

FLOW CONTROL STATEMENTS

Omogočajo nam kontrolo sprememb in logike programa.

If statement

```
if <expr>:  
    <statement>  
    <statement>  
    ...  
    <statement>  
<following_statement>
```

<expr> je izraz ovrednoten v Boolean kontekstu.

<statement> je Python izraz (nadaljevanje naše kode), ki je pravilno zamaknjen.

Če je <expr> **True**, potem se izvedejo <statement>. Če je <expr> **False**, potem se <statement> preskoči in se ne izvede.

Nato se program nadaljuje z <following_statement>

Indentation / Zamikanje

Pri Pythonu se zamikanje (indentation) uporablja za definiranje blokov kode. Vse vrstice z istim zamikom se smatrajo kot isti blok kode.

Bloke kode se lahko poljubno globoko "nesta".

Zamikanje je določeno z tabulatorjem ali presledki. Ni važno točno število, važno je, da je skozi kodo enako.

In [5]:

```
x = 0  
y = 5  
  
if x < y:  
    print("Smo znotraj if.")  
    print("End if")  
print("End")
```

Smo znotraj if.
End if
End

Else

Včasih želimo, da če je nekaj res se izvede določen blok kode, če stvar ni res pa naj se izvede drug del kode.

To dosežemo z `else`.

```
if <expr>:
    <statement(s)>
else:
    <statement(s)>
<following_statement>
```

Če je `<expr>` `True` se izvede blok direktno pod njem, če pa je `<expr>` `False` se ta blok kode preskoči in se izvede blok pod `else`.

In [4]:

```
x = 100

if x < 50:
    print('(first block)')
    print('x is small')
else:
    print('(second block)')
    print('x is large')

print("End")
```

```
(second block)
x is large
End
```

Elif

Če želimo še večjo razvejanost naših možnosti lahko uporabimo `elif` (`else if`).

```
if <expr>:
    <statement(s)>
elif <expr>:
    <statement(s)>
elif <expr>:
    <statement(s)>
else:
    <statement(s)>
<following_statement>
```

Python preveri vsak `<expr>` posebej. Pri ta prvem, ki bo `True`, bo izvedel njegov blok kode.

Če ni nobeden `True` se bo izvedel `else` blok kode.

In [5]:

```

x = 20
if x > 100:
    print('x je večje od 100')
elif x > 50:
    print('x večje od 50 in manjše od 100')
elif x > 30:
    print('x večje od 30 in manjše od 50')
elif x > 10:
    print('x večje od 10 in manjše od 30')
else:
    print("x manjše od 10")

print("End")

```

```

x večje od 10 in manjše od 30
End

```

One-line if statement

Obstaja način zapisa if stavka v eni vrstici ampak se ta način odsvetuje, ker napravi kodo nepregledno.

<https://realpython.com/python-conditional-statements/> (<https://realpython.com/python-conditional-statements/>)

```
<expr1> if <conditional_expr> else <expr2>
```

```
z = 1 + x if x > y else y + 2
```

If <conditional_expr> is true, <expr1> is returned and <expr2> is not evaluated.

If <conditional_expr> is false, <expr2> is returned and <expr1> is not evaluated.

In [6]:

```

x = 8
z = 1 + x if x > 10 else x**2
print(z)

x = 20
z = 1 + x if x > 10 else x**2
print(z)

```

```

64
21

```

The pass statements

Uporablja se kot "placeholder", da nam interpreter ne meče napak.

In [11]:

```
if True:
    print("Hello") # should give IndentationError

File "<ipython-input-11-33a91c099307>", line 3
    print("Hello") # should give IndentationError
    ^
IndentationError: expected an indented block
```

In [12]:

```
if True:
    pass
print("Hello") # should be fine now with the pass added
```

Hello

Vaja 01

Napišite program, ki bo uporabnika uprašal naj vnese neko celoštevilsko vrednost. Program naj nato izpiše ali je vrednost deljiva z 3 ali ne.

In [1]:

```
x = int(input("Vnesi celoštevilsko vrednost: "))
if x%3 == 0:
    print("Število je deljivo s 3")
else:
    print("Število ni deljivo s 3")
```

Vnesi celoštevilsko vrednost: 10
Število ni deljivo s 3

In []: