

Fundamentos da Programação

Aula 9

FUNÇÕES

Abstração procedimental. Erros. Módulos. Exemplos.

ALBERTO ABAD, IST, 2024-25

Funções

Abstração procedimental

- As funções permitem aos programadores pensar no **que** (faz a função) e não no **como** (a função é implementada).

Exemplo #1

```
def soma_prog_arit(n):  
    iter = 1  
    soma = 0  
    while iter <= n:  
        soma = soma + iter  
        iter = iter + 1  
    return soma  
  
def soma_prog_arit(n):  
    if n < 1:  
        return 0  
    else:  
        return n*(n+1)//2
```

Funções

Abstração procedimental

- As funções permitem aos programadores pensar no **que** (faz a função) e não no **como** (a função é implementada).

Exemplo #2

```
def quadrado(x):  
    return x * x  
  
def quadrado(n):  
    pass # complete
```

```
In [4]: def quadrado_v1(x):  
        pass  
  
def quadrado_v2(x): # **  
    return x**2  
  
def quadrado_v3(x): # pow  
    return pow(x, 2)  
  
def quadrado_v4(x): # iter  
    soma, n = 0, x  
    while n > 0:  
        soma += x  
        n -= 1  
    return soma  
  
print(quadrado_v1(4), quadrado_v2(4), quadrado_v3(4), quadrado_v4(4))
```

None 16 16 16

Funções

Erros/Exceções

- Nas aulas anteriores falamos dos tipos de erros: sintaxe, semântica e *runtime*
- As funções podem *lançar* erros quando os argumentos utilizados são de tipo inválido e/ou estão fora do domínio.
 - As exceções **interrompen o fluxo de execução**, o que não acontece se fizermos um simples *print*
- Para isso podemos utilizar a instrução *raise* que gera um erro de execução, em BNF:

```
<instrução raise> ::= raise <nome>(<mensagem>)  
<mensagem> ::= <cadeia de caracteres>
```

- *nome* corresponde à identificação de um dos tipos de erros (ou exceções) conhecidos pelo Python (ou a novos tipos de erros definidos pelo programador):

AttributeError, IndexError, KeyError, NameError, SyntaxError, ValueError e ZeroDivisionError.

Funções

Erros/Exceções

Nome	Situação correspondente ao erro
AttributeError	Referência a um atributo não existente num objeto.
ImportError	Importação de uma biblioteca não existente.
IndexError	Erro gerado pela referência a um índice fora da gama de um tuplo ou de uma lista.
KeyError	Referência a uma chave inexistente num dicionário.
NameError	Referência a um nome que não existe.
SyntaxError	Erro gerado quando uma das funções <code>eval</code> ou <code>input</code> encontram uma expressão com a sintaxe incorreta.
ValueError	Erro gerado quando uma função recebe um argumento de tipo correto mas cujo valor não é apropriado.
ZeroDivisionError	Erro gerado pela divisão por zero.

Tabela 3.3: Alguns dos identificadores de erros em Python.

- Python (como outras linguagens) fornecem um *protocolo* para tratar das exceções (*try/except*) que veremos nas próximas semanas

Funções

Erros/Excepções, Exemplo:

```
In [13]: ## Definição da função
         def invert(x):
             return 1/x

#invocação/chamada
invert(20)
```

```
-----
ZeroDivisionError                                Traceback (most recent call last)
Cell In [13], line 10
      7     return 1/x
      9 #invocação/chamada
----> 10 invert(0)

Cell In [13], line 6, in invert(x)
      4     raise ValueError("argumentos invalidos: precisa ser um número")
      5 elif x== 0:
----> 6     raise ZeroDivisionError("argumento errado: nao pode ser 0")
      7 return 1/x

ZeroDivisionError: argumento errado: nao pode ser 0
```

Funções

Módulos: Importar

- Não é preciso reinventar a roda, Python fornece um grande número de bibliotecas (*libraries*) ou módulos com funções que podemos importar:
- Lista de módulos disponíveis por omissão: <https://docs.python.org/3/py-modindex.html>

```
<instrução import> ::=  
    import <módulo> {as <nome>} NEWLINE |  
    from <módulo> import <nomes a importar> NEWLINE  
  
<módulo> ::= <nome>  
  
<nomes a importar> ::= * | <nomes>  
  
<nomes> ::= <nome> | <nome>, <nomes>
```

Funções

Módulos: Aceder funções dum módulo

- Necessário no caso de *import* (sem *from*):

```
<composed name> ::= <simple name>.<simple name>
```

Exemplos:

```
>>> import math
>>> math.pi
3.141592653589793
>>> math.sin(math.pi/2)
1.0

>>> from math import pi, sin
>>> pi
3.141592653589793
>>> sin(pi/2)
1.0
```

```
In [25]: math.sin(pi)
```

Out[25]:

```
1.2246467991473532e-16
```


Funções

Módulos: Construir módulos

- Colocar funções num ficheiro `.py` (ex: `fp_utils.py`)
- Importar utilizando o nome do ficheiro/módulo (sem extensão):

```
>>> import fp_utils  
>>> fp_utils.soma_prog_arit(100)  
5050
```

```
In [ ]:
```

Funções

Funções e parâmetros em Python ++ (opcional)

- Python permite maior flexibilidade na definição e passagem dos parâmetros numa função:
 - **Default parameters**
 - **Keyword arguments**
 - Número variável de parâmetros posicionais e keyword (não nesta disciplina)

```
In [ ]: def dividir(num, den):  
        return num/den  
  
print("Ex1:", dividir(10,2))  
print("Ex2:", dividir(10))  
print("Ex3:", dividir())  
print("Ex4:", dividir(den=4))
```

A treinar mais!!!!

Funções

Exemplo 1, Tabela conversão temperaturas

- Escreva uma função que converta temperaturas de Fahrenheit para Celsius $C = 5 * (F - 32) / 9$
- Escreva uma função que recebe uma temperatura mínima e máxima (inteiros) em Fahrenheit e imprime a tabela de conversão para Celsius

```
In [ ]: def fahr_para_cent(fahr):  
        pass  
  
def tabela(min_f, max_f):  
    pass  
  
tabela(40, 50)
```

Funções

Exemplo 2, Potência de dois números inteiros

- Solução iterativa para k positivos
- E para k negativos?
- E para qualquer k ?

```
In [7]: def potencia(x, n):  
        pass  
  
        # Power of two numbers inteiros  
x = eval(input("Escreva a base da potencia: "))  
n = eval(input("Escreva a potencia: "))  
  
print(potencia(x, n))
```

```
Escreva a base da potencia: 2  
Escreva a potencia: -4  
0.0625
```

Funções

Exemplo 3, Factorial

- Para números inteiros não negativos ($\text{fact}(0) = 1$) iterativo

```
In [ ]: def factorial(x):  
        pass  
  
x = eval(input('Inteiro: '))  
f = factorial(x)  
print(f)
```

Funções - Tarefas próxima aula

- Trabalhar matéria apresentada hoje
- Fazer todos os exercícios/programas

