

AC-309 – Atividades Complementares

(Engenharias Biomédica e de Telecomunicações)



2. Tipos de Dados, Variáveis e Operadores

Prof. Evandro Luís Brandão Gomes

A IDE PyCharm

Disponibilizada pela empresa JetBrains

Link para download: <https://www.jetbrains.com/pycharm/>

Versões: Professional ou Free Community



A IDE PyCharm



Version: 2019.3.3
Build: 193.6494.30
6 February 2020

[System requirements](#)

[Installation Instructions](#)

[Other versions](#)

Download PyCharm

[Windows](#) [Mac](#) [Linux](#)

Professional

For both Scientific and Web Python development. With HTML, JS, and SQL support.

[Download](#)

Free trial

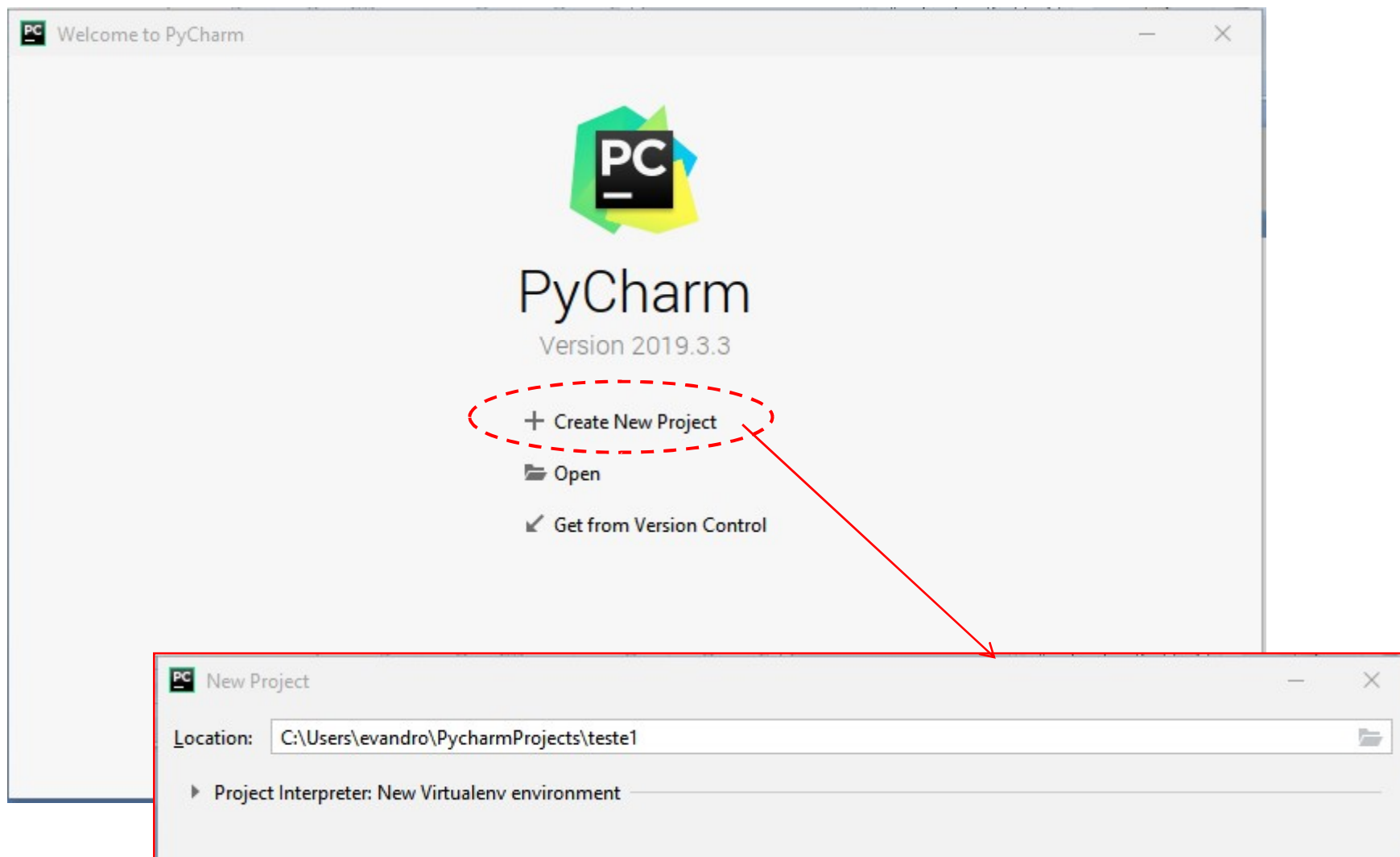
Community

For pure Python development

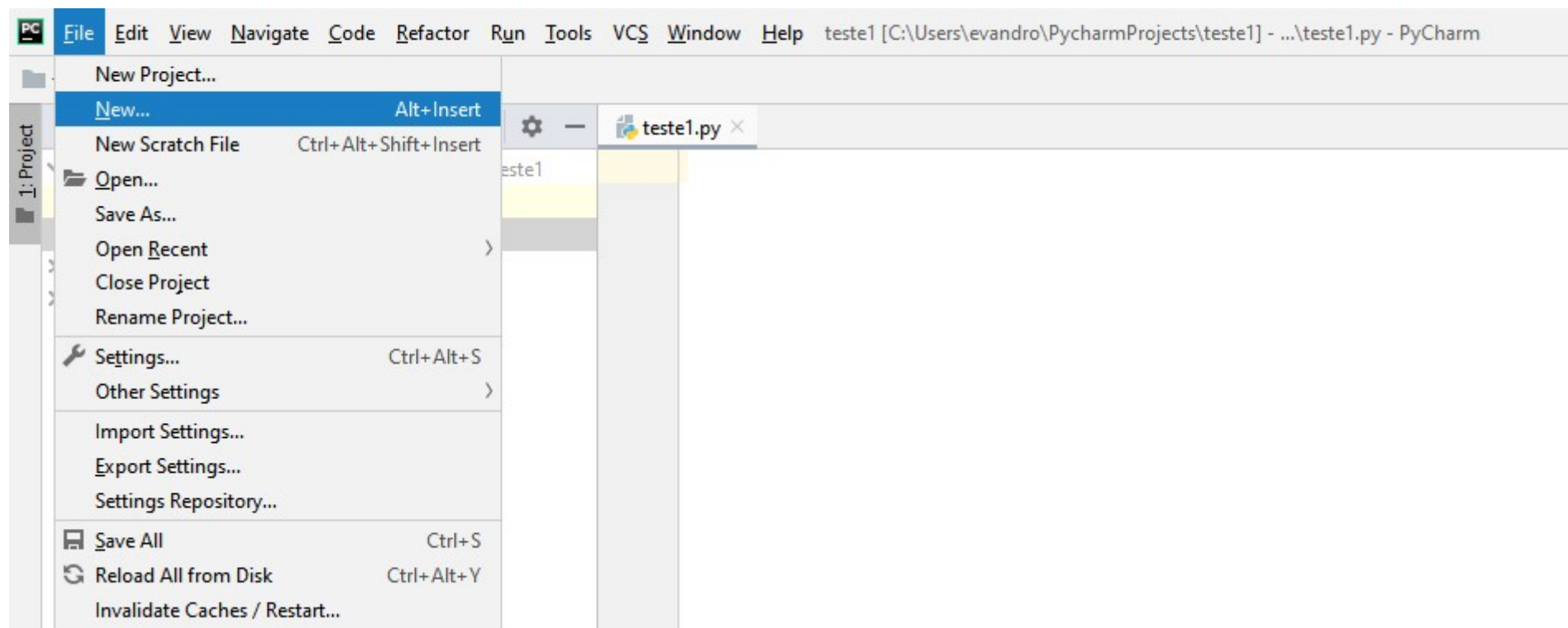
[Download](#)

Free, open-source

A IDE PyCharm



A IDE PyCharm



A IDE PyCharm

- O PyCharm oferece muitas facilidades e recursos ao programador. Caso queira customizar a IDE (cores, tamanho das letras, etc), vá em:

File -> Settings

- Para executar um script python vá em:

Run -> Run

ou clique em cima do nome do script com o botão direito e use Run ou seta verde para direita

Tipos de dados do python

Os Tipos Primitivos de Dados da Linguagem python são:

- **int**: números inteiros
- **float**: números reais;
- **bool**: valores booleanos (True / False);
- **str**: Conjunto de caracteres (string)

Não existe declaração de variáveis. Ao inicializar uma variável ela assume seu tipo de acordo com o dado inicializado.

Para saber o tipo de uma variável, usar comando: *type(variável)*

Tipos de dados do python

Para as variáveis tipo str (string) em python existem vários métodos de verificação dos valores de seus caracteres. A seguir alguns:

- *var.isnumeric()* → retorna True se o conteúdo da variável (var) é composto apenas de caracteres numéricos;
- *var.isalpha()* → retorna True se o conteúdo da variável (var) é composto apenas de caracteres alfabéticos;
- *var.isalnum()* → retorna True se o conteúdo da variável (var) é composto de caracteres alfabéticos e numéricos;
- *var.isupper()* → retorna True se o conteúdo da variável (var) é composto apenas de caracteres alfabéticos maiúsculos;
- *var.islower()* → retorna True se o conteúdo da variável (var) é composto apenas de caracteres alfabéticos minúsculos;

Operadores Aritméticos

Os operadores aritméticos do python são:

+	→	adição
-	→	subtração
*	→	multiplicação
/	→	divisão
**	→	potenciação
//	→	divisão inteira
%	→	resto da divisão

Os operadores + e * podem ser usados com objetos do tipo str:

Exemplos:

```
X="AB"  
print(X*5)  
ABABABABAB
```

```
print(X+"CD")  
ABCD
```

Funções Embutidas

A linguagem python possui várias funções embutidas no interpretador, que sempre estão disponíveis.

		Funções embutidas		
<code>abs()</code>	<code>delattr()</code>	<code>hash()</code>	<code>memoryview()</code>	<code>set()</code>
<code>all()</code>	<code>dict()</code>	<code>help()</code>	<code>min()</code>	<code>setattr()</code>
<code>any()</code>	<code>dir()</code>	<code>hex()</code>	<code>next()</code>	<code>slice()</code>
<code>ascii()</code>	<code>divmod()</code>	<code>id()</code>	<code>object()</code>	<code>sorted()</code>
<code>bin()</code>	<code>enumerate()</code>	<code>input()</code>	<code>oct()</code>	<code>staticmethod()</code>
<code>bool()</code>	<code>eval()</code>	<code>int()</code>	<code>open()</code>	<code>str()</code>
<code>breakpoint()</code>	<code>exec()</code>	<code>isinstance()</code>	<code>ord()</code>	<code>sum()</code>
<code>bytearray()</code>	<code>filter()</code>	<code>issubclass()</code>	<code>pow()</code>	<code>super()</code>
<code>bytes()</code>	<code>float()</code>	<code>iter()</code>	<code>print()</code>	<code>tuple()</code>
<code>callable()</code>	<code>format()</code>	<code>len()</code>	<code>property()</code>	<code>type()</code>
<code>chr()</code>	<code>frozenset()</code>	<code>list()</code>	<code>range()</code>	<code>vars()</code>
<code>classmethod()</code>	<code>getattr()</code>	<code>locals()</code>	<code>repr()</code>	<code>zip()</code>
<code>compile()</code>	<code>globals()</code>	<code>map()</code>	<code>reversed()</code>	<code>__import__()</code>
<code>complex()</code>	<code>hasattr()</code>	<code>max()</code>	<code>round()</code>	

Acesse <https://docs.python.org/pt-br/3/library/functions.html> para detalhes de cada função

Funções Embutidas

Alguns exemplos:

bin(x) – converte um inteiro em binário

```
>>> bin(10)
0b1010
```

hex(x) – converte um inteiro em hexadecimal

```
>>> hex(255)
0xff
```

len(s) – devolve o tamanho de uma string

```
>>> x = "INATEL 2020"
>>> print(len(x))
11
```

A Biblioteca Padrão do python

A biblioteca padrão do python é muito extensa, oferecendo uma ampla gama de recursos que pode ser vista em:

[**https://docs.python.org/pt-br/3/library/**](https://docs.python.org/pt-br/3/library/)



Além da biblioteca padrão, há uma coleção crescente de vários milhares de componentes (de programas e módulos individuais a pacotes e frameworks de desenvolvimento de aplicativos inteiros), disponíveis em [**https://pypi.org/**](https://pypi.org/)

A Biblioteca Padrão do python

Importando uma Biblioteca ou parte dela para o script:

import <biblioteca>

→ importa a biblioteca inteira

from <biblioteca> import <parte>

→ importa apenas o recurso indicado

Exemplo:

```
import math  
num = 2.583  
print(math.trunc(num))  
print(math.ceil(num))  
print(math.floor(m))
```

Alguns Recursos da Biblioteca MATH

ceil -> arredonda um número para cima;
floor -> arredonda um número para baixo;
trunc -> pega apenas a parte inteira de um número;
*pow -> potência (funciona igual ao operador **);*
sqrt -> retorna a raiz quadrada de um número;
factorial -> retorna o fatorial de um número;

A Biblioteca Padrão do python

Exercícios:

Faça um script ou use o modo comando, para ler um número inteiro e calcular:

- O $\log_{(10)}$ do número;
- O $\log_{(2)}$ do número;
- A raiz quadrada do número;
- O seno do número;
- Mostre a constante “math.pi” e “math.e”

OBS: pesquise as funções no site indicado

Manipulando Strings

Uma string (cadeia de caracteres) é armazenada da seguinte forma:

var="Hello World"

H	e	l	l	o		W	o	r	l	d
0	1	2	3	4	5	6	7	8	9	10

O python oferece várias formas de acessar partes da string:

- *var[6] -> captura a letra W da nossa String exemplo;*
- *var[:5] -> captura a palavra Hello;*
- *var[6:11] -> captura a palavra World; (6 inclusive e 11 exclusive);*
- *var[6:] -> também captura a palavra World;*
- *var[0:10:2] -> mostra HloWrD (ou seja, pula de 2 em 2);*
- *var[6::3] -> mostra Wl (dentro da palavra World, pula de 3 em 3);*

Manipulando Strings

Funções e métodos de manipulação de strings:

var="Hello World"

H	e	l	l	o		W	o	r	l	d
0	1	2	3	4	5	6	7	8	9	10

- *len(var) -> retorna o número de caracteres em uma string;*
- *var.count('o') -> conta o número de o's da nossa palavra;*
- *var.count('l',0,5) -> conta quantos l's tem dentro de Hello;*
- *var.find('lo') -> indica a posição de onde começa lo;*
- *"World" in var -> retorna true se a palavra World estiver dentro de var;*
- *var.replace('World','Python') -> Troca a palavra World po Python;*
- *var.upper() -> troca todas as letras para MAIÚSCULAS;*
- *var.lower() -> troca todas as letras para minúsculoas;*
- *var.split() -> quebra a frase em partes;*

Manipulando Strings

Exercício:

Faça um script em python que leia um nome completo de uma pessoa e mostre:

- *O nome com todas as letras maiúsculas;*
- *O nome com todas as letras minúsculas;*
- *O total de letras do nome;*
- *Mostrar quantas letras “a” existe no nome*
- *Trocar o ultimo sobrenome por “do Inatel”*

Manipulando Strings

Outras funções de manipulação de strings:

Método	Descrição	Exemplo
<code>capitalize()</code>	Retorna a string com a primeira letra maiúscula	<code>a = "python"</code> <code>a.capitalize()</code> 'Python'
<code>swapcase()</code>	Inverte o conteúdo da string (Minúsculo / Maiúsculo).	<code>k = "Python"</code> <code>k.swapcase()</code> 'pYTHON'
<code>title()</code>	Converte para maiúsculo todas as primeiras letras de cada palavra da string.	<code>l = "apostila de python"</code> <code>l.title()</code> 'Apostila De Python'
<code>split()</code>	Transforma a string em uma lista, utilizando os espaços como referência.	<code>m = "cana de açúcar"</code> <code>m.split()</code> ['cana', 'de', 'açúcar']

Manipulando Strings

Outras funções de manipulação de strings:

Método	Descrição	Exemplo
ljust()	Ajusta a string para um tamanho mínimo, acrescentando espaços à direita se necessário.	<pre>p = "Python" p.ljust(15) 'Python '</pre>
rjust()	Ajusta a string para um tamanho mínimo, acrescentando espaços à esquerda se necessário.	<pre>q = "Python" q.rjust(15) ' Python'</pre>
center()	Ajusta a string para um tamanho mínimo, acrescentando espaços à esquerda e à direita, se necessário.	<pre>r = "Python" r.center(10) ' Python '</pre>
lstrip()	Remove todos os espaços em branco do lado esquerdo da string.	<pre>s = " Python " s.lstrip() 'Python '</pre>
rstrip()	Remove todos os espaços em branco do lado direito da string.	<pre>t = " Python " t.rstrip() ' Python'</pre>
strip()	Remove todos os espaços em branco da string.	<pre>u = " Python " u.strip() 'Python'</pre>