Python
...für Anfänger

Unit 13

by

Dr. Günter Kolousek

20. September 2015

Bitoperationen - 1

```
>>> 12 & 6
4
>>> bin(12)
'0b1100'
>>> bin(6)
'0b110'
>>> 0b1100 & 0b0110
4
>>> bin(4)
'0b100'
>>>
```

Bitoperationen - 2

```
>>> 12 | 6
14
>>> bin(14)
'0b1110'
>>> 12 ^ 6
10
>>> bin(10)
'0b1010'
>>> ~12
-13  # ? -> unendlich lange Zahlen!
>>> bin(12 ^ 0xF)
'0b11'
```

Bitoperationen - 3

```
>>> True & False
False
>>> True | False
True
>>> True ^ True
False
>>> ~True  # Achtung!
-2
>>> bool(~True)
True
>>> ~False
-1
```

Vertauschung ohne Hilfsvariable - 3

ohne arithmetische Operationen und ohne Tupel?

Vertauschung ohne Hilfsvariable - 3

ohne arithmetische Operationen und ohne Tupel?

```
>>> a = 1
>>> b = 2
>>> a = a ^ b
>>> b = a ^ b
>>> a = a ^ b
>>> a
2
>>> b
1
```

Sequenzielle Suche - 1

```
>>> lst = [3, 1, 5, 9, 7, 6]
>>> x = 5
>>> for e in lst:
...     if x == e:
...         print(”found”)
...
found
```

Sequenzielle Suche - 2

```
>>> lst = [3, 1, 5, 9, 7, 6]
>>> x = 5
>>> i = 0
>>> for e in lst:
...     if x == e:
...         print("found at index:", i)
...     i += 1
found at index: 2
```

Sequenzielle Suche - 3

```
>>> lst = [3, 1, 5, 9, 7, 6]
>>> x = 0
>>> i = 0
>>> for e in lst:
...     if x == e:
...         print(”found at index:”, i)
...     i += 1
...
>>>
```

Sequenzielle Suche - 4

```
lst = [3, 1, 5, 9, 7, 6]
def search(x, lst):
    i = 0
    for e in lst:
        if x == e:
            print("found at index:", i)
        i += 1
    if i == len(lst):
        print("not found")
search(5, lst)
```

```
found at index: 2
not found
```

Sequenzielle Suche - 5

```
lst = [3, 1, 5, 9, 7, 6]
def search(x, lst):
    i = 0
    for e in lst:
        if x == e:
            print("found at index:", i)
            break
        i += 1
    if i == len(lst):
        print("not found")
search(5, lst)

found at index: 2
```

Sequenzielle Suche - 6

```
lst = [3, 1, 5, 9, 7, 6]
def search(x, lst):
    i = 0
    for e in lst:
        if x == e:
            return i
        i += 1
    return
print(search(5, lst))  # -> 2
print(search(0, lst))  # -> None
```

Sequenzielle Suche - 7

```
lst = [3, 1, 5, 9, 7, 6]

pos = search(5, lst)

if pos:
    print("found at:", pos)  # -> found at: 2
else:
    print("not found")
```

Binäre Suche

```
def binary_search(x, lst):
    lb = 0
    ub = len(lst) - 1
    while lb <= ub:
        mid = (lb + ub) // 2
        if x == lst[mid]:
            return mid
        elif x < lst[mid]:
            ub = mid - 1
        else:
            lb = mid + 1

lst = [1, 3, 5, 6, 7, 9]
print(binary_search(9, lst))
```