**Python**

**Qual python usar Ordem de pior ao melhor.**

Tem diversos programas: python 3.6, Spyder que vem no python anaconda, Pycharm, Atom(pesado) e Visual Studio Code que o blogger disse usar, melhor e pesado também. Olhar link abaixo pra entender caso ainda funcione:

<https://paulovasconcellos.com.br/quais-s%C3%A3o-as-melhores-ide-para-python-confira-a-lista-279b54bef301>

**Introdução**

**Cores/tipos: Palavras em Roxo/Funções embutidas:**(não precisa programalas) obs: função()

**Verde/texto(String):** Tudo entre aspas**. Variaveis/preto:** São apenas nomes para objetos.

**Diretivas/amarelas:** (if,else,while): controlam o fluxo do código que esta rodando.

**Identação/espaços. Atribuição=(**Faz com que um valor passe para a variável).

**Comparação: ==(**Passa a ser uma pergunta).

**ERROS:**

**Erro de sintaxe:** Erro de funções e pontuações. **Solução:** Programando mais.

**Erro em tempo de execução:** Quando o arq não esta ou não tem internet caso o programa precise acessa-lo. **Solução:** Fazendo tratamento de exceções

**Erros semânticos:** Logica de programação incorreta. **Solução:** Phyton tutor

* **print (‘Alô mundo’) :** O comando só funciona com letra minúscula e com aspas(tendo em mente que também pode ser aspas duplas) e sem espaço.
* Em python os espaço são para definir blocos de comandos.
* Pode fazer contas no programa ( \* é multiplicação,/ divisão,\*\* elevado).
* **Type(42):** Mostra classe do numero, que no caso é int.
* **Type(3.14):** Mostra que é classe float.
* **Type(‘abacate’):** Mostra que é da classe str(texto).
* As informações escritas já estão na memoria mas não podem ser manuseadas, para isso servem as variáveis.
* **a=42:** O 42 já estava na memoria e chamou de “a”, sendo assim estando no mesmo local.
* **id(a):** ele vai te dar o endereço do objeto, sendo o mesmo que o id(42).
* **2\*a:**84 (você pode manipular a variável).
* Indo do modo interativo para o modo de edição .  **file>new window.**
* No modo edição você edita as linhas e executa podendo ser soma de texto também.
* **dir(‘abacate’):** ele vai te dar todos os métodos aplicáveis a este texto.(pegando o upper como exemplo.)
* **help(‘abacate’.upper)=** ira exibir a função dele.
* **help(print)=**mostra a função do print.
* **a>b:**  ele da a relação de duas variáveis.(True)
* **type(true):** é da classe bool (uma classe que só te 2 objetos \*vdd e falso\*).
* **a==42:** é uma comparação.
* **a!=42:** Diferente.
* **And e or: (e) e (ou).**
* **Alt+P:** Mostra comando antigos alt+n mostra pra frente.
* **MARCADORES:** print(‘o numero é’ ,a) print(‘O numero’,a,’é muito legal’) print(‘O numero %d é muito legal’ %a) print(‘Uma fruta muito gostosa é o %s’ %a).
* **Str(42)+’mamao’ :**Para mudar numero pra texto.
* **A,b=b,A:** para trocar as variáveis sem perder nenhum numero.
* **A,b,c=1,2,3**
* **A,b,c=’OBA’:** a=O , b=B e c=A.
* **www.pythontutor.com**:Depurando programa.
* **nome= input (‘Digite o seu nome: ’):** Entrada de dados(retorna STR).
* **N1=int(input(‘Digite um numero para soma-lo))’**Para números (float para num quebrados).
* **#frase**  Modo de fazer comentário no python.
* **Len(a):** Lê quantas variáveis na variavel “a”.
* **If:**Sempre colocar (:) depois da condição do if.
* ‘’’: com 3 aspas em cima e embaixo é uma comentário de varias linhas.
* **Elif:** deixa a condição reta (você usa entre if e else).
* **While true:** o código não para o laço
* **Break:** Para o laço.
* **Lista=[]** lista de lugares (vazia)
* **Notas=[0,8,5,3**,] lista com 4 notas
* **Lista.append(‘nova variavel’)**
* **Lista.insert(0,”X”)=** insere X no índice 0 da lista.
* **Lista[0]=Y =** Troca o X pelo Y
* **Lista.pop(0)=**remove o elemento do índice 0.
* **Del(Lista[0])=**apaga conteúdo do índice 0.
* **Lista.remove(Y)=** remove o conteúdo Y.
* **Print(lista[2]);** nova variável
* **Nota[0]= 10** mudando nota 0 para 10.
* **Soma += notas**  é um incremento que é igual **soma=soma+notas**
* **Not in: Procure no código consoantes.**
* **X=’0123456789’ >> print(x[0:2]) = 01**  chama-se fatiamento,
* **Print(x[:2])=01 >> print(x[4:])= 456789 print(x[4:-1])=45678**
* **Print(x[-4:-1])=678 >> print(x[:])=0123456789**
* **Texto=’batatinha quando nasce’**
* **Texto[::2]:pulando de 2 em 2**
* **Texto[::-1]: invertendo a string.**
* **Concatenação: mudar letra da string criada(veja no programa concatenação)**
* **Arquivo.startwish(‘p’) :** pergunta se a variável arquivo(string) começa com p
* **Arquivo.**endwish(‘py’):pergunta se a variável arquivo(string)termina com py
* **Arquivo.upper()** deixa tudo maiúsculo
* **Arquivo.lower()** deixa tudo minúsculo
* **Arquivo.lower() in ‘sim não yes no’** eu quero saber o que o usuário digitou se esta nessas opções
* **S=’um tigre, dois tigres, três tigres’**
* **S.find(‘tigre’)=** procura na variável S a palavra tigre
* **S.find(‘tigre’,4)=**  procura a palavra apartir de uma parte da string(se retornar -1 ele não achou)
* **S.replace(‘tigre’,’gato’)=**troca as palavras mas não atribui.
* **S=S.replace(‘tigre’,’gato’):**  para mudar o texto original.
* **Txt=’batatinha quando nasce’**
* **Txt.split():**  ele pega palavra por palavra
* **Data:’21/02/2011’**
* **Data.split(‘/’) pega palavra por palavra (‘21’,’02’,’2011’)**
* **Ip= ‘198.188.10.144’**
* **Ip.split(‘.’)** separa por pontos tendo 4 elementos
* **Times=[‘palmeras’,’santos’,’corinthians’]**
* **JOIN GRUDA SPLIT SEPARA**
* **‘/’.join(times)** fica ‘palmeras/santos/corinthians’
* **For** letra **in** ‘aeiou’:
* **Print (letra) =** a e i o u
* **Range(5)=**  gera um intervalo de números inteiros (do 0 ao 4) tem um prog no \*for e while\*
* **Def e return=** cria sua própria função
* **Import :** chama modulo
* **Import(random)**  mexe com coisas aleatórias
* **Dir(random)**  mostra comandos com random
* **Random.randint(1,100)**  mostra um numero aleatório de 1 a 100
* **Para não precisar usar Random.randint(1,100) coloque no começo from random import randint, e use só x=randint(1,100)**
* **Random.choice([‘palmeiras’,’santos’, ‘sport’]):**  escolhe um nome aleatório
* **Random.shuffle(lista):** pega essa lista e mostra em ordem aleatória.
* **List(nome):** separa a palavra dentro da variável nome.
* **Lista.sort():** poe a lista em ordem
* **Ord(a):** mostra os valor do caráter
* **Chr(120):** mostra o caráter correspondente
* **Open :** Abre os arquivos. E os modos são r-leitura, w-escrita, a-append, b-binario,+(atualização)
* **Read e write :**Ler e escrever.
* **Close:** fechar arquivos
* **Arquivo.readlines():** gera uma lista onde cada elemento é uma linha lida.
* **Print(linha.rstrip()):** para quando ler o arquivo não pular 2 linhas.**OBS: no lugar do print(linha)**
* **HTML:**  tag html começam com <e terminam com >. A pag web é escrita entre <HTML>e </html> que é a tag de maior nível.
* **Arquivo=open(‘ola.html’,’w’,encoding=’utf-8’)=** o acento aparece corretamente com o encoding.
* **D={}**:Dicionario
* **D[‘a’]=’alpha’**
* **D.keys():**mostra só as chaves do D
* **D.values():**mostra só os valores.
* ‘g’ in D: pergunta se o G esta no dicionário ’d’
* **For chave in d: print (chave):** imprime as chaves
* **For chave in d{chve}: print (chave):**
* **Class televisão**: criou a classe televisão
* **\_\_init\_\_(self)**: funções reservadas, função construtora self é o obj criado
* **CLASSES**
* **Lista=list()**
* **Texto=str()**
* **K=int()**
* **X=float()**
* **Dic=dict()**
* **O que foi visto até agora?**
* **\* Condiçoes \*Repetiçoes \*Listas \*Strings \*Funções \*Arquivos \*Dicionários \*Classes e objetos.**
* **SCRAPING=**raspagem de dados da internet.
* **Urllib.request=**permite conversar com a internet.
* **Import antigravity:** acessar diretamente no navegador.
* **Time.clock():**hora atual em segundos.
* **Time.daylight():** estou no horário de verão?
* **Time.sleep(secs):** Dormir alguns segundos.
* **Time.timezone():**fuso horário.
* **Texto[4]:**acessa o 5º caráter
* **Texto[4:9]:**acessa do 5º ao 9 º caráter
* **Texto.find()**:procura um substring.
* **Float()**:
* **Funções básicas: procurar na pasta funções.**
* **Json(biblioteca)**
* **Sort:** ordena os dados
* **Reverse:** coloca em ordem decrescente
* **DICIONARIO:** exemplos na pastas dicionários.
* **.items** : já pega a chave e o valor do dicionário
* **.keys**
* **Sorted(notas):** Faz copia do conteúdo ordenado.
* **Import diss:** codigo em baixo nível (TWP410 python em baixo nível modulo dis).
* **PARA DEIXAR IF OU QUALQUER COISA EM BRANCO:** PASS
* **Assert:** Caso a condição não for cumprida ele da erro no código.
* **Sum([1,2,3]):**Faz a soma dos valores.
* **Zip:** pega elementos do mesmo índice.
* **Import** math
  + **math.pi**
  + **math.sqrt(x)-** Raiz quadrada de X
  + **math.fabs(x)**=valor absoluto de X
  + **math.factorial(x)**=Fatorial de X
  + **math.log10(x)=** Logaritmo de X na base 10
  + **math.pow(x,y)=** X elevado a Y
  + **math.ceil(x)=**Retorna o menor inteiro maior ou igual a X
  + **math.floor(x)=** Retorna o maior inteiro menor ou igual a X

SPYDER

Para gerar gráficos:

* **from pylab import \*:** pode importar essa... ou as duas juntas de abaixo.
* **import matplotlib.pyplot as plt**
* **import numpy as np**

Para características do gráfico:

* **plt.title(“”):**Para titulo
* **plt.plot(x , y ):** Para coordenadas do gráfico( de preferencia X e Y como listas)(para outra linha colocar o mesmo código embaixo com listas diferentes.)
* **plt.xlabel('nome'):** Para nome em X
* **plt.ylabel('ROUND'):** Para nome em Y
* **plt.grid(True):** Para habilitar grades no gráfico.
* **plt.show():**Para exibir gráfico
* **plt.rcParams['figure.figsize'] = (11,7):**Para tamanho do gráfico.(usar sempre 12,8)

**OBS:** No caso de alguma duvida olha nos códigos da iniciação.