

- File:
- Kursnamen, (di folder Kursnamen)
 - Bewertung Unterlagen, (di folder Stellen setzen - Kurs)
 - befragen haben di folder
 - ~~über~~ Folien + über & über (week 1, Day 3 ...)

```
for i in range(0, len(mystring))
```

↑ ↑
inclusive exclusive

Zip-Funktion

↳ combine 2 list

```
for i, j in zip(mylist, mylist2):
    ....
```

Enumerate Funktion

If-statement

If (...):

else: ~~....~~

```
{
  if ....:
  elif ....:
  else:
  ..
}
```

List comprehension

```
mylist = [x for x in range(0, 8)]
```

```
mylist2 = [x for x in range(0, 8) if x%2 == 0 and x < 5]
```

function & docstring

function

global variable

Eigenes "package" erstellen

Einzelnen Projekt = niceout.py
↳ nice()

```
import niceout
niceout.nice("...")
```

Default parameter

Default parameter

Non-keyword arguments : *args

↳ Jeder argument wird als Element einer tuples.

↓
tuple with the length 1 → *args

~~Keyword Arguments~~ :

Keyword Arguments : **kwargs

Package typen

```
def myfunction(x: int, text: str) → list:
    out = [x, text]
    return out
```

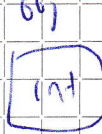
function als parameter → function is first class objects.

Reference

$x = 5$



Referenz

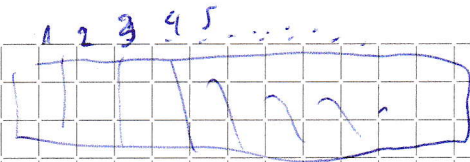


Wert 5

id

2210

(virtual)



Physikalisch
im RAM

25

pick integer more than

funktion aus Modul importieren

Nachvollziehen Referenz innerhalb der funktion

Beachten innerhalb fehlt \rightarrow id

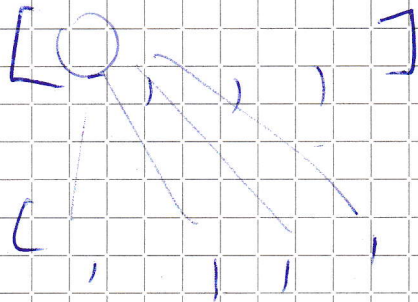
13.20

lambda funktion

map funktion \rightarrow element wise

results = map [funktion, mylist]

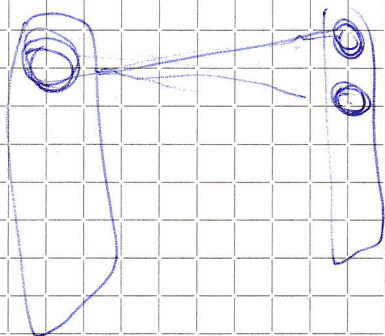
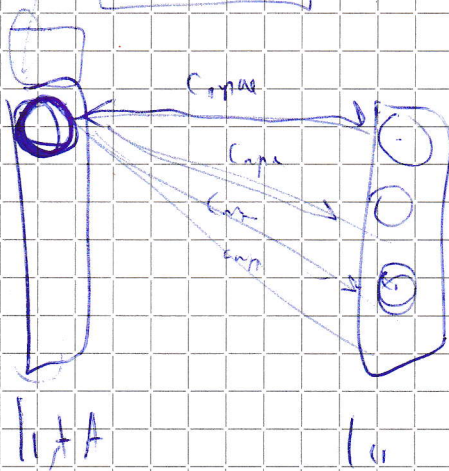
Recursion



0, 1, 2, 3, 4



Dungeon Crawler



designing:
Game for next week
- What purpose...

text based Adventure Game
text-based adventure Game

console → interface

Spoken

also