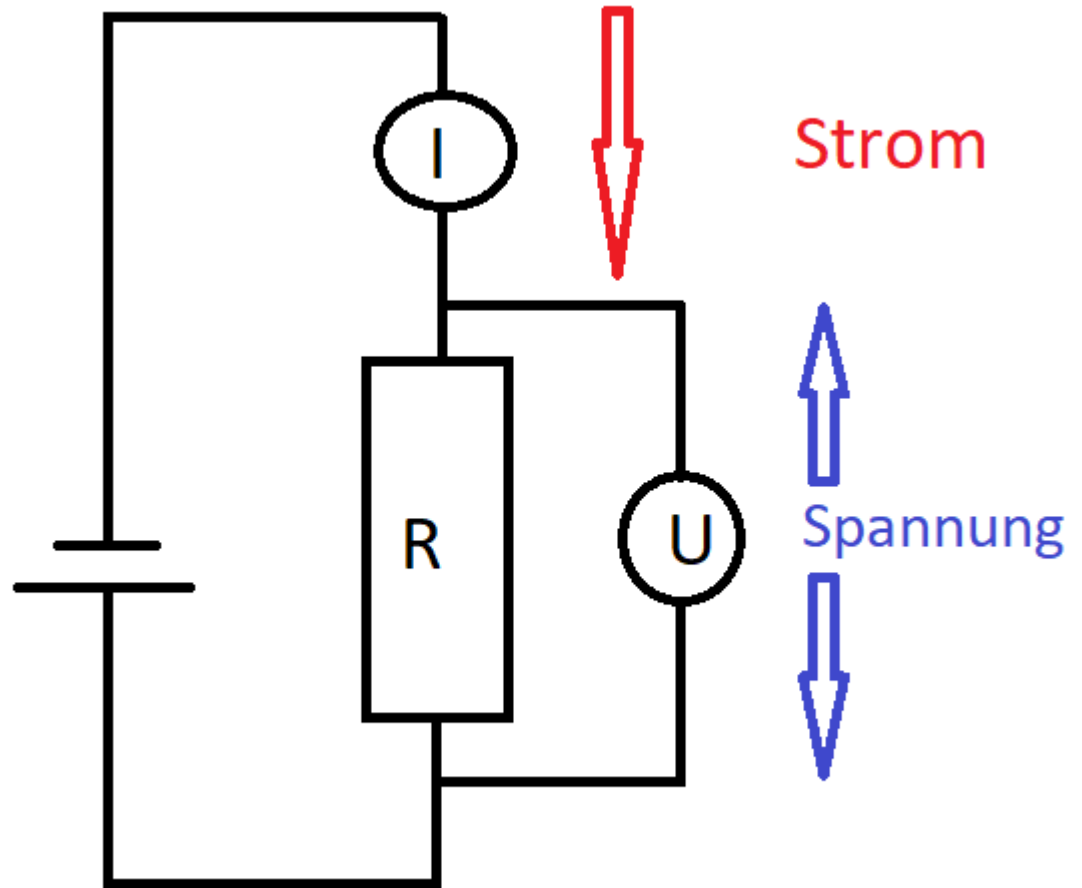
Bildquelle: <https://www.python.org/>

Einfaches Python-Programm

Mit einem Programm soll
die elektrische
Spannung **U**
aus dem Produkt von
Strom * Widerstand
berechnet werden:

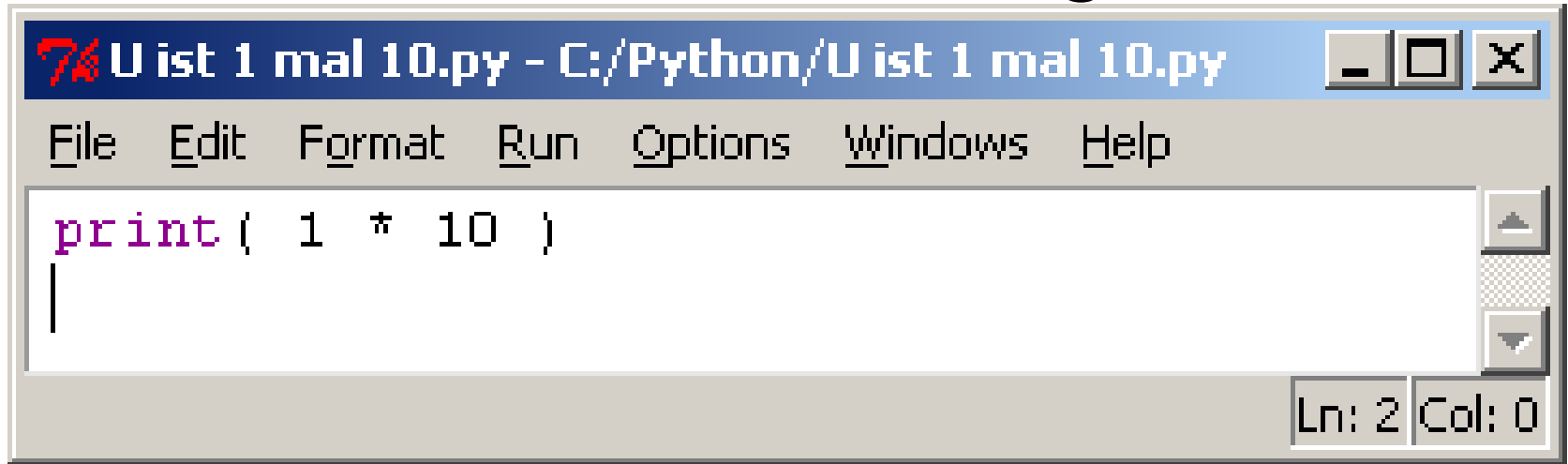
Formel: **U = I * R**

Der Strom beträgt 1 A
und der Widerstand 10
Ohm.



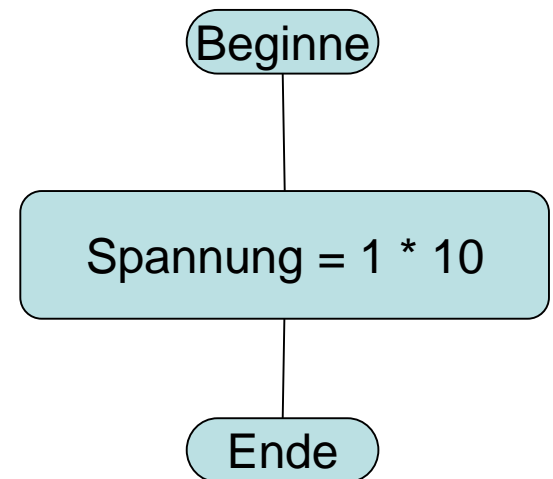
Einfaches Python-Programm

Verarbeitung



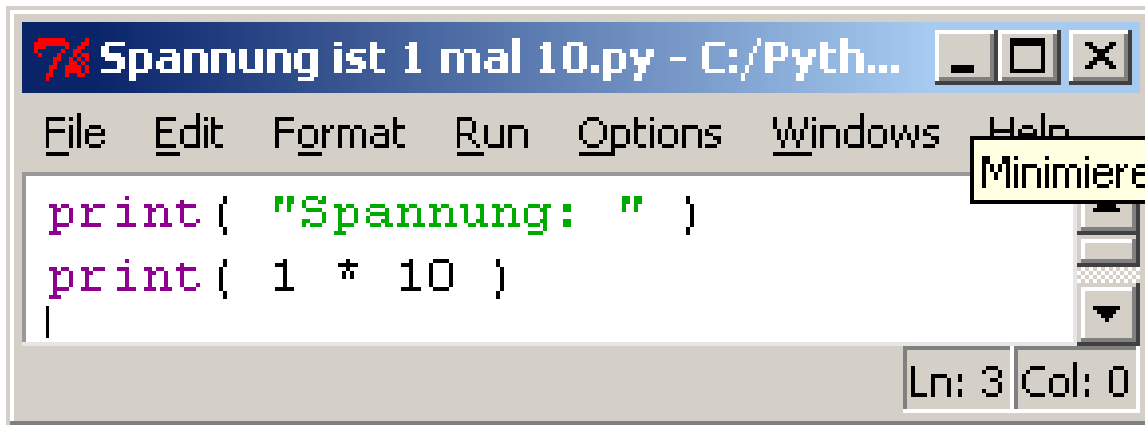
```
U ist 1 mal 10.py - C:/Python/U ist 1 mal 10.py
File Edit Format Run Options Windows Help
print( 1 * 10 )
|
Ln: 2 Col: 0
```

```
>>>
10
>>>
```



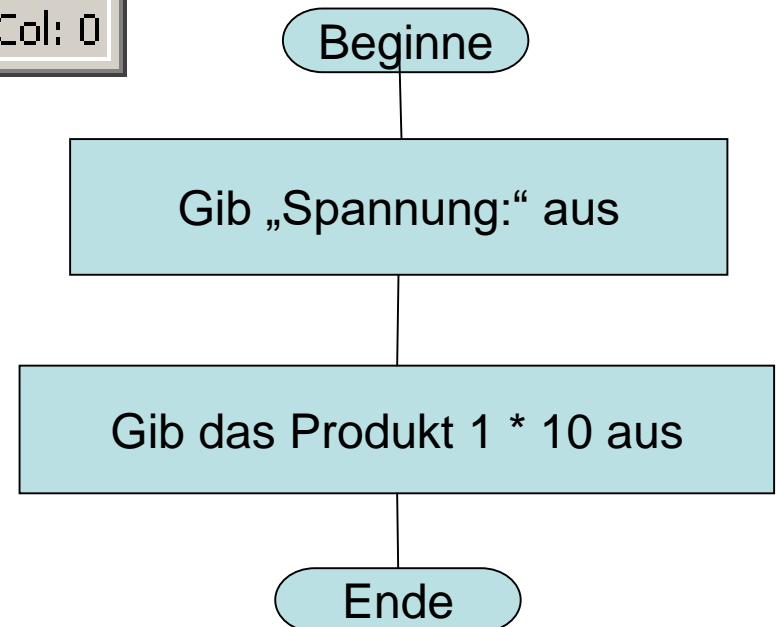
Einfaches Python-Programm - Folge

Über dem Ergebniswert 10 soll die Überschrift „Spannung:“ stehen.



```
print( "Spannung: " )
print( 1 * 10 )
```

```
>>>
Spannung:
10
>>>
```



Einfaches Python-Programm

Ein Programm wird leichter lesbar und dadurch verständlicher, wenn man im Programm treffend benannte Variablen verwendet.

```
ich_bin_eine_Variable = "ich bin ein Zeichenkettenwert"
```

```
print( "nun wird der Inhalt der Variablen ausgegeben" )  
print( ich_bin_eine_Variable )
```

```
>>>
```

```
nun wird der Inhalt der Variablen ausgegeben  
ich bin ein Zeichenkettenwert
```

```
>>>
```

Einfaches Python-Programm

Ein Programm wird viel leichter lesbar und verständlicher, wenn man im Programm Variablen verwendet.

ich_bin_eine_Variable = "ich bin ein Zeichenkettenwert"

Das ist der Name
der Variablen.

Das ist der
Zuweisungsoperator.

Das ist der Wert,
welcher der Variablen
zugewiesen wird.

Merke: Eine Variable ist ein Behälter, der einen Inhalt hat.

Einfaches Python-Programm

Ein Programm wird viel leichter lesbar und verständlicher, wenn man im Programm Variablen verwendet.

Wo die Daten herkommen bezeichnet man als **Datenquelle**.

`ich_bin_eine_Variable` = "ich bin ein Zeichenkettenwert"

Wo die Daten hingehen bezeichnet man als **Datensenke**.

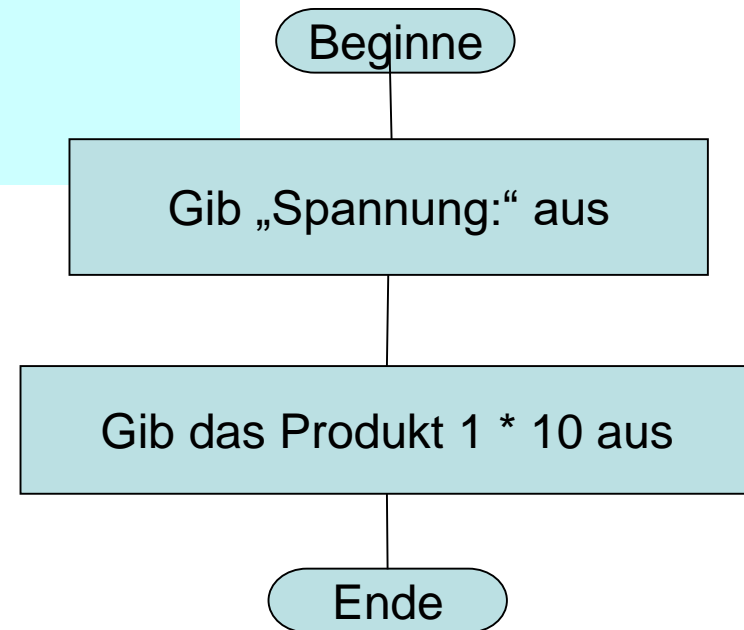
Der Zuweisungsoperator besagt, dass ein **Datentransport** von der **Quelle** zur **Senke** erfolgen soll.

Einfaches Python-Programm mit Variablen

```
strom      = 1
widerstand = 10
spannung   = strom * widerstand

print( "Spannung: " )
print( spannung      )
```

```
>>>
Spannung:
10
>>>
```



Datentypen von Zahlen in Python

Python kennt zwei eingebaute Datentypen für Zahlen:

- Ganzzahlen (Integer) z.B. 12345
- Fließkommazahlen (reelle Zahlen, float):
z.B. 3.14159 oder 17.3e+02

Datentypen von Zahlen in Python

Wichtige Zahlen haben die Formate
Ganzzahl oder Gleitkommazahl

Ganzzahl
print (1)

```
>>>  
1
```

Gebrochene Zahl, intern eine Gleitkommazahl
print (3.14)

```
3.14  
>>>
```

Datentypen von numerischen Variablen

Jeder Datenwert hat einen **Datentyp**.

z. B. der **Wert 1** hat den Datentyp **Ganzzahl**

z. B. der **Wert 3.14** hat den Datentyp **Gleitpunktzahl**

- Der aktuelle Datentyp einer Variable ist der Datentyp des Wertes, der ihr gerade zugewiesen wurde.
- Bei Python kann der Typ einer Variablen von Zuweisung zu Zuweisung wechseln.
- In Python werden die Variablen somit dynamisch typisiert.

Datentypen von numerischen Variablen

```
# Datentypen von Variablen
```

```
# Der Datentyp wird bei der Wertzuweisung festgelegt
```

```
ganzzahl          = 1      # ganzzahlige Konstante
```

```
gleitpunktzahl = 3.14 # Gleitkommakonstante
```

```
print( ganzzahl      )
```

```
print( gleitpunktzahl )
```

```
>>>
```

```
1
```

```
3.14
```

```
>>>
```

Datentypen von Variablen

Dynamische Datentypisierung

```
variable = 1          # Datentyp ist Ganzzahl  
print( variable )
```

```
variable = "hallo"    # Datentyp ist Zeichenkette  
print( variable )
```

>>>

1

hallo

>>>

Die dynamische Typisierung von Variablen ist komfortabel bei der Programmnederschrift.

Sie birgt jedoch die Gefahr, dass man in einem längeren Programm die Übersicht verliert und unzulässige Operationen mit der Variablen ausführt.