

SEMANA PYTHON ISTA

*Seja disputado pelo
mercado e comece
ganhando 5x mais!*

APOSTILA 1

★ **DALTON PEIXOTO** ★

OBJETIVOS DA SEMANA PYTHONISTA



PRA QUEM É ESSE CURSO?

- ◆ Quem está entrando no mercado de trabalho da T.I.
- ◆ Deseja entrar na faculdade de Ciência da Computação, Análise de Sistemas, Engenharia Robótica ou relacionadas.
- ◆ Quer economizar tempo e ir direto ao ponto.
- ◆ Deseja conhecer melhor o Python e se desenvolver na linguagem.



Quem sou eu?

Meu nome é *Dalton Peixoto*.

Sou programador sênior com mais de 7 anos de experiência no mercado de desenvolvimento.

Possuo conhecimentos em Python, C++, Java, Javascript, PHP, Bancos de Dados e Business Intelligence.

Programador Aventureiro

É uma empresa educacional que atua no desenvolvimento pessoal e profissional de seus alunos na área da programação, TI e design.

Criada com o objetivo de disseminar a informação e capacitar programadores para o mercado de trabalho. Nascemos baseados no pilar da transparência e simplicidade e trazemos isso em todos os nossos cursos.



O QUE É UM ALGORITMO?

Formalmente, um algoritmo pode ser entendido como a sequência de instruções lógicas que chegam a solução de um problema.

Em outras palavras, é o passo-a-passo (detalhado) para resolvermos um problema.



LINGUAGEM DE PROGRAMAÇÃO

Assim como usamos as regras e padrões do Português para comunicarmos com outras pessoas...

Uma linguagem de programação é um conjunto de regras, padrões e instruções para comunicarmos com a máquina e gerarmos um software.



ALGORITMO VS CÓDIGO

Enquanto algoritmos são instruções para se resolver um problema...

Os códigos são as instruções para que a máquina resolva esse problema!

Ou seja, um código nada mais é que a implementação de um algoritmo usando uma linguagem de programação



O QUE É COMPILAÇÃO?

Quando temos um código feito em uma linguagem de programação precisamos que aquilo vire um programa executável (software)...

Essa transformação de código-fonte (o que você escreveu) em código-alvo (o que a máquina entende) é o que nós chamamos de Compilar o Código.



INTERPRETAÇÃO

Outra forma de executarmos um código é através da Interpretação.

Diferente da compilação, nesta etapa não traduzimos para a linguagem das máquinas, mas sim para uma linguagem intermediária (chamada de bytecode).

Feito isso, o interpretador vai conseguir ler e executar esse bytecode!



COMPILAR VS INTERPRETAR

Como um código compilado é traduzido para a linguagem da máquina, ele pode ser executado direto no CPU (sem precisar do interpretador).

Sendo assim, eles são os mais rápidos!

Já os código interpretados, como são executados pelos interpretadores, são independentes do Sistema Operacional.

Isto é, o mesmo código pode rodar tanto em Windows quanto Linux, MacOS ou na sua geladeira!



Python

O Python é uma linguagem interpretada, tendo o mesmo código rodando em todos os Sistemas Operacionais existentes e com a filosofia de “quanto mais simples melhor”, sendo uma das linguagens mais fáceis de se usar.

Além disso, o Python é totalmente gratuito e com código aberto, tendo alto apelo da comunidade de desenvolvedores, o que a fez se tornar uma das linguagens mais populares, poderosas e versáteis do mundo!





Nessa aula você irá aprender:

- O que são variáveis e como defini-las no Python.
- Quais as regras de nomenclatura de variáveis
- Como receber dados digitados pelo usuário no seu programa
- Converter dados entre diferentes tipos
- Operadores aritméticos e como utilizá-los para fazer cálculos

Além disso também verá dois projetos práticos:

- Calculadora de IMC
- Calculadora de média de notas de um aluno

E no final terá exercícios para praticar tudo o que aprendeu!

O que são variáveis?

Variável é um local de memória reservado que se utiliza para guardar alguma informação

```
In [3]: a = 1
```

```
In [4]: a
```

```
Out[4]: 1
```

Ao tentarmos chamar uma variável que ainda não foi definida o Python levantará um erro


```
In [5]: b
```

```
-----  
-----  
NameError                                Traceback (most recent c  
all last)  
<ipython-input-5-89e6c98d9288> in <module>()  
----> 1 b  
  
NameError: name 'b' is not defined
```

```
In [6]: b = 5
```

Podemos usar as variáveis para diversos fins como fazer operações matemáticas por exemplo, em suma o que está acontecendo é que o Python está utilizando os valores armazenados nessas variáveis para fazer os cálculos

```
In [7]: a + b # 1 + 5
```

```
Out[7]: 6
```

Uma string é uma cadeia de caracteres e todo texto que quisermos atribuir a uma variável no Python sempre será uma string:

- Definimos uma string usando aspas para delimitar seu início e seu fim
- Tanto faz utilizar aspas simples ou duplas
- Podemos escrever uma string em mais de uma linha colocando ela entre parênteses ou utilizando uma tríplice de aspas simples (ou duplas) no seu início e final

```
In [8]: texto = "todo texto é uma string"
```

```
In [9]: texto
```

```
Out[9]: 'todo texto é uma string'
```

```
In [10]: type(texto)
```

```
Out[10]: str
```

```
In [11]: type(a)
```

```
Out[11]: int
```

Números também podem ter tipos diferentes como valores inteiros ou reais (que chamamos de **ponto flutuante**)

```
In [12]: decimal = 1.123
```

```
In [13]: type(decimal)
```

```
Out[13]: float
```

Atribuição de Dados

A atribuição de dados a uma variável é feita utilizando um símbolo de igualdade onde o lado direito é o valor que será atribuído e o lado esquerdo é a variável que receberá esse valor

- `variavel = valor`

Outro tipo de dados que podemos salvar em variáveis são valores booleanos, no Python as duas palavras reservadas da linguagem para esse tipo de dado são **True** e **False**

```
In [14]: c = True
```

```
In [15]: type(c)
```

```
Out[15]: bool
```

Também obtemos um valor booleano quando fazemos algum tipo de comparação, por exemplo, se quisermos comparar se o valor da variável **b** é igual a **11** teríamos uma resposta booleana do programa retornando:

- **True** se o valor de **b** for igual a **11**
- **False** caso não seja

```
In [16]: b == 11
```

```
Out[16]: False
```

Nomenclatura de variáveis

- Começar com letra ou underscore
- Pode conter números mas não começar com eles
- Não pode ter caracteres especiais (espaço), exceto o underscore
- Variáveis são escritas em lowercase com separação por underscore

Entrada de Dados

Utilizamos a função nativa do Python **input()** para receber os dados digitados pelo usuário

```
In [17]: input()
```

```
alguma coisa
```

```
Out[17]: 'alguma coisa'
```

```
In [18]: nome = input('Qual o seu nome?')  
print('Prazer em te conhecer', nome)
```

```
Qual o seu nome?Dalton  
Prazer em te conhecer Dalton
```

```
In [20]: print(nome, 'seja bem-vindo ao curso de Python do Programador Avent  
ureiro!')
```

```
Dalton seja bem-vindo ao curso de Python do Programador Aventureir  
o!
```

Projetos Práticos

1. Calculando o IMC do usuário

- Utilizaremos a função **input()** para receber a altura (em metros) e peso (em kg) do usuário
- Converteremos esses valores para um número de ponto flutuante utilizando a função de conversão **float()**
- Faremos o cálculo e mostraremos pro usuário o seu IMC

$$\text{IMC} = \frac{\text{Peso}}{\text{Altura} \times \text{Altura}}$$


```
In [21]: peso = input('Qual o seu peso (em kg)?')
```

```
Qual o seu peso (em kg)?72
```

```
In [22]: altura = input('Qual sua altura (em metros)?')
```

```
Qual sua altura (em metros)?1.72
```

Todo valor de um **input()** é uma string, para poder fazer operações matemáticas precisamos primeiro converter os valores para um número

```
In [23]: imc = peso / (altura * altura)
```

```
-----  
-----  
TypeError                                Traceback (most recent c  
all last)  
<ipython-input-23-42d7e23d0f4e> in <module>()  
----> 1 imc = peso / (altura * altura)  
  
TypeError: can't multiply sequence by non-int of type 'str'
```

Funções de conversão:

- **int()**: Converte um valor para um número inteiro
- **float()**: Converte um valor para um número de ponto flutuante
- **str()**: Converte um valor para uma string
- **bool()**: Converte um valor para um booleano

```
In [24]: # Conversões de valores para um número inteiro  
int() # devolve 0  
int(2.71) # trunca a parte decimal  
int(104.768982321)  
int('4') # string para inteiro
```

```
Out[24]: 4
```

```
In [25]: # Conversões de valores para um número de ponto flutuante  
float() # devolve 0.0  
float('4.73')  
float('inf') # infinito  
float(100) # devolve 100.0
```

```
Out[25]: 100.0
```

```
In [26]: # Conversões de valores para uma string
str() # ''
str(3) # inteiro
str(7.81) # float
str(True) # booleano
```

```
Out[26]: 'True'
```

Voltando ao exemplo da calculadora de IMC podemos agora fazer os cálculos convertendo as variáveis de **peso** e **altura** para float.

```
In [27]: imc = float(peso) / float(altura)**2
```

```
In [28]: print('Seu IMC é', imc)
```

```
Seu IMC é 24.337479718766904
```

2. Média dos alunos

Faremos um programa que recebe as notas de português, matemática, inglês e história de um aluno e diz qual foi a sua média.

- Criaremos 4 variáveis, cada uma irá armazenar a nota de uma matéria
- Converteremos essa nota para **float** para podermos fazer cálculos com elas
- Criaremos a variável **media** que irá armazenar a média das notas digitadas pelo usuário

```
In [29]: nota1 = float(input('Qual sua nota em português?'))
```

```
Qual sua nota em português?7
```

```
In [30]: nota2 = float(input('Qual sua nota em matemática?'))
```

```
Qual sua nota em matemática?5.2
```

```
In [31]: nota3 = float(input('Qual sua nota em inglês?'))
```

```
Qual sua nota em inglês?8.5
```

```
In [32]: nota4 = float(input('Qual sua nota em história?'))
```

```
Qual sua nota em história?10
```

```
In [33]: media = (nota1 + nota2 + nota3 + nota4) / 4
```

```
In [34]: print('Sua média é', media)
```

```
Sua média é 7.675
```

Operadores Artiméticos

São os operadores usados para realizar cálculos matemáticos no Python

Operador	Conceito	Exemplo
+ (Adição ou sinal positivo)	Realiza a soma entre operandos	10 + 4
- (Subtração ou sinal negativo)	Realiza a subtração entre operandos	12 - 8
* (Multiplicação)	Realiza a multiplicação entre operandos	3 * 4
/ (Divisão)	Realiza a divisão entre operandos	100 / 5
// (Divisão inteira)	Realiza a divisão inteira entre operandos (descarta a parte decimal)	10 // 6
% (Módulo)	Retorna o resto da divisão entre operandos	4 % 2
** (Exponenciação)	Retorna um número elevado a potência de outro	4 ** 2

Vamos definir duas variáveis e verificar o resultados das operações da tabela acima:

```
In [35]: numero_1 = 5  
numero_2 = 2
```

```
In [36]: # Soma  
numero_1 + numero_2
```

Out[36]: 7

```
In [37]: # Subtração  
numero_1 - numero_2
```

Out[37]: 3

```
In [38]: # Multiplicação  
numero_1 * numero_2
```

Out[38]: 10

```
In [39]: # Divisão (a divisão sempre retorna um float)  
numero_1 / numero_2
```

Out[39]: 2.5

```
In [40]: # Divisão inteira (descarta a parte decimal ficando apenas com o va  
lor inteiro)  
numero_1 // numero_2
```

Out[40]: 2


```
In [41]: # Módulo (retorna o resto da divisão)
numero_1 % numero_2
```

Out[41]: 1

```
In [43]: # Exponenciação (retorna um valor elevado a potência de outro)
numero_1 ** numero_2
```

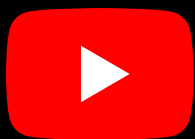
Out[43]: 25

Exercícios

1. Escreva um programa que leia o ano de nascimento do usuário e retorne quantos anos ele fará em 2035
2. Escreva um programa que leia um número digitado pelo usuário e retorne o seu antecessor e o seu sucessor. Vamos considerar como antecessor o número inteiro imediatamente inferior ao número digitado e como sucessor o inteiro imediatamente superior ao número digitado.
3. Faça um programa que leia a largura e a altura de uma parede em metros, calcule a sua área e a quantidade de tinta necessária para pintá-la, sabendo que cada litro de tinta pinta uma área de 2 metros quadrados.

SIGA-NOS NAS REDES SOCIAIS

Clique no ícone para ser redirecionado:



YouTube

Aqui eu compartilho vídeos com muito conteúdo para ajudar você a conquistar seus objetivos o mais rápido possível.



Blog

Este é o portal de conteúdo, você vai encontrar diversos artigos passo a passo com estratégias para seu desenvolvimento pessoal na programação sobre diversas linguagens.



Instagram

Aqui eu compartilho imagens dos meus bastidores, do meu dia a dia, pequenos vídeos com dicas etc.



Facebook

Aqui você verá sempre a atualização das postagens do blog, comentários e engajamento dos meus alunos, fotos e dicas rápidas de programação.