



POLYTECHNIC
OF PORTO
SUPERIOR SCHOOL
OF
MEDIA ARTS
AND DESIGN

P.PORTO

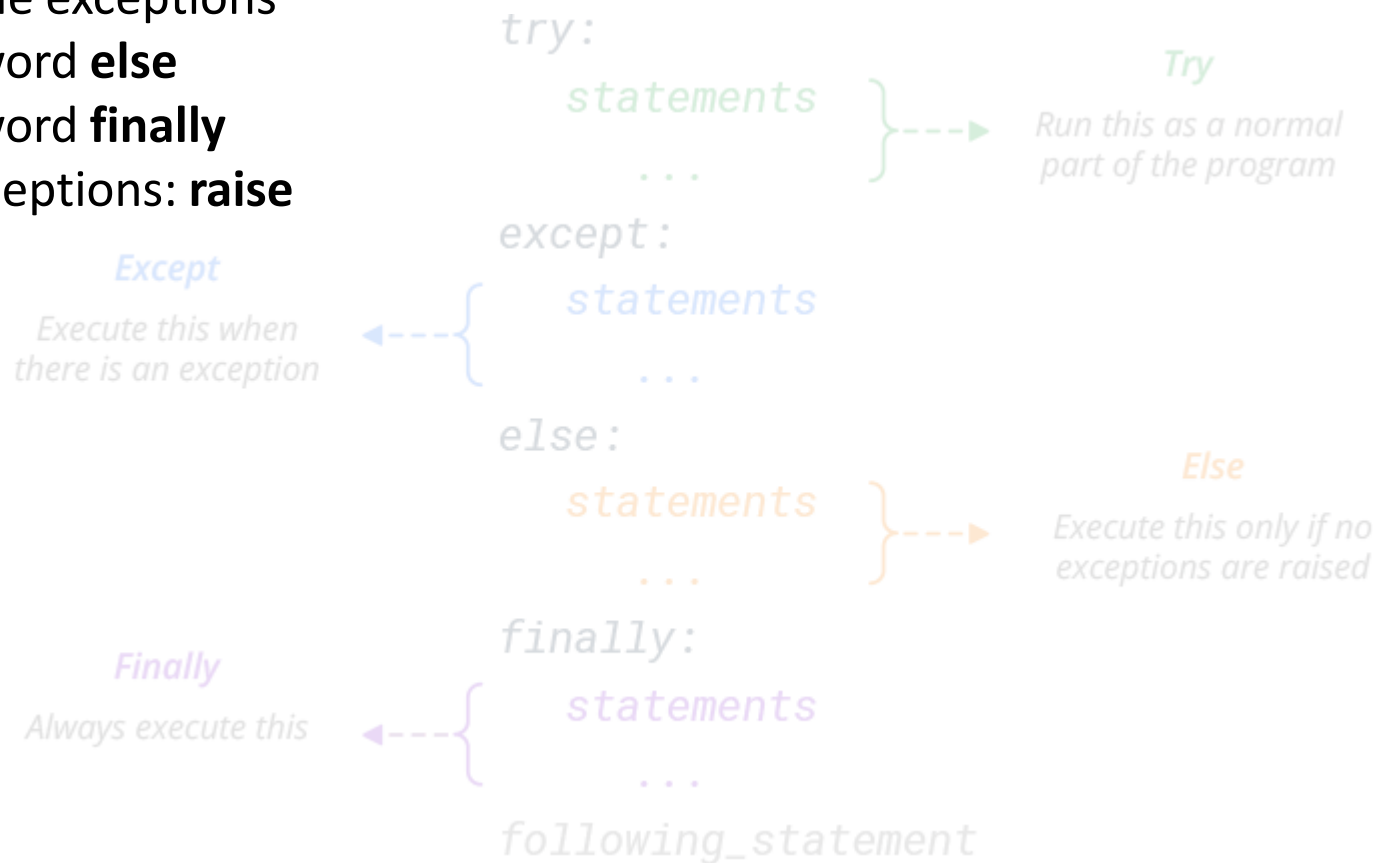
ALGORITHMS AND DATA STRUCTURES

EXCEPTIONS

TECHNOLOGIES AND INFORMATION SYSTEMS FOR THE WEB

1.Exceptions

- ❑ Exceptions
- ❑ try–exception
- ❑ Multiple exceptions
- ❑ A keyword **else**
- ❑ A keyword **finally**
- ❑ Set exceptions: **raise**



❖ Exceptions

- ❑ When executing the code, different errors may occur:
 - ❑ Coding errors made by the programmer (syntax or logic)

```
1 while True
2     print('Hello world')
```

- ❑ Errors due to incorrect data entry

```
temp1.py > number
1 number = int(input('Número:'))
```

Exception has occurred: ValueError ✕
invalid literal for int() with base 10: 'r'
File "C:\Users\mario\OneDrive\AED\4 - Exercicios\Ficha 05\temp1.py", line 1, in <module>

c:\Users\mario\OneDrive\AED\4 - Exercicios\Ficha 05 - VS Code Console

Número:r

- ❑ Unpredictable errors due to external factors
 - ❑ For exemple, a removed file, a non-existing folder, ...

❖ Exceptions

- ❑ Correct expressions or statements may cause an error
 - ❑ e.g. a division by a variable without content or with a value of 0
- ❑ When an error or exception occurs, python normally stops the execution and generates an error message.
- ❑ These exceptions can be handled using the structures **try-except**
- ❑ The objective is to capture execution errors, treat them, and not allow the program to crash.

❖ try-except

- ❑ Python executes the block included in **try** clause as a normal part of the program code
- ❑ When an error occurs during this execution, the instructions included in **try** statement are cancelled, and the code included in the **except** clause is executed



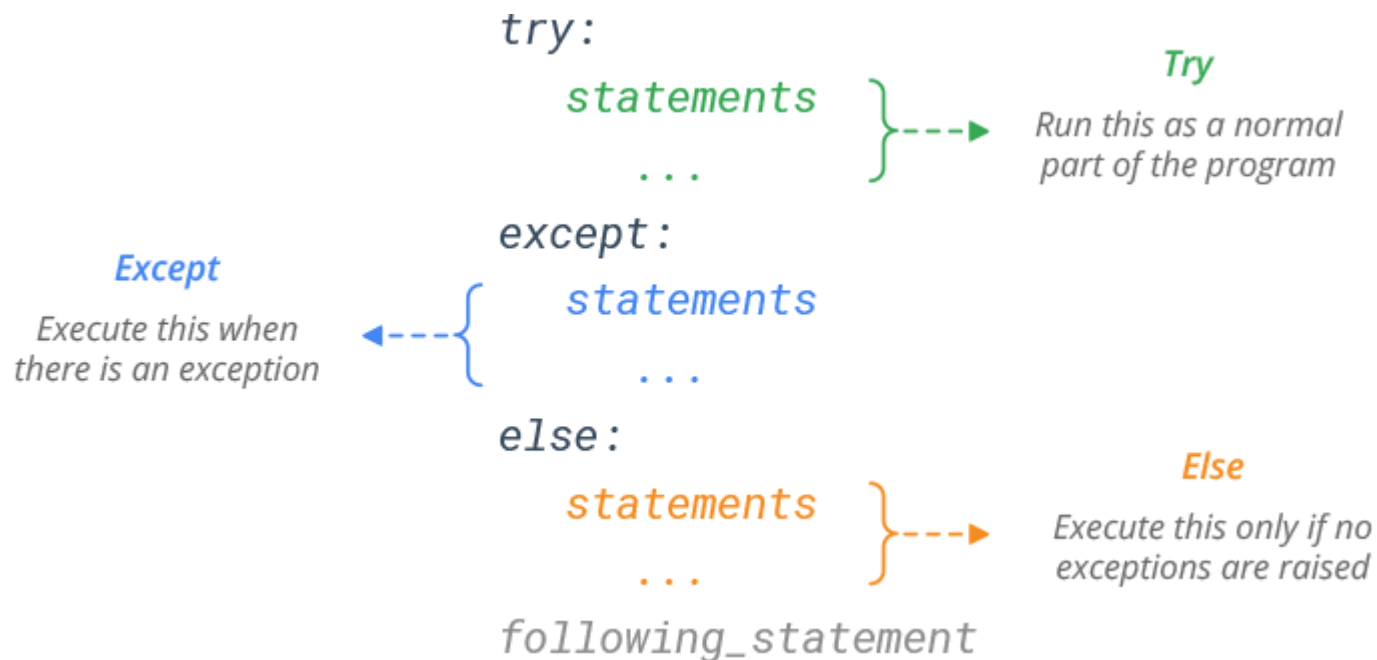
❖ try-except-except...

- ❑ We can define several exception blocks, to capture and deal with specific errors / exceptions, giving more objective information to the user

```
1
2  numero=10
3  try:
4      divisor = int(input('Número:'))
5      divisao= numero/divisor
6  except ZeroDivisionError:
7      print("Divisão por zero não é possível")
8  except:
9      print("Não é possível efetuar a divisão")
10
```

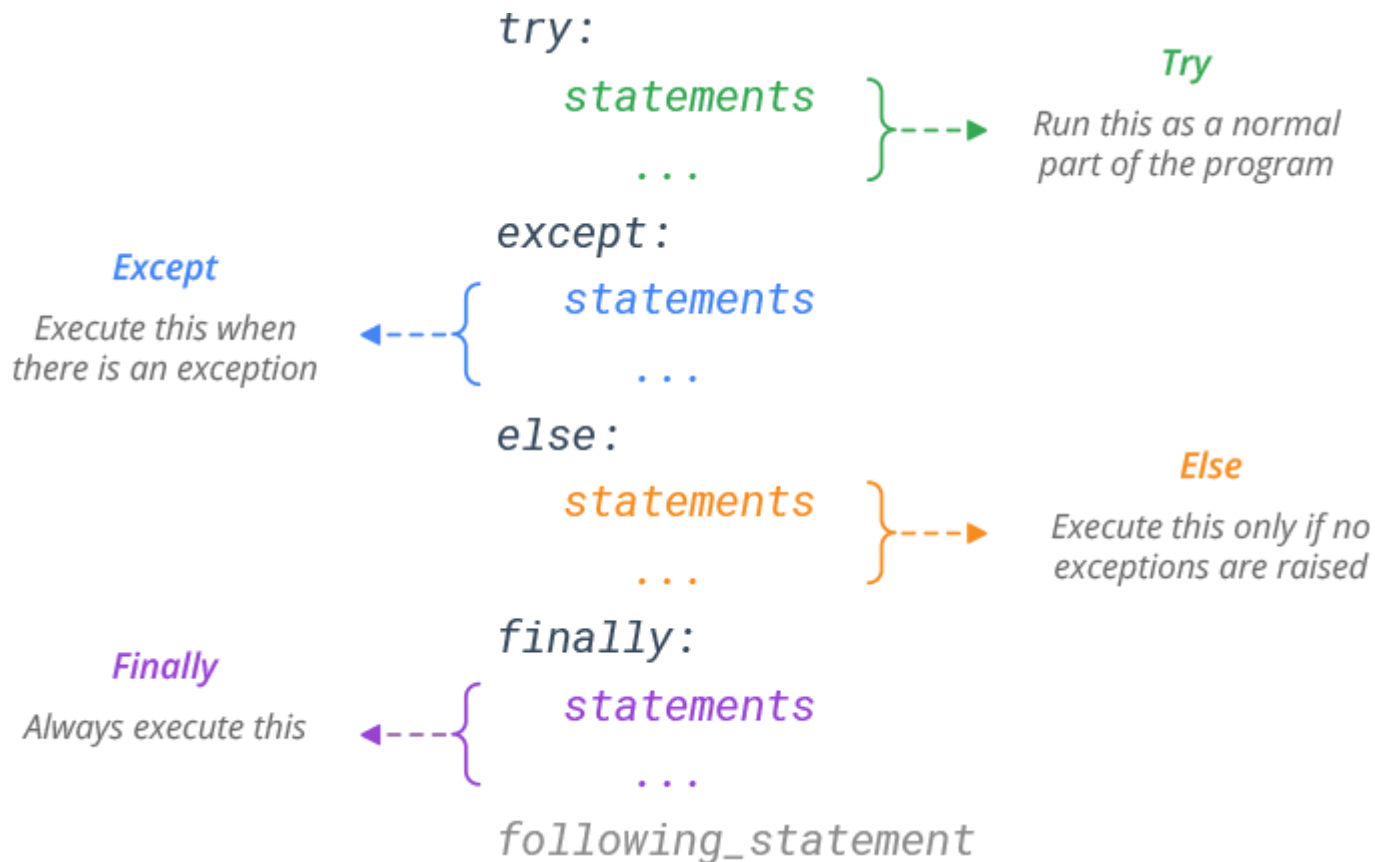
❖ try-except-else

- ❑ The **else** clause is optional
- ❑ The **else** clause is executed only when no exception (error) occurs



❖ try-except-else-finally

- ❑ The **finally** clause is also optional
- ❑ The **finally** clause is always executed, whether an exception occurred or not



❖ try-except-else-finally

- ❑ The **finally** clause is also optional
- ❑ The **finally** clause is always executed, whether an exception occurred or not
- ❑ We use the **finally** clause to define actions that must be performed in any situation, such as closing a file

```
# finally clause is always executed
try:
    x = 1/0
except:
    print('Something went wrong')
finally:
    print('Always execute this')
# Prints Something went wrong
# Prints Always execute this
```

```
# Exception handling during file manipulation
f = open('myfile.txt')
try:
    print(f.read())
except:
    print("Something went wrong")
finally:
    f.close()
```

❖ try-except-else-finally

- ❑ List of standard exceptions in Python:

https://www.tutorialspoint.com/python/standard_exceptions.htm

<https://docs.python.org/3/library/exceptions.html#concrete-exceptions>

Exceptions	Description
ValueError	Data with invalid content for the type of data
ZeroDivisionError	Attempt to divide by zero
TypeError	Invalid operation for data type
IOError	Attempt to open a file that does not exist
IndexError	When a specified index is not valid
ImportError	When it is not possible to import a certain module
ArithmeticError	When the specified arithmetic operation is not possible to perform

❖ Examples

Exceptions.py > ...

```
1
2
3 try:
4     numero = int(input("Número: "))
5 except ValueError:
6     print("O valor é incorreto")
7 except:
8     print("Ocorreu um erro na inserção de dados")
9
10 print("código a seguir ao tratamento de erro!")
11
12
```

C:\WINDOWS\py.exe

Número: 123
código a seguir ao tratamento de erro!

❖ Examples

```
try:
    numero = int(input("Número: "))
except ValueError:
    print("O valor é incorreto")
except:
    print("Ocorreu um erro na inserção")

print("código a seguir ao tratamento d
```


C:\WINDOWS\py.exe

```
Número: a123
O valor é incorreto
código a seguir ao tratamento de erro!
```

❖ Examples

```
try:
    numero = int(input("Número: "))
    divisor = int(input("Divisor: "))
    quociente = numero / divisor
    print(quociente)
except ValueError:
    print("O número inserido está incorreto")
except ZeroDivisionError:
    print("Não é possível dividir por 0!")
except:
    print("Erro no cálculo!")

print("código a seguir ao try-except")
```

 C:\WINDOWS\py.exe

Número: 10

Divisor: 2

5.0

código a seguir ao try-except

❖ Examples

Exveptions1.py > ...

```
1  try:
2      numero = int(input("Número: "))
3      divisor = int(input("Divisor: "))
4      quociente = numero / divisor
5      print(quociente)
6  except ValueError:
7      print("O número inserido está incorreto")
8  except ZeroDivisionError:
9      print("Não é possível dividir por 0!")
10 except:
11     print("Erro no cálculo!")
12
13 print("código a seguir ao try-except")
```



C:\WINDOWS\py.exe

Número: 120
Divisor: 0
Não é possível dividir por 0!
código a seguir ao try-except

❖ Examples

```
Exceptions.py / ...
1  try:
2      numero = int(input("Número: "))
3      divisor = int(input("Divisor: "))
4      quociente = numero / divisor
5      print(quociente)
6  except ValueError:
7      print("O número inserido está incorreto")
8  except ZeroDivisionError:
9      print("Não é possível dividir por 0!")
0  except:
1      print("Erro no cálculo!")
2  finally:
3      print("código executado!")
4
5  print("código a seguir ao try-except")
6  input()
```

C:\WINDOWS\py.exe

Número: 10
Divisor: 3
3.3333333333333335
código executado!
código a seguir ao try-except

❖ Examples

```
try:
    f = open("teste.txt")
    f.write("teste de escrita em ficheiro")
except:
    print("Erro na abertura do ficheiro!")
finally:
    if f.closed == False:
        f.close()
```


❖ Create Exceptions

- ❑ It is possible to define our own exceptions, defining conditions for the occurrence of these exceptions
- ❑ To throw (define) an exception use the keyword **raise**

```
Exp3.py > ...
1
2 try:
3     numero = int(input("indique um Número positivo:"))
4     if numero < 0:
5         raise ValueError()
6 except:
7     print("valor incorreto")
8
9
10 input()
11
```

C:\WINDOWS\py.exe

indique um Número positivo:-1
valor incorreto

❖ Create Exceptions

- ❑ It is possible to define our own exceptions, defining conditions for the occurrence of these exceptions
- ❑ To define an exception we use the keyword **raise**

```
try:  
    numero = int(input("indique um Número entre [0 e 20]: "))  
    if numero < 0 or numero > 20:  
        raise ValueError()  
except:  
    print("valor não está dentro dos limites definidos")
```

input()

C:\WINDOWS\py.exe

```
indique um Número entre [0 e 20]: 24  
valor não está dentro dos limites definidos  
-
```

❖ Create Exceptions

- ❑ It is possible to define our own exceptions, defining conditions for the occurrence of these exceptions
- ❑ To throw (define) an exception we use the keyword **raise**

```
try:
    numero = int(input("\n indique um Número entre [0 e 20]: "))
    if numero < 0 or numero > 20:
        raise ValueError()
except ValueError:
    print("O número inserido está incorreto!")
except:
    print("Ocorreu um erro!")
```

input()

C:\WINDOWS\py.exe

indique um Número entre [0 e 20]: a12
O número inserido está incorreto!

❖ Create Exceptions

- ❑ It is possible to define our own exceptions, defining conditions for the occurrence of these exceptions
- ❑ To throw (define) an exception we use the keyword **raise**

```
valido=False
while not valido:
    try:
        numero = int(input("\n indique um Número entre [0 e 20]: "))
        if numero < 0 or numero > 20:
            raise ValueError()
    except ValueError:
        print("O número inserido está incorreto!")
    except:
        print("Ocorreu um erro!")
    else:
        valido = True

input()
```

C:\WINDOWS\py.exe

indique um Número entre [0 e 20]: a12
O número inserido está incorreto!

indique um Número entre [0 e 20]: 21
O número inserido está incorreto!

indique um Número entre [0 e 20]: 15

Tip !

else clause is executed only when no exception (error) occurs