



Lektionstillfälle 6

"Föreläsning 6 - Villkor och Loopar"

Utbildare: Robert Westin

NACKADEMIN

Kort summering av föregående lektion/ev. lektioner

Föregående lektion:

- Datastrukturer - (List, Tuple, Set, Dict)
- [], (), {}
- List och Tuple indexeras med heltal.
- Containers
- Immutable och Mutable
- Dictionary {key: value}
- Exempel i Jupyter

Lektionstillfällets mål och metod

Mål med lektionen:

- Villkor & Loopar

Lektionens arbetsmetod/er:

- Föreläsning & Labb

Begreppsgenomgång

Logiska värden: True, False

Logiska operatorer: not, and, or,

Logiska jämförande operatorer: \leq , \neq , $==$

Logiskt uttryck: $x > 10$ and $x < 20$

Villkor – if / else

Loopar

List comprehension

Logiska uttryck

Sant eller Falskt, 1 eller 0

- $(x == y)$
- $(x \text{ or } y \text{ or } z)$
- $(x \text{ and } y \text{ and } z)$

Jämförande, tillhörande och identiteter

- $(x < 20)$
- $(x \text{ in } [1, 2, 3, 4])$
- $(x \text{ is } y)$ eller $(\text{id}(x) == \text{id}(y))$

Logiska jämförande operatorer

<	strictly less than
<=	less than or equal
>	strictly greater than
>=	greater than or equal
==	equal
!=	not equal
is	object identity
is not	negated object identity

<https://docs.python.org/3/library/stdtypes.html#comparisons>

Operatorordning

Operator	Description
...	...
lambda	Lambda expression
if – else	Conditional expression
or	Boolean OR
and	Boolean AND
not x	Boolean NOT
in , not in , is , is not , <, <=, >, >=, !=, ==	Comparisons, including membership tests and identity tests

<https://docs.python.org/3/reference/expressions.html#operator-precedence>

Logisk operator not

True

- not False
- not 1 > 5
- 1 not in [2, 3,4]

False

- not True
- not "volvo" == "saab" # läs som not ("volvo" == "saab")

Logisk operator "and" och "or"

Operation	Resultat
x or y	om x är false, returnera y, annars x
x and y	om x är false, returnera x, annars y
not x	om x är false, returnera True, annars False

<https://docs.python.org/3/library/stdtypes.html#boolean-operations-and-or-not>

True and False or not False
(True and False) or (not False)
False or True

False and not False or True
(False and (not False)) or True

Att kontrollera flödet i programmet

The if statement is used for conditional execution:

```
if_stmt ::= "if" assignment expression ":" suite  
          ("elif" assignment expression ":" suite)*  
          ["else" ":" suite]
```

https://docs.python.org/3/reference/compound_stmts.html#the-if-statement

Fler exempel:

- For-sats
- If-sats
- While-sats
- Try-sats
- Break / Continue / Pass

<https://docs.python.org/3/tutorial/controlflow.html#more-control-flow-tools>

If-sats

```
x = 5
if(x > 5):
    pass
elif(x == 5):
    pass
else:
    pass
```

for-sats

```
for i in range(10):  
    print(i)
```

0
1
2
3
4
5
6
7
8
9

```
a_list = ["a", "b", "c"]  
for element in a_list:  
    print(element)
```

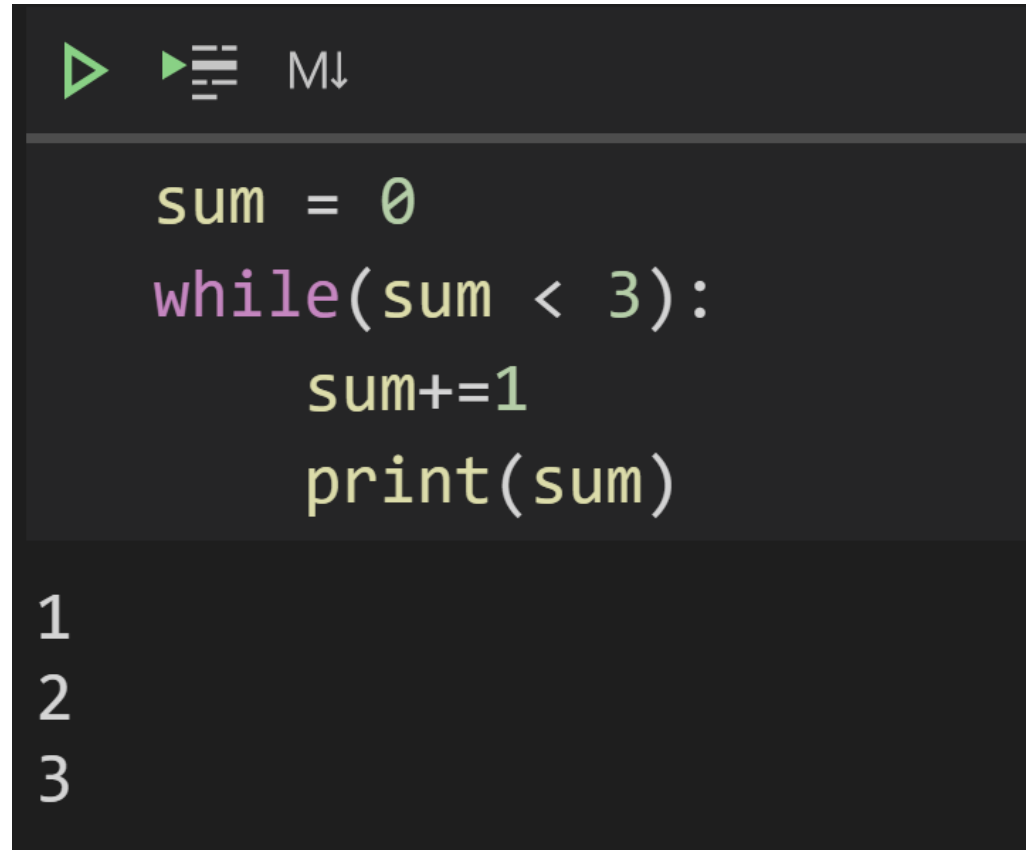
a
b
c

for-sats

```
cars = ["volvo", "saab", "tesla"]  
for i, car in enumerate(cars):  
    print(f'index: {i} car: {car}')  
    cars[i] = "sold"  
print(cars)
```

```
index: 0 car: volvo  
index: 1 car: saab  
index: 2 car: tesla  
['sold', 'sold', 'sold']
```

While-sats

A dark-themed Python code editor interface. At the top, there is a toolbar with a green play button, a button with three horizontal lines and a right-pointing arrow, and a button with 'M' and a downward arrow. Below the toolbar, the code is written in a monospaced font with syntax highlighting: 'sum = 0' in yellow, 'while(sum < 3):' in purple, 'sum+=1' in yellow, and 'print(sum)' in yellow. At the bottom left, the output of the code is shown as the numbers '1', '2', and '3' stacked vertically in white.

```
sum = 0
while(sum < 3):
    sum+=1
    print(sum)
```

1
2
3

Summering av dagens lektion

- Logiska uttryck
- Vi kan styra programflödet med hjälp av exempelvis if-satser och loopar
- Frågor? Är det något som kräver mer övning?

Framåtblick inför nästa lektion

- Återanvändning av kod
- Funktioner, moduler och paket