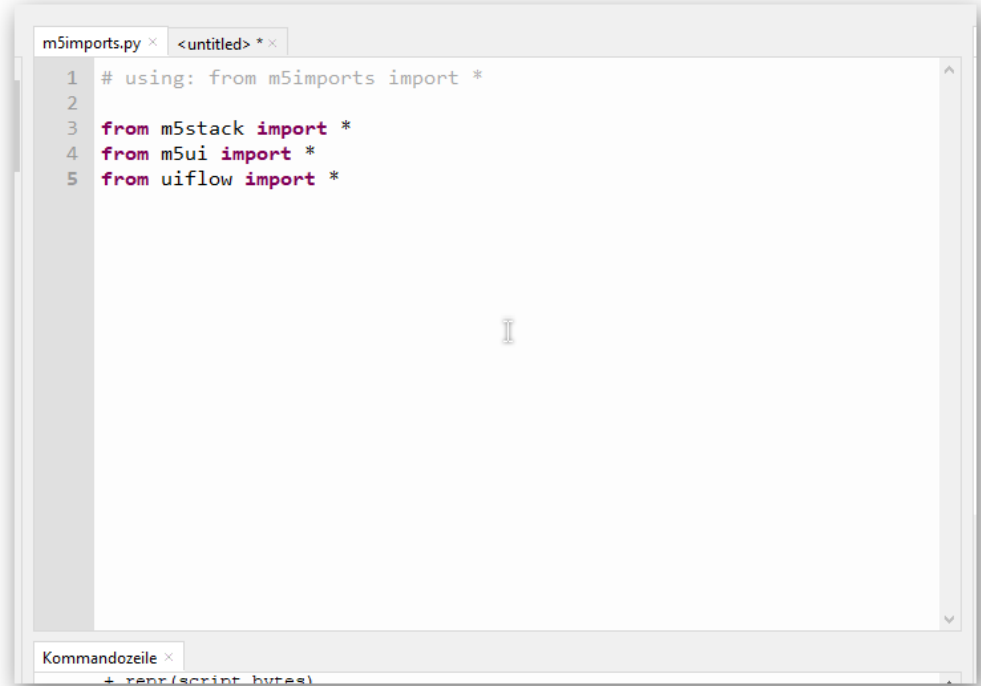


In Python wird ein Programm
Script genannt.

Ein Script ist ein einfaches
Textfile, dass keine
Formatierungen enthalten darf.

In Thonny befindet sich der
Editor im oberen Fenster. Hier
kann Text eingegeben werden.

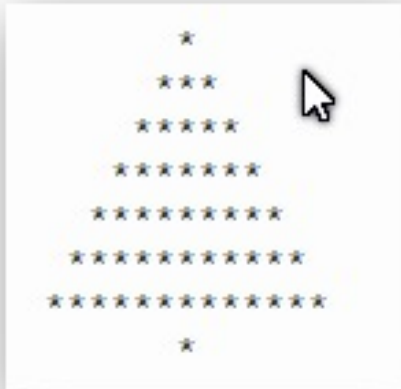
Scripte



```
m5imports.py x <untitled> * x
1 # using: from m5imports import *
2
3 from m5stack import *
4 from m5ui import *
5 from uiflow import *

Kommandozeile x
+ repr(script_bytes)
```

Wir wollen nun unser erstes Programm/Script erstellen:



Scripte

Lösung aus meinem Buch:

```
Print("      *")
Print("      ***")
Print("      *****")
Print("      *****")
Print("      *****")
Print("      *****")
Print("      *****")
Print("      *****")
Print("      *")
```

Dieses 'Programm' entspricht
überhaupt nicht meinen Vorstellungen
vom Programmieren!

Scripte

Wenn eine Reihe fasst gleicher Zeilen hintereinander steht liegt es nahe dazu eine Schleife zu verwenden.

Wie können wir in der Schleife die unterschiedlich langen Sternestrings erzeugen?

Wenn eine Reihe fasst gleicher Zeilen hintereinander steht liegt es nahe dazu eine Schleife zu verwenden.

Wie können wir in der Schleife die unterschiedlich langen Sternestrings erzeugen?

Scripte

```
>>> zweig = '*' * n  
  
>>> zweig = '*' * 1  
>>> zweig  
'*'  
  
>>> zweig = '*' * 3  
>>> zweig  
'***'
```

Scripte

Nun muss dieser String zentriert ausgegeben werden.

Dazu gibt es die String-Methode **center(Spaltenbreite)**.

Scripte

Nun muss dieser String zentriert ausgegeben werden.

Dazu gibt es die String-Methode **center**(Spaltenbreite).

```
>>> zweig.center(13)  
'      ***      '  
>>>
```

Scripte

Wie erzeugen wir die die Zahl
für die Anzahl der Sterne?

Wie erzeugen wir die die Zahl für die Anzahl der Sterne?

Mit der **range()** Funktion.

range(start, ende+1, schritt)

range() ist ein Generator und funktioniert nur mit einer Schleife, die die Werte nacheinander abrufen.

Scripte

```
>>> range(1, 13+1, 2)
range(1, 14, 2)
>>> for n in range(1, 13+1, 2):
    print(n)
```

```
1
3
5
7
9
11
13
>>>
```

Nun haben wir alles zusammen
um unser erstes Script zu
schreiben:

Scripte

```
for n in range(1,13+1,2):  
    zweig = '*' * n  
    print(zweig.center(13))  
print('*'.center(13))
```

Scripte

Nun wollen wir das Script so ändern, das wir unterschiedlich große Tannenbäume erstellen können.

Außerdem wollen wir unser Script als Funktion in ein Modul packen, dass wir dann importieren können.

Zuerst machen wir aus dem
Script ein Modul

Scripte

Funktion aus dem Script machen:

```
def tanne():  
    for n in range(1,13+1,2):  
        zweig = '*' * n  
        print(zweig.center(13))  
    print('*'.center(13))  
    return
```

Nun muss der Endwert flexibel werden.

Scripte

Funktion aus dem Script machen:

```
def tanne():  
    for n in range(1, 13+1, 2):  
        zweig = '*' * n  
        print(zweig.center(13))  
    print('*'.center(13))  
    return
```

Scripte

Dazu verwenden wir eine Variable.

Diesen übergeben wir unserer Funktion **tanne(anzahl_sterne)** beim Aufruf.

```
# Funktion aus dem Script machen:  
# Anzahl der Sterne übergeben  
  
def tanne(anzahl_sterne):  
    for n in range(1,anzahl_sterne+1,2):  
        zweig = '*' * n  
        print(zweig.center(anzahl_sterne))  
    print('*'.center(anzahl_sterne))  
    return
```

Nun können wir unser erstes eigenes Modul benutzen.

Dazu muss es zuerst in das Verzeichnis **/flash** des M5Stick geladen werden.

Scripte

```
>>> import tannen_baum as tb
>>> tanne(13)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'tanne' isn't defined
>>> tb.tanne(13)
*
***
*****
*****
*****
*****
*****
*****
*
>>>
```