

Errors

Error's so napake v programu, ki nam ponavadi zaustavijo izvajanje programa.

Klasificiramo jih v:

- Syntax errors
- Runtime errors
- Logical errors

Syntax errors

Syntax errors so napake pri uporabi Python jezika.

Python bo našel te napake med parsanjem našega programa. Če najde takšno napako bo exit-u brez, da bi pognal ta del kode.

Najbolj pogoste Syntax napake so:

- izpustitev keyword
- uporaba keyword na napačnem mestu
- izpustitev simbolov, kot je :
- napačno črkovanje
- napačen indentation

In [1]:

```
# Primer: manjka keyword def
myfunction(x, y):
    return x + y
```

```
File "<ipython-input-1-8b32d31d1203>", line 2
    myfunction(x, y):
                    ^
```

SyntaxError: invalid syntax

In [2]:

```
else:
    print("Hello!")
```

```
File "<ipython-input-2-429811f9164b>", line 1
    else:
    ^
```

SyntaxError: invalid syntax

In [3]:

```
# Primer: manjka :  
if mark >= 50  
    print("You passed!")
```

```
File "<ipython-input-3-2bfd10af2cba>", line 2  
    if mark >= 50  
        ^
```

SyntaxError: invalid syntax

In [4]:

```
# Primer: napačno črkovanje "else"  
if arriving:  
    print("Hi!")  
esle:  
    print("Bye!")
```

```
File "<ipython-input-4-1cca186d8b5e>", line 4  
    esle:  
        ^
```

SyntaxError: invalid syntax

In [5]:

```
# Primer: napačen indentation  
if flag:  
print("Flag is set!")
```

```
File "<ipython-input-5-2009e1311970>", line 3  
    print("Flag is set!")  
    ^
```

IndentationError: expected an indented block

Runtime errors

Primer runtime errors:

- Deljenje z 0
- Dostopanje do elementov, ki ne obstajajo
- Dostopanje do datotek, ki ne obstajajo

- division by zero

- performing an operation on incompatible types
- using an identifier which has not been defined
- accessing a list element, dictionary value or object attribute which doesn't exist
- trying to access a file which doesn't exist

In [6]:

```
# Primer: deljenje z 0  
1 / 0
```

```
-----  
-----  
ZeroDivisionError                                Traceback (most recent call  
last)  
<ipython-input-6-ecalcclfcdb> in <module>  
      1 # Primer: deljenje z 0  
----> 2 1 / 0
```

ZeroDivisionError: division by zero

Logical errors

Logične napake nam povzročijo napačne rezultate. Program je lahko sintaksično pravilno zapisan ampak nam ne bo vrnil iskanega rezultata.

Primeri

- Uporabna napačne spremenljivke
- napačna indentacija
- uporaba celoštevilskega deljenja in ne navadnega deljenja

Logical errors are the most difficult to fix. They occur when the program runs without crashing, but produces an incorrect result. The error is caused by a mistake in the program's logic. You won't get an error message, because no syntax or runtime error has occurred. You will have to find the problem on your own by reviewing all the relevant parts of your code – although some tools can flag suspicious code which looks like it could cause unexpected behaviour.

In []:

The try and except statements

Da obvladujemo morebitne napake uporabljamo try-except:

In [7]:

```

for _ in range(3):
    x = int(input("Vnesi prvo številko: "))
    y = int(input("Vnesi drugo številko: "))
    rezultat = x / y
    print(f"{x}/{y} = {rezultat}")
    print()

```

Vnesi prvo številko: 1
 Vnesi drugo številko: 2
 1/2 = 0.5

Vnesi prvo številko: a

```

-----
-----
ValueError                                Traceback (most recent call
last)
<ipython-input-7-b27c485d0cd1> in <module>
      1 for _ in range(3):
----> 2     x = int(input("Vnesi prvo številko: "))
      3     y = int(input("Vnesi drugo številko: "))
      4     rezultat = x / y
      5     print(f"{x}/{y} = {rezultat}")

```

ValueError: invalid literal for int() with base 10: 'a'

Ko se zgodi napaka, Python preveri ali se naša koda nahaja znotraj **try** bloka. Če se ne nahaja, potem bomo dobili error in izvajanje programa se bo ustavilo.

Če se nahaja znotraj try-except blocka, se bo izvedla koda znotraj **except** bloka in program bo nadaljeval z izvajanjem.

In [8]:

```

for _ in range(3):
    try:
        x = int(input("Vnesi prvo številko: "))
        y = int(input("Vnesi drugo številko: "))
        rezultat = x / y
        print(f"{x}/{y} = {rezultat}")
    except:
        print("Prislo je do napake!")
    print()

```

Vnesi prvo številko: 1
 Vnesi drugo številko: 0
 Prislo je do napake!

Vnesi prvo številko: 1
 Vnesi drugo številko: a
 Prislo je do napake!

Vnesi prvo številko: 1
 Vnesi drugo številko: 2
 1/2 = 0.5

In []:

Če se je napaka zgodila znotraj funkcije in znotraj funkcije ni bila ujeta (ni bila znotraj try-except bloka), potem gre Python preverjati ali se klic te funkcije nahaja znotraj try-except bloka.

In [9]:

```
def delilnik():
    try:
        x = int(input("Vnesi prvo številko: "))
        y = int(input("Vnesi drugo številko: "))
        rezultat = x / y
        print(f"{x}/{y} = {rezultat}")
    except:
        print("Prislo je do napake!")

for _ in range(3):
    delilnik()
    print()
```

Vnesi prvo številko: a
Prislo je do napake!

Vnesi prvo številko: a
Prislo je do napake!

Vnesi prvo številko: 1
Vnesi drugo številko: 2
1/2 = 0.5

In [10]:

```
def delilnik():
    x = int(input("Vnesi prvo številko: "))
    y = int(input("Vnesi drugo številko: "))
    rezultat = x / y
    print(f"{x}/{y} = {rezultat}")

for _ in range(3):
    try:
        delilnik()
    except:
        print("Prislo je do napake!")
    print()
```

Vnesi prvo številko: 1
Vnesi drugo številko:
Prislo je do napake!

Vnesi prvo številko: s
Prislo je do napake!

Vnesi prvo številko: 1
Vnesi drugo številko: 0
Prislo je do napake!

In []:

Naloga:

Napišite funkcijo **fakulteta**, ki uporabnika vpraša naj vnese cifro in izračuna fakulteto te cifre. Fakulteta se izračuna: $3! = 3 \cdot 2 \cdot 1 = 6$

Funkcija naj vrne rezultat. Oziroma, če uporabnik ni vnesel številke naj funkcija ponovno zahteva od uporabnika vnos cifre.

INPUT:

```
print(fakulteta())
```

OUTPUT:

Vnesi cifro: a

To ni bila številka.

Vnesi cifro: b

To ni bila številka.

Vnesi cifro: 3

6

In [11]:

```
def fakulteta():  
    while True:  
        try:  
            num = int(input("Vnesi cifro: "))  
            rezultat = 1  
            for i in range(1, num+1):  
                #print(i)  
                rezultat *= i  
            return rezultat  
        except:  
            print("To ni bila številka.")  
  
print(fakulteta())
```

Vnesi cifro: a

To ni bila številka.

Vnesi cifro:

To ni bila številka.

Vnesi cifro: b

To ni bila številka.

Vnesi cifro: 3

6

In []:

