

```
a = "Die Zahl lautet"  
b = 5
```

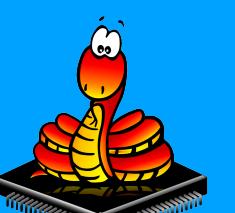
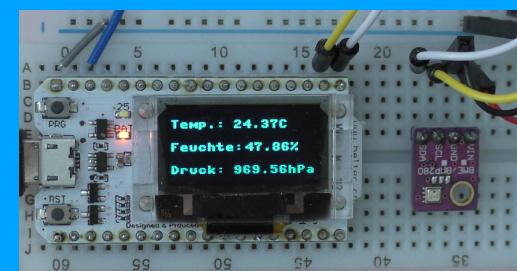
a und b werden als globale Variablen erzeugt.

```
def ausgabe1():  
    print("**** Ausgabe 1 ****")  
    print(a,b)  
    print()
```

Die globalen Variablen a und b werden ausgegeben.

```
print("Globale Werte: a = {} , b = {}".format(a,b))  
ausgabe1()  
print("Globale Werte: a = {} , b = {}".format(a,b))
```

Hier werden immer die globalen Variablen a und b ausgegeben.



```
a = "Die Zahl lautet"  
b = 5
```

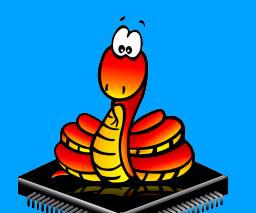
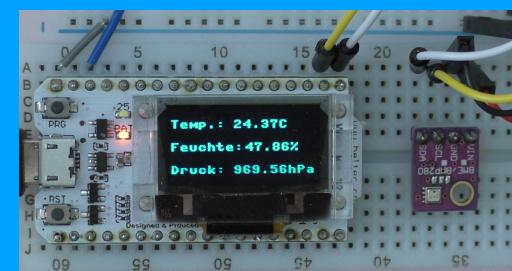
a und b werden als globale Variablen erzeugt.

```
def ausgabe2 ():  
    print("**** Ausgabe 2 ****")  
    a = "Eine andere Zahl:"  
    b = 7  
    print(a,b)  
    print()
```

Es werden neue lokale Variablen a und b angelegt.
Diese sind unabhängig von den globalen variablen a und b.

```
ausgabe2()  
print("Globale Werte: a = {}, b = {}".format(a,b))
```

Die globalen Variablen a und b wurden nicht verändert.



```
a = "Die Zahl lautet"  
b = 5
```

a und b werden als globale Variablen erzeugt.

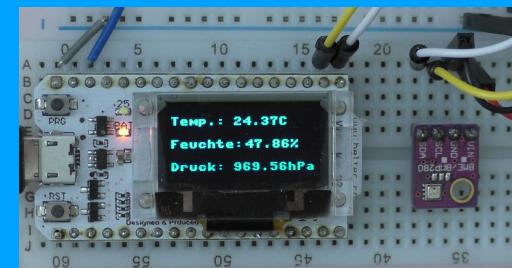
```
def ausgabe2 ():  
    print("**** Ausgabe 2 ****")  
    global a,b  
    a = "Eine andere Zahl:"  
    b = 7  
    print(a,b)  
    print()
```

Mit global wird gesagt, dass wir die globalen Variablen verwenden möchten. Diese werden dann verändert.

Die geänderten globalen Variablen a und b werden ausgegeben.

```
ausgabe2()  
print("Globale Werte: a = {}, b = {}".format(a,b))
```

Die globalen Variablen a und b wurden verändert.



```
a = "Die Zahl lautet"  
b = 5
```

a und b werden als globale Variablen erzeugt. Eine Variable c wird hier nicht angelegt.

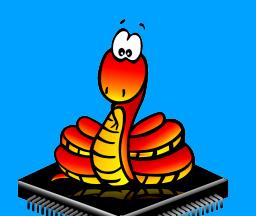
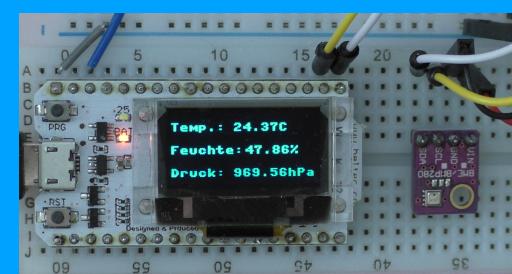
```
def ausgabe2 ():  
    print("**** Ausgabe 2 ****")  
    global a,b,c  
    a = "Eine andere Zahl:"  
    b = 7  
    c = 10  
    print(a,b)  
    print()
```

Mit global legen wir fest, dass wir die globalen Variablen a, b und c verwenden möchten. a und b werden dann verändert, c wird neu angelegt.

Die globalen Variablen a und b werden ausgegeben.

```
ausgabe2()  
print("Globale Werte: a = {}, b = {}".format(a,b))  
print(c)
```

Die globalen Variablen a und b wurden verändert. Die neue globale Variable c kann hier ebenfalls ausgegeben werden.



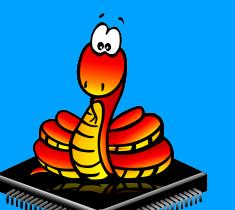
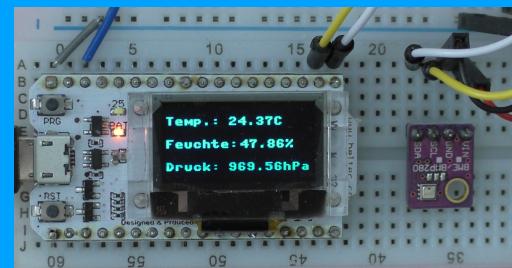
```
def ausgabe1():
    print("*** Ausgabe 1 ***")
    print(a,b,c)
    print()
```

Die globalen Variablen a, b und c können hier ausgegeben werden, obwohl sie erst weiter unten erzeugt werden.

```
a = "Die Zahl lautet"
b = 5
c = [1, 3, 5]
```

Die globalen Variablen a, b und c werden hier erzeugt. Da dieser Programmteil vor dem Aufruf von ausgabe1() durchlaufen wird, stehen die Variablen in der Funktion dann zur Verfügung.

```
ausgabe1()
print("Globale Werte: a = {}, b = {}, c = {}".format(a,b,c))
```

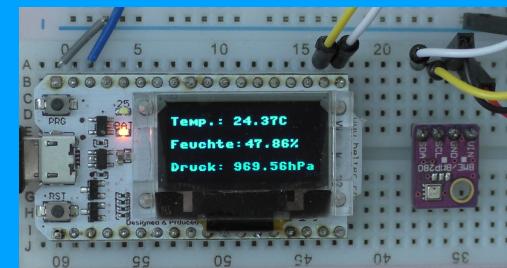


```
def ausgabe2(a,b):  
    print("*** Ausgabe 2 ***")  
    # global a,b  
    a = "Eine andere Zahl:"  
    b = 7  
    print(a,b,c)  
    print()
```

a und b werden als Argumente übergeben.

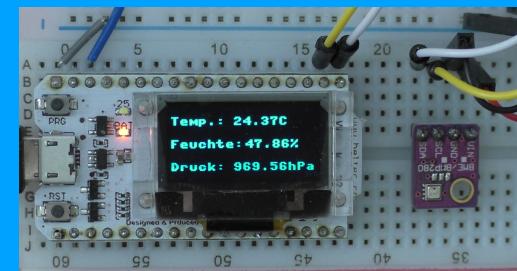
Ein Zugriff auf die globalen Variablen ist nicht mehr möglich.

Es werden neue lokale Variablen angelegt.

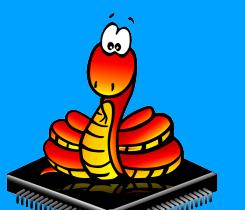


```
def ausgabe2(a,b):  
    print("*** Ausgabe 2 ***")  
    # global a,b # nicht erlaubt!  
    a = "Eine andere Zahl:"  
    b = 7  
    c[1] = 10  
    print(a,b,c)  
    print()
```

Das ändert ein Element in der globalen Liste c.



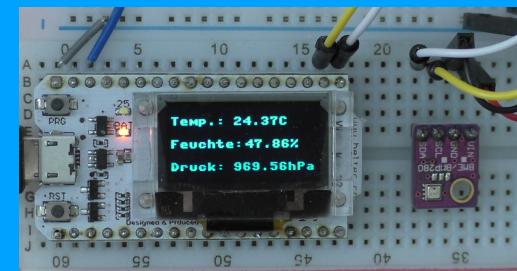
Der Hobbyelektroniker - Micropython Grundlagen
Globale und lokale Variablen



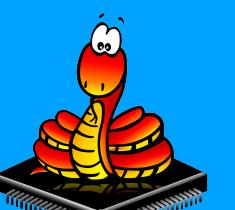
<https://micropython.org>

```
def ausgabe2(a,b):  
    print("*** Ausgabe 2 ***")  
    # global a,b # nicht erlaubt!  
    a = "Eine andere Zahl:"  
    b = 7  
    c = [2,7,4,6,5]  
    print(a,b,c)  
    print()
```

Das erzeugt eine neue lokale Liste c.



Der Hobbyelektroniker - Micropython Grundlagen
Globale und lokale Variablen



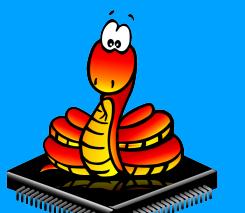
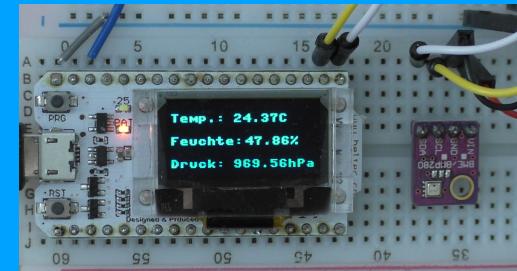
<https://micropython.org>

```
def ausgabe2(a,b):  
    print("*** Ausgabe 2 ***")  
    # global a,b  
    a = "Eine ande...  
    b = 7  
    global c  
    c[1] = 10  
    c = [2,7,4,6,5]  
    print(a,b,c)  
    print()
```

Es soll die globale Variable c verwendet werden

Das ändert ein Element in der globalen Liste c.

Die globale Liste c wird durch diese Liste ersetzt.



```
def ausgabe3(a,b,c):  
    print("*** Ausgabe 3 ***")  
    a = "Eine andere Zahl"  
    b = 7  
    c[1] = 2  
    c = [1, 7, 4, 6, 5]  
    c[2] = 12  
    print(a,b,c)  
    print()
```

c wurde als Argument übergeben und ist eine Referenz auf die globale Liste c. Darum wird diese verändert.

Eine neue lokale Liste c wird angelegt.

Hier wird die neue lokale Liste verändert.

