



Guia Rápido Python

Anotações Python

Sumário

- [Anotações Python](#)
- [Sumário](#)
- [Anotações Gerais](#)
- [Tipos Primitivos](#)
- [Operações Aritméticas](#)
- [Ordem De Precedência](#)
- [Anotações Print](#)
- [Bibliotecas](#)
 - [Math](#)
 - [Random](#)
 - [Time](#)
 - [Datetime](#)
 - [Operator](#)
 - [Urllib](#)
- [Manipulação De Texto](#)
 - [Localização](#)
 - [Transformações](#)
 - [Divisão](#)
 - [Junção](#)
- [Estrutura Condicional Simples E Composta](#)
 - [Tipos](#)
 - [Sequencial](#)
 - [Pythonizar](#)
 - [Simples](#)
 - [Composto](#)
 - [Ninhadas](#)
 - [If](#)
- [Estrutura De Repetição Ou Laços Ou Iterações](#)
 - [Laço Com Variável De Controle](#)
 - [For](#)
 - [Laço Com Teste Lógico](#)
 - [While](#)
- [Coleções](#)



- Coleções
 - Anotações
 - Tuplas
 - Listas
 - Dicionários
- Cores No Terminal Python
 - Ansi
 - Style
 - Text
 - Back
- Conversão Base De Dados
- Funções
 - O Que É
 - Declaração Basica Função Sem Parâmetros
 - Declaração Basica Função Com Parâmetros
 - Empacotar Parâmetros - Receba Vários Parâmetros
 - Interactive Help
 - Docstrings
 - O Que É
 - Como Criar
 - Exemplo
 - Parâmetros Opcionais
 - O Que É
 - Como Criar
 - Exemplo
 - Escopo De Variáveis
 - Definição Escopo
 - Escopo Local
 - Escopo Global
 - Dica
 - Exemplo
 - Retorno De Resultados
 - Como Criar
 - Exemplo
- Modularização
 - O Que É
 - Focos
 - Como Criar

Anotações gerais

Refinamento sucessivo : particionar um aplicação e realizar testes em quando desenvolve, para por exemplo evitar erros no começo meio, e corrigi los enquanto em fase de desenvolvimento.

Inverter string digitada com : `var[::-1]`

Flag - Ponto de parada

Consegue-se receber o mesmo valor para várias variáveis utilizando:

```
ex = ex1 = ex2 = ex3 = 0
```

reverse = True - torna algumas funções invertidas ex: `L.sort()`

função == Método

toda função abre e fecha parênteses após o nome ex: `f()`

Tipos Primitivos

- int - inteiro
- bool - booleano / true , false
- float - números com vírgula
- str - cadeia de texto

Operações Aritméticas

- Adição: +
- Subtração: -
- Multiplicação: *
- Divisão: /
- Potência: **
- Divisão: //
- Resto da divisão/Módulo : %
- Comparação: ==
- Recebe: =

Ordem de precedência

1. :()
2. :**
3. :*,/,//,%
4. :+,-

Anotações Print

Bibliotecas

Math

`ceil(N)` : arredonda para cima
`floor(N)` : arredonda para baixo
`trunc(N)` : reduz casas decimais sem arredondar
`pow(N)` : potência
`sqrt(N)` : raiz quadrada
`factorial(N)` : fatorial

Random

`random()` : gera número aleatório entre 0 e 1
`randint(numero inicial, Numero Final)` : gera um número aleatório inteiro em que é possível escolher o range
`shuffle(L)` : Embaralha uma lista

Time

`sleep(segundos)` : faz o processo aguardar a quantidade de tempo definido antes de continuar a execução

Datetime

`date.today().year` : ano atual

Operator

`itemgetter` - usado para buscar itens dentro de dicionários

Urllib

`urllib.request.urlopen(Url de algum site)` - tenta acessar algum site
`variavel_com_url_site.getcode()` - Retorna um código para a tentativa de acesso sendo 200 bem sucedido.

Manipulação de texto

Localização

`frase[9:21]` - Retorna o valor de frase começando na posição 9 e indo até a 20 (o python desconsidera a última)
`len(f)` - quantidade de caracteres em uma string ou lista
`(T).count()` - conta a quantidade de um caracter dentro de uma string / list
`(T).find()` - mostra a posição de um caracter dentro de uma string / lista. se não existir ele retorna -1
`(T).index(Valor procurado, Início)` - mostra a posição do valor procurado dentro de uma variável composta
`in` - retorna valor booleano