

PPT11 – Fundamentos de Python (UFCD 10793)

Sandra Liliana Meira de Oliveira



Tratamento de Erros

ERROS – Introdução

- Quando ocorre um erro, ou exceção, o Python pára e gera uma mensagem de erro.
- Essas exceções podem ser tratadas usando o *try* .



[Documentação Oficial](#)

ERROS – Tipos de blocos de código

try

permite testar um bloco de código quanto a erros.

except

permite tratar o erro.

else

permite executar código quando não há erro.

finally

permite que você execute código, independentemente do resultado dos blocos try e except



ERROS - Exemplos

```
1  try:
2      |   print(x)
3
4  except:
5      |   print("ERRO - Variavel não reconhecida")
6
```

```
try:
    |   print(x)
except NameError:
    |   print("Variable x não declarada")
except:
    |   print("Alguma coisa correu mal!!")
```

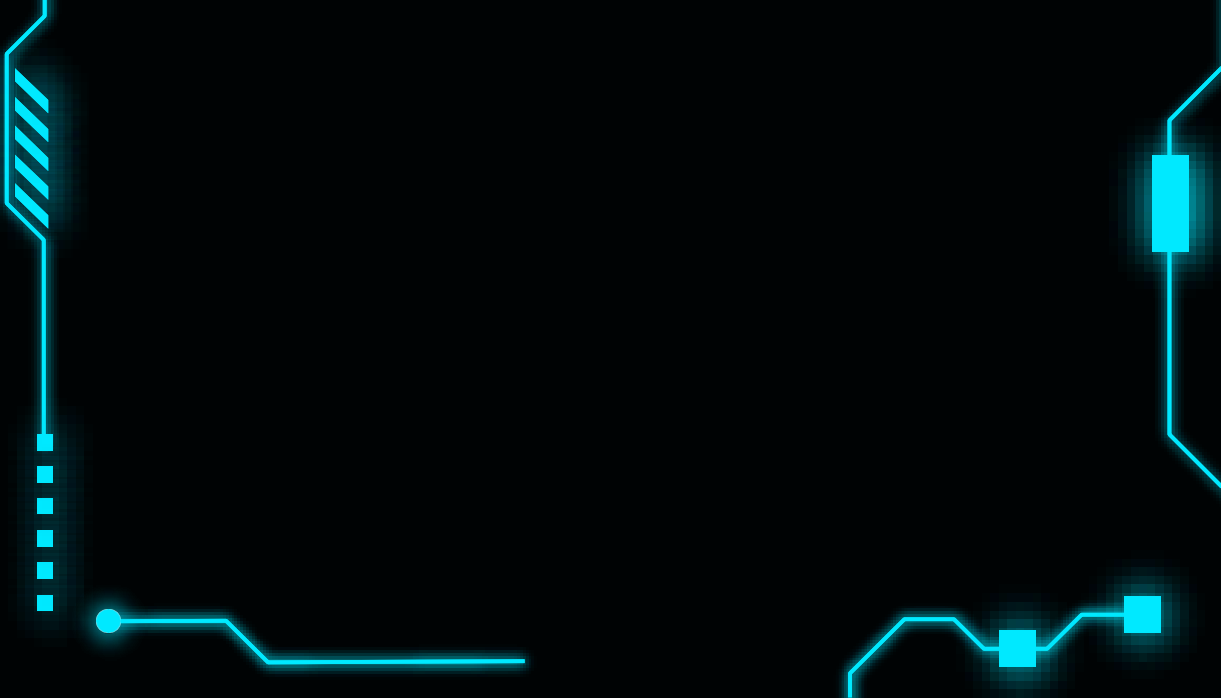
ERROS - Exemplos

```
try:  
    print("Olá Mundo!!")  
except:  
    print("Alguma coisa correu mal!!")  
else:  
    print("Correu tudo bem!")
```

```
try:  
    print(x)  
except:  
    print("Alguma coisa correu mal!!")  
finally:  
    print("O bloco 'try except' terminou!")
```

ERROS - Exemplos

```
try:
    f = open("teste.txt")
    try:
        f.write("Estou a escrever no ficheiro!")
    except:
        print("Alguma coisa correu mal!!")
    finally:
        f.close()
except:
    print("Alguma coisa correu mal na abertura do ficheiro!")
```



Módulos

Módulos em Python

Módulo = biblioteca de código

Ficheiro .py que podemos referenciar e incluir na nossa aplicação – Basta criar o ficheiro

Módulos em Python - Exemplo

1. Guardar o seguinte código num ficheiro `mymodule.py`

```
def greeting(name):  
    print("Hello, " + name)
```

2. Importar o modulo chamando o `mymodule` do seguinte modo:

```
import mymodule  
  
mymodule.greeting("Jonathan")
```

Variáveis num módulo

3. Guardar o seguinte código no ficheiro `mymodule.py`

```
person1 = {  
    "name": "John",  
    "age": 36,  
    "country": "Norway"  
}
```

4. Chamar o Código `mymodule.py`

```
import mymodule  
  
a = mymodule.person1["age"]  
print(a)
```

Nomear um módulo

Pode criar um **alias** ao importar um módulo, usando **as** como palavra-chave

Crie um alias para `mymodule` chamado `mx` :

```
import mymodule as mx  
  
a = mx.person1["age"]  
print(a)
```