

Itaú Unibanco

Itaú

Programa de formação

ITAú analytics.



Módulo I – Fundamentos Computacionais
Sessão 2 - Aula 1a - Entrada e Saída em Python
Prof. Dr. Luiz Alberto Vieira Dias
Prof. Dr. Lineu Mialaret

Entrada e Saída em Python

- A linguagem de programação Python disponibiliza de funções pré-construídas que permitem a realização de entrada e saída de forma simples
 - Função `print()`
 - Função `input()`
- A sintaxe da função `print()` é

```
print(*objects, sep=' ', end='\n', file=sys.stdout, flush=False)
```

Função print()

- A função print() é utilizada para gerar saída padrão para a console
 - Ela imprime uma sequência qualquer de argumentos, separados por espaços e seguido por um caractere de nova linha (*trailing newline character*)

- Exemplo:

```
>>> print ('maroon' , 5)  
( 'maroon', 5)
```

- Os argumentos da função não necessitam ser strings
 - Um argumento x que não é string será mostrado como str(x)
- Sem argumentos, print () imprime um caractere de nova linha

Função print() (cont)

- A função print() pode ser customizada por meio de parâmetros chave (*keyword parameters*)
 - Por padrão, a função print() insere espaços na saída entre cada par de argumentos recebidos
 - O separador pode ser customizado com algum separador desejado usando-se o parâmetro chave sep

```
>>> a = "texto"
>>> b = "correto"
>>> c = "impresso"
>>> print (a,b,c)           # with spaces
>>> print (a,b,c,sep=':')  # with :
>>> print (a,b,c,sep='')   # without spaces
texto correto impresso
texto:correto:impresso
textocorretoimpresso
```

Função print() (cont.)

- Por padrão, a função print() se posiciona em uma nova linha depois de apresentar o argumento final
 - Uma string de posicionamento diferente pode ser atribuída usando-se o parâmetro chave end
 - Se end = '' não há o posicionamento em nova linha

```
>>> a = 'texto'
>>> b = "correto"
>>> c = "impresso"
>>> print (a,b,c)
texto correto impresso
>>> print (a,b,c,end="")
texto correto impresso>>>
```

Função print() (cont.)

- Por padrão, a função print() envia a saída para a console padrão
- Entretanto, a a saída da função print() pode ser direcionada para um arquivo, usando-se o parâmetro chave file
- O padrão é a saída para sys.stdout

```
>>> sourceFile = open('d:\python.txt', 'w')  
>>> print('Pretty cool, huh!', file = sourceFile)  
>>> sourceFile.close()
```

Função print() (cont.)

- As vezes pode ser necessário formatar a saída para torná-la mais atrativa
- O método `str.format()` é utilizado para realizar tal formatação
- O método é visível para qualquer objeto string
- `{}` são utilizados como marcadores (*placeholders*)

```
>>> x = 5
>>> y = 10
>>> print('The value of x is {} and y is {}'.format(x,y))
The value of x is 5 and y is 10
```

Função print() (cont.)

- Pode-se especificar a ordem com que as strings serão impressas usando números (índice de tupla)
- Pode-se utilizar também de argumentos chaves para formatação da string

```
>>> # Output: I love bread and butter
... print('I love {0} and {1}'.format('bread','butter'))
I love bread and butter
>>>
>>> # Output: I love butter and bread
... print('I love {1} and {0}'.format('bread','butter'))
I love butter and bread
```

```
>>> print('Hello {name}, {greeting}'.format(greeting = 'Good morning', name = 'John'))
Hello John, Good morning
```

Função print() (cont.)

- Pode-se também formatar a string de saída como no estilo da linguagem C (`sprintf () style`)
- Utilizar o operador % para implementar este tipo de formatação (uma sobrecarga do operador)

```
>>> x = 12.3456789
>>> print('The value of x is %f' %x)
The value of x is 12.345679
>>> print('The value of x is %3.2f' %x)
The value of x is 12.35
>>> print('The value of x is %3.4f' %x)
The value of x is 12.3457
>>> print('The value of x is %3.2e' %x)
The value of x is 1.23e+01
>>> print('The value of x is %d' %x)
The value of x is 12
```

Função print() (cont.)

Format	Meaning
%s	a string
%d	an integer
%0xd	an integer padded with x leading zeros
%f	decimal notation with six decimals
%e	compact scientific notation, e in the exponent
%E	compact scientific notation, E in the exponent
%g	compact decimal or scientific notation (with e)
%G	compact decimal or scientific notation (with E)
%xz	format z right-adjusted in a field of width x
%-xz	format z left-adjusted in a field of width x
%.yz	format z with y decimals
%x.yz	format z with y decimals in a field of width x
%%	the percentage sign % itself

Método format()

- Formatação em strings
 - O método format(), já apresentado, é disponível para objetos do tipo string sendo muito utilizado para se formatar strings
 - Podem ser usados argumentos posicionais ou palavras-chave para se especificar a ordem de formatação

```
>>> # default(implicit) order
... default_order = "{}, {} and {}".format('John','Bill','Sean')
>>> print('\n--- Default Order ---')
--- Default Order ---
>>> print(default_order)
John, Bill and Sean
>>>
>>> # order using positional argument
... positional_order = "{1}, {0} and {2}".format('John','Bill','Sean')
>>> print('\n--- Positional Order ---')
--- Positional Order ---
>>> print(positional_order)
Bill, John and Sean
>>>
>>> # order using keyword argument
... keyword_order = "{s}, {b} and {j}".format(j='John',b='Bill',s='Sean')
>>> print('\n--- Keyword Order ---')
--- Keyword Order ---
>>> print(keyword_order)
Sean, Bill and John
```

Método format() (cont.)

- Formatação em strings
 - O método format() disponibiliza uma série de especificações de formato adicionais

```
>>> # formatting integers
... "Binary representation of {0} is {0:b}".format(12)
'Binary representation of 12 is 1100'
>>> # formatting floats
... "Exponent representation: {0:e}".format(1566.345)
'Exponent representation: 1.566345e+03'
>>> # round off
... "One third is: {0:.3f}".format(1/3)
'One third is: 0.333'
>>> # string alignment
... "|{:<10}|{: ^10}|{:>10}|".format('butter','bread','ham')
'|butter  | bread  |   ham|'
```

Função input()

- A função input() é modo básico de obter informação de uma console no Python.
- Sintaxe:

```
input([prompt])
```

- A função mostra um prompt, que é uma string passada como parâmetro opcional, e então espera até que o usuário entre com alguma sequência de caracteres seguido pela tecla enter ou return.
- A função então retorna a sequência de caracteres digitadas
- A função recebe como entrada strings e caso seja necessário se trabalhar internamente como números, deve-se a fazer necessária de string para dado numérico

Função input() (cont.)

- Exemplo:

```
>>> year = int(input( 'In what year were you born?' ))
```

- Como a função input() retorna string como resultado, o uso da função pode ser combinado com funcionalidades da classe string.

- Exemplo:

```
>>> reply = input( 'Enter x and y, separated by  
spaces:' )  
... pieces = reply.split( ) # returns a list of  
strings, as separated by spaces  
... x = float(pieces[0])  
... y = float(pieces[1])  
... print (x, y)  
...  
Enter x and y, separated by spaces:>? 12 14  
12.0 14.0
```