

# Utvärdering av uttryck steg för steg

Aritmetiska uttryck

Logiska uttryck

Funktionsanrop

# Exekvering av kontrollstrukturer steg för steg

if-else

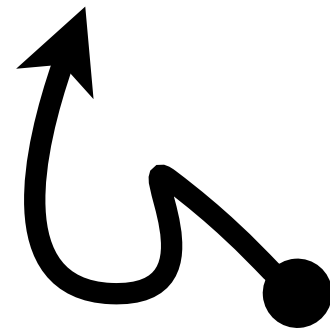
for-in

# Uttryck

Vad menas med begreppet uttryck?

Ett uttryck är i matematik en **kombination** av **siffror**, **operatorer**, **grupperande symboler** (som klammer och parentes) och/eller **variabler** ...

... **ordnade** på ett meningsfullt sätt så att uttrycket kan bli **utvärderat**.



Kallas även för **evaluerat**.

# Aritmetiska uttryck i Python

Exempel på aritmetiska i Python

```
>>> 3 + 2 * 10 / 2
```

```
13
```

```
>>> 2 * (7 / 2) + 7 % 2
```

```
7
```

```
>>> (1*2**0) + (0*2**1) + (1*2**2)
```

```
5
```

```
>>> 0b101 + 0x2b
```

```
48
```

# Logiska uttryck i Python

Exempel på logiska uttryck i Python

```
>>> 127 == 999
```

```
False
```

```
>>> 127 < 999
```

```
True
```

```
>>> 127 != 99
```

```
True
```

```
>>> not 127 < 999
```

```
False
```

# Utvärdering av uttryck

Exempel på vad som kan ingå i olika uttryck.

- Aritmetiska uttryck
- Logiska uttryck
- Variabler
- Funktionsanrop
- Ett uttryck kan i sin tur bestå av flera andra uttryck.

Uttryck utvärderas “**bottom-up**” (inre uttryck först).

**Utvärdering av**  
**aritmetiskt**  
**uttryck**  
**steg för steg**

# Utvärdering av uttryck steg för steg

Uttryck utvärderas “**bottom-up**” (inre uttryck först).

$$3 + ( 2 * (10 / 2) )$$

# Utvärdering av uttryck steg för steg

Uttryck utvärderas “**bottom-up**” (inre uttryck först).

Först utvärderas

$$10 / 2 \rightarrow 5$$

$$3 + ( 2 * \boxed{(10 / 2)} )$$

Diagram illustrating the first step of evaluation. The innermost expression  $(10 / 2)$  is highlighted with a red box and labeled with a purple circle containing the number 1.

Sedan utvärderas

$$2 * 5 \rightarrow 10$$

$$3 + ( \boxed{2 * 5} )$$

Diagram illustrating the second step of evaluation. The expression  $2 * 5$  is highlighted with a red box and labeled with a purple circle containing the number 2.

Slutligen utvärderas

$$3 + 10 \rightarrow 13$$

$$\boxed{3 + 10}$$

Diagram illustrating the third step of evaluation. The entire expression  $3 + 10$  is highlighted with a red box and labeled with a purple circle containing the number 3.

Resultatet av hela  
uttrycket

$$\boxed{13}$$



**Utvärdering av**  
**funktionsanrop**  
**steg för steg**

# Utvärdering av **funktionsanrop** steg för steg

```
def double( n ):
    return n * 2
```

```
def tripple( n ):
    return 3 * n
```

```
x = double( 11 - tripple(3) )
```

# Utvärdering av **funktionsanrop** steg för steg

```
def double( n ):
    return n * 2
```

```
def tripple( n = 3 ):
    return 3 * n
```

```
x = double( 11 - tripple(3) )
```

# Utvärdering av **funktionsanrop** steg för steg

```
def double( n ):
    return n * 2
```

```
def tripple( n = 3 ):
    return 3 * n
```

```
x = double( 11 - tripple(3) )
```

# Utvärdering av **funktionsanrop** steg för steg

```
def double( n ):  
    return n * 2
```

```
def tripple( n = 3 ):  
    return 3 * 3
```

```
x = double( 11 - tripple(3) )
```

# Utvärdering av **funktionsanrop** steg för steg

```
def double( n ):  
    return n * 2
```

```
def tripple( n = 3 ):  
    return 3 * 3
```

```
x = double( 11 - tripple(3) )
```

# Utvärdering av **funktionsanrop** steg för steg

```
def double( n ):  
    return n * 2
```

```
def tripple( n = 3 ):  
    return 9
```

```
x = double( 11 - tripple(3) )
```

# Utvärdering av **funktionsanrop** steg för steg

```
def double( n ):
    return n * 2
```

```
def tripple( n ):
    return 3 * n
```

```
x = double( 11 - 9 )
```



# Utvärdering av **funktionsanrop** steg för steg

```
def double( n ):
    return n * 2
```

```
def tripple( n ):
    return 3 * n
```

```
x = double( 11 - 9 )
```

# Utvärdering av **funktionsanrop** steg för steg

```
def double( n ):
    return n * 2
```

```
def tripple( n ):
    return 3 * n
```

```
x = double( 2 )
```

# Utvärdering av **funktionsanrop** steg för steg

```
def double( n ):
    return n * 2
```

```
def tripple( n ):
    return 3 * n
```

```
x = double( 2 )
```

# Utvärdering av **funktionsanrop** steg för steg

```
def double( n = 2 ):
    return n * 2
```

```
def tripple( n ):
    return 3 * n
```

```
x = double( 2 )
```

# Utvärdering av **funktionsanrop** steg för steg

```
def double( n = 2 ):
    return n * 2
```

```
def tripple( n ):
    return 3 * n
```

```
x = double( 2 )
```

# Utvärdering av **funktionsanrop** steg för steg

```
def double( n = 2 ):
```

```
    return 2 * 2
```

```
def tripple( n ):
```

```
    return 3 * n
```

```
x = double( 2 )
```

# Utvärdering av **funktionsanrop** steg för steg

```
def double( n = 2 ):
```

```
    return 2 * 2
```

```
def tripple( n ):
```

```
    return 3 * n
```

```
x = double( 2 )
```

# Utvärdering av **funktionsanrop** steg för steg

```
def double( n = 2 ):
```

```
    return 4
```

```
def tripple( n ):
```

```
    return 3 * n
```

```
x = double( 2 )
```



# Utvärdering av **funktionsanrop** steg för steg

```
def double( n ):
    return n * 2
```

```
def tripple( n ):
    return 3 * n
```

x = 4

# Kontrollstrukturer

För att styra exekveringen av ett program genom att göra val eller upprepning används olika typer av kontrollstrukturer.

I Python används kontrollstrukturen **if-elif-else** för att att göra val och kontrollstrukturerna **while** och **for-in** för upprepning.

**Exekvering av**

**if-else**

**steg för steg**

# Utvärdering av *if-else* steg för steg

**x = 5**

**if** **x** % 2 == 1 :

**x = 2 \* x**

**else:**

**x = 2 \* x - 1**

# Utvärdering av *if-else* steg för steg

**x = 5**

**if** **5** % 2 == 1 :

**x = 2 \* x**

**else:**

**x = 2 \* x - 1**

# Utvärdering av *if-else* steg för steg

**x = 5**

**if** **5 % 2 == 1 :**

**x = 2 \* x**

**else:**

**x = 2 \* x - 1**

# Utvärdering av *if-else* steg för steg

**x = 5**

**if** 1 **== 1 :**

**x = 2 \* x**

**else:**

**x = 2 \* x - 1**

# Utvärdering av *if-else* steg för steg

**x = 5**

**if** **1 == 1** :

**x = 2 \* x**

**else:**

**x = 2 \* x - 1**



# Utvärdering av *if-else* steg för steg

**x = 5**

**if**

**True**

**:**

**x = 2 \* x**

**else:**

**x = 2 \* x - 1**

# Utvärdering av *if-else* steg för steg

```
x = 5
```

```
if True :
```

```
    x = 2 * x
```

```
else:
```

```
    x = 2 * x - 1
```

# Utvärdering av *if-else* steg för steg

```
x = 5
```

```
if True :
```

```
    x = 2 * 5
```

```
else:
```

```
    x = 2 * x - 1
```

# Utvärdering av *if-else* steg för steg

```
x = 5
```

```
if True :
```

```
    x = 2 * 5
```

```
else:
```

```
    x = 2 * x - 1
```

# Utvärdering av *if-else* steg för steg

```
x = 5
```

```
if True :
```

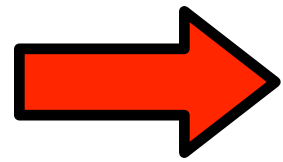
```
    x = 10
```

```
else:
```

```
    x = 2 * x - 1
```

**Exekvering av**  
**for-in**  
**steg för steg**

# Slingan **for-in** steg för steg



**xs** = [1, 9, 3]

**sum** = 0

**for** **x** **in** **xs**:

**sum** = **sum** + **x**

Minnet		
variabel	datatyp	värde

# Slingan **for-in** steg för steg

**xs** = [1, 9, 3]

 **sum** = 0

**for** **x** **in** **xs**:

**sum** = **sum** + **x**

Minnet				
variabel	datatyp	värde		
xs	list	1	9	3



# Slingan **for-in** steg för steg

**xs** = [1, 9, 3]

**sum** = 0

 **for** **x** **in** **xs**:

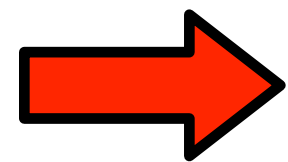
**sum** = **sum** + **x**

Minnet				
variabel	datatyp	värde		
xs	list	1	9	3
sum	int	0		

# Slingan **for-in** steg för steg

**xs** = [1, 9, 3]

**sum** = 0



**for** **x=1** **in** **xs**:

**sum** = **sum** + **x**

Minnet				
variabel	datatyp	värde		
xs	list	1	9	3
sum	int	0		
x	int	1		

# Slingan **for-in** steg för steg

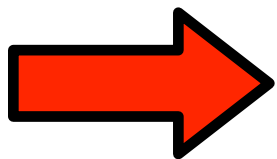
**xs** = [1, 9, 3]

**sum** = 0

**for** **x=1 in xs:**

**sum** = **sum** + **x**

Minnet				
variabel	datatyp	värde		
xs	list	1	9	3
sum	int	0		
x	int	1		



# Slingan **for-in** steg för steg

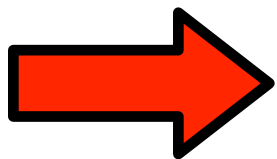
**xs** = [1, 9, 3]

**sum** = 0

**for** **x=1** **in** **xs**:

**sum** = **sum** + **x**

Minnet				
variabel	datatyp	värde		
xs	list	1	9	3
sum	int	0		
x	int	1		



# Slingan **for-in** steg för steg

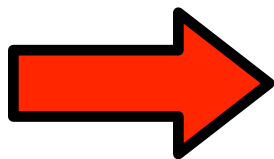
**xs** = [1, 9, 3]

**sum** = 0

**for** **x=1** **in** **xs**:

**sum** = 0 + **x**

Minnet				
variabel	datatyp	värde		
xs	list	1	9	3
sum	int	0		
x	int	1		

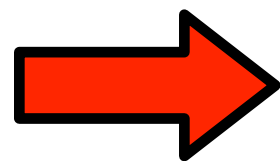


# Slingan **for-in** steg för steg

**xs** = [1, 9, 3]

**sum** = 0

**for** **x=1** **in** **xs**:



**sum** = 0 + **x**

Minnet				
variabel	datatyp	värde		
xs	list	1	9	3
sum	int	0		
x	int	1		

# Slingan **for-in** steg för steg

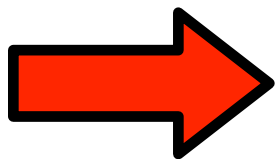
**xs** = [1, 9, 3]

**sum** = 0

**for** **x=1** **in** **xs**:

**sum** = 0 + 1

Minnet				
variabel	datatyp	värde		
xs	list	1	9	3
sum	int	0		
x	int	1		



# Slingan **for-in** steg för steg

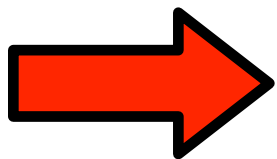
**xs** = [1, 9, 3]

**sum** = 0

**for** **x=1** **in** **xs**:

**sum** = 0 + 1

Minnet				
variabel	datatyp	värde		
xs	list	1	9	3
sum	int	0		
x	int	1		





# Slingan **for-in** steg för steg

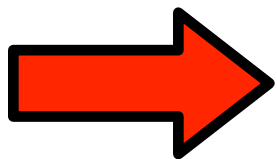
**xs** = [1, 9, 3]

**sum** = 0

**for** **x=1** **in** **xs**:

**sum** = 1

Minnet				
variabel	datatyp	värde		
xs	list	1	9	3
sum	int	0		
x	int	1		



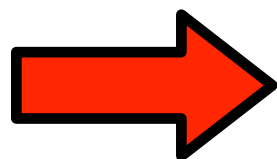
# Slingan **for-in** steg för steg

**xs** = [1, 9, 3]

**sum** = 0

**for** **x=1** **in** **xs**:

**sum** = 1

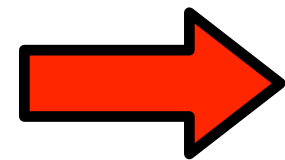


Minnet				
variabel	datatyp	värde		
xs	list	1	9	3
sum	int	1		
x	int	1		

# Slingan **for-in** steg för steg

**xs** = [1, 9, 3]

**sum** = 0



**for** **x=9** **in** **xs**:

**sum** = **sum** + **x**

Minnet				
variabel	datatyp	värde		
xs	list	1	9	3
sum	int	1		
x	int	9		

# Slingan **for-in** steg för steg

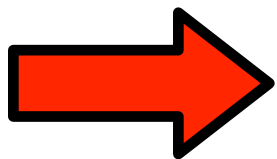
**xs** = [1, 9, 3]

**sum** = 0

**for** **x=9** **in** **xs**:

**sum** = **sum** + **x**

Minnet				
variabel	datatyp	värde		
xs	list	1	9	3
sum	int	1		
x	int	9		



# Slingan **for-in** steg för steg

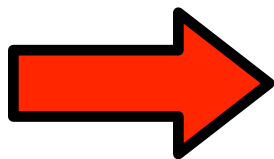
**xs** = [1, 9, 3]

**sum** = 0

**for** **x=9** **in** **xs**:

**sum** = 1 + **x**

Minnet				
variabel	datatyp	värde		
xs	list	1	9	3
sum	int	1		
x	int	9		



# Slingan **for-in** steg för steg

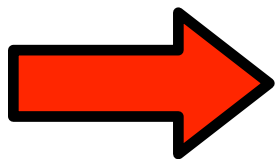
**xs** = [1, 9, 3]

**sum** = 0

**for** **x=9** **in** **xs**:

**sum** = 1 + **x**

Minnet				
variabel	datatyp	värde		
xs	list	1	9	3
sum	int	1		
x	int	9		



# Slingan **for-in** steg för steg

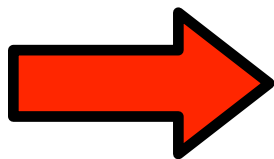
**xs** = [1, 9, 3]

**sum** = 0

**for** **x=9** **in** **xs**:

**sum** = 1 + 9

Minnet				
variabel	datatyp	värde		
xs	list	1	9	3
sum	int	1		
x	int	9		



# Slingan **for-in** steg för steg

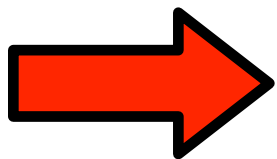
**xs** = [1, 9, 3]

**sum** = 0

**for** **x=9** **in** **xs**:

**sum** = 1 + 9

Minnet				
variabel	datatyp	värde		
xs	list	1	9	3
sum	int	1		
x	int	9		





# Slingan **for-in** steg för steg

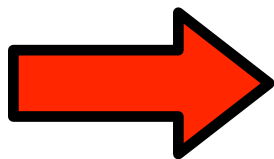
**xs** = [1, 9, 3]

**sum** = 0

**for** **x=9** **in** **xs**:

**sum** = 10

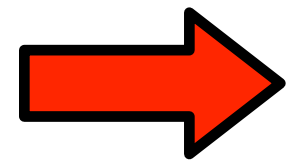
Minnet				
variabel	datatyp	värde		
xs	list	1	9	3
sum	int	1		
x	int	9		



# Slingan **for-in** steg för steg

**xs** = [1, 9, 3]

**sum** = 0



**for** **x=3** **in** **xs**:

**sum** = **sum** + **x**

Minnet				
variabel	datatyp	värde		
xs	list	1	9	3
sum	int	10		
x	int	3		

# Slingan **for-in** steg för steg

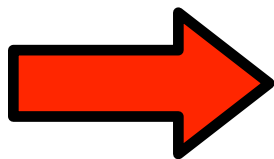
**xs** = [1, 9, 3]

**sum** = 0

**for** **x=3** **in** **xs**:

**sum** = **sum** + **x**

Minnet				
variabel	datatyp	värde		
xs	list	1	9	3
sum	int	10		
x	int	3		



# Slingan **for-in** steg för steg

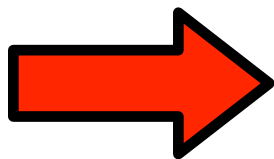
**xs** = [1, 9, 3]

**sum** = 0

**for** **x=3** **in** **xs**:

**sum** = 10 + **x**

Minnet				
variabel	datatyp	värde		
xs	list	1	9	3
sum	int	10		
x	int	3		



# Slingan **for-in** steg för steg

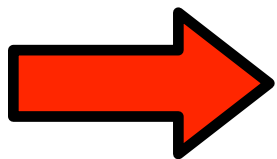
**xs** = [1, 9, 3]

**sum** = 0

**for** **x=3** **in** **xs**:

**sum** = 10 + **x**

Minnet				
variabel	datatyp	värde		
xs	list	1	9	3
sum	int	10		
x	int	3		



# Slingan **for-in** steg för steg

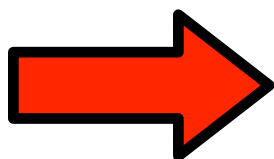
**xs** = [1, 9, 3]

**sum** = 0

**for** **x=3** **in** **xs**:

**sum** = 10 + 3

Minnet				
variabel	datatyp	värde		
xs	list	1	9	3
sum	int	10		
x	int	3		



# Slingan **for-in** steg för steg

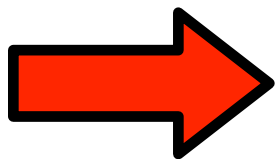
**xs** = [1, 9, 3]

**sum** = 0

**for** **x=3** **in** **xs**:

**sum** = 10 + 3

Minnet				
variabel	datatyp	värde		
xs	list	1	9	3
sum	int	10		
x	int	3		



# Slingan **for-in** steg för steg

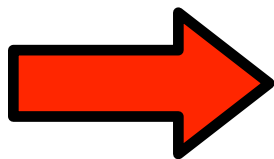
**xs** = [1, 9, 3]

**sum** = 0

**for** **x=3** **in** **xs**:

**sum** = 13

Minnet				
variabel	datatyp	värde		
xs	list	1	9	3
sum	int	10		
x	int	3		





# Slingan **for-in** steg för steg

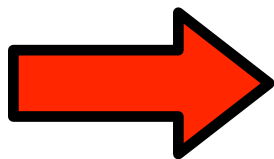
**xs** = [1, 9, 3]

**sum** = 0

**for** **x=3** **in** **xs**:

**sum** = 13

Minnet				
variabel	datatyp	värde		
xs	list	1	9	3
sum	int	13		
x	int	3		



# Slingan **for-in** steg för steg

**xs** = [1, 9, 3]

**sum** = 0

**for** **x=3** **in** **xs**:

**sum** = 13

Minnet				
variabel	datatyp	värde		
xs	list	1	9	3
sum	int	13		
x	int	3		

