

# Introdução a Linguagem de Programação Python

---

## **Instruções:**

- Todas as atividades devem ser implementadas em Python.
- Os alunos deverão implementar as atividades individualmente.
- Todas as orientações necessárias para execução do código (sistemas operacional, arquivos de entrada, etc) devem ser descritas em um arquivo README
- Enviar código em arquivo .pdf e, separadamente, o código completo compactado. Logo, devem ser enviados 2 arquivos no sistema (.pdf e .zip)

**1.** A padaria do Pão & Pão vende seus pães por R\$ 4,60 cada. O pão do dia anterior (pão dormido) tem um desconto de 60%. Implemente um programa que leia o número de pães dormidos que o usuário está comprando e exiba o preço normal do pão, o desconto e o preço total. Todos os valores devem ser exibidos usando duas casas decimais, e as casas decimais em todos os números devem ser alinhadas.

**2.** Uma roleta possui 38 espaços. Desses, 18 são pretos e 18 são vermelhos e 2 são verdes. Os espaços verdes são numerados 0 e 00. Os espaços vermelhos são numerados 1, 3, 5, 7, 9, 12, 14, 16, 18, 19, 21, 23, 25, 27, 30, 32, 34 e 36. Os números inteiros restantes entre 1 e 36 são utilizados para numerar os espaços pretos.

Muitas apostas podem ser feitas na roleta. Vamos considerar apenas o seguinte subconjunto:

- \* Número único (1 a 36, 0 ou 00)
- \* Vermelho x Preto
- \* Ímpar contra par (0 e 00 não são considerados)
- \* 1 a 18 contra 19 a 36

Escreva um programa que simule o giro da roleta usando o método aleatório do Python de gerador de números. Mostre o número que foi selecionado e todas as apostas que devem ser pagas.

Por exemplo, se o número 13 foi selecionado, seu programa deve exibir:

*O resultado da rodada é 13  
Pagar 13  
Pagar Preto  
Pagar ímpar  
Pagar 1 a 18*

Se o resultado da rodada é 0 ou 00, então o programa deve exibir:

*Pagar 0  
ou  
Pagar 00*

**3.** O objetivo desse exercício é retornar o maior valor em uma coleção de inteiros. A coleção de inteiros será gerada aleatoriamente a partir dos números entre 1 e 100, podendo haver números repetidos e nem todos os números entre 1 e 100 precisam estar contidos na coleção.

Crie um programa que comece selecionando um número inteiro aleatório entre 1 e 100. Salve esse número inteiro como o número máximo encontrado até agora. Após o número inteiro inicial ter sido selecionado, gere 99 números inteiros aleatórios adicionais entre 1 e 100. Verifique cada número inteiro conforme é gerado para ver se é maior do que o número máximo encontrado até agora. Se for esse o caso, seu programa deve atualizar o maior número.

Exiba cada inteiro depois de gerá-lo e, sempre que o maior número foi atualizado, exibir atualização na tela.

Depois de exibir 100 inteiros, seu programa deve exibir o valor máximo encontrado, junto com o número de vezes que o valor máximo foi atualizado durante o processo.

Um exemplo de saída do programa é mostrada abaixo:

*30  
74 (atualizado)  
58  
17  
...  
52  
46  
80 (atualizado)  
37  
97 (atualizado)  
...  
O valor máximo encontrado foi 97  
O número máximo de vezes que o maior valor foi atualizado  
foi 3 vezes.*

**4.** A maioria dos anos tem 365 dias. No entanto, o tempo necessário para a Terra orbitar o Sol é na verdade um pouco mais do que isso. Como resultado, um dia extra, 29 de fevereiro, está incluído em alguns anos para corrigir essa diferença. Esses anos são chamados de anos bissextos.

As regras para determinar se um ano é ou não um ano bissexto são as seguintes:

- \* Qualquer ano divisível por 400 é um ano bissexto.

- \* Dos anos restantes, qualquer ano divisível por 100 não é um ano bissexto.

- \* Dos anos restantes, qualquer ano divisível por 4 é um ano bissexto.

- \* Todos os outros anos não são anos bissextos.

Faça uma função que receba um ano e retorne se o ano é bissexto.

**5.** Escreva uma função que determina quantos dias há em um determinado mês. Sua função terá dois parâmetros: o mês como um número inteiro entre 1 e 12 e o ano como um número inteiro de quatro dígitos. Certifique-se de que sua função informe o número correto de dias em fevereiro para os anos bissextos. Utilizar a função implementada no exercício 4.

**6.** Uma data mágica é uma data em que o dia multiplicado pelo mês é igual ao ano de dois dígitos. Por exemplo, 10 de junho de 1960 é uma data mágica porque junho é o sexto mês e 6 vezes 10 é 60, que é igual ao ano de dois dígitos. Escreva uma função que determina se uma data é ou não uma data mágica. Use sua função para criar um programa principal que encontra e exibe todas as datas mágicas do século XX. Utilize a função implementada no exercício 5.

**7.** O Crivo de Eratóstenes é uma técnica desenvolvida há mais de 2.000 anos para encontrar facilmente todos os números primos entre 2 e algum limite. Segue-se uma descrição do algoritmo:

```
Defina uma lista com todos os números de 2 ao limite
```

```
Defina  $p$  igual a 2
```

```
Enquanto  $p < \text{limite}$ , faça
```

```
    Elimine da lista todos os múltiplos de  $p$  (exceto  $p$ )
```

```
    Defina  $p$  igual ao próximo número da lista
```

```
Imprima a lista
```

Crie um programa Python que use esse algoritmo para exibir todos os números primos entre 2 e um limite inserido pelo usuário. Se você implementar o algoritmo corretamente, deverá ser capaz de exibir todos os números primos menores que 1.000.000 em alguns segundos.

**8.** No jogo Scrabble, cada letra possui uma pontuação. A pontuação total da palavra é a soma da pontuação das letras. As letras mais comuns valem menos pontos, as letras menos comuns valem mais pontos. Os pontos associados a cada letra são:

1 ponto	A, E, I, L, N, O, R, S, T, U
2 pontos	D, G
3 pontos	B, C, M, P
4 pontos	F, H, V, W, Y
5 pontos	K
8 pontos	J, X
10 pontos	Q, Z

Escreva um programa que computa e apresenta a pontuação para uma palavra. Crie um dicionário que mapeie as letras e valores. Utilize o dicionário para computar a pontuação.

**9.** Crie um programa Python que leia um ou mais arquivos-fonte Python e identifique funções que não são imediatamente precedidas por um comentário. Para os fins deste exercício, suponha que qualquer linha que comece com “def”, seguida por um espaço, seja o início de uma definição de função. Suponha que o caractere de comentário, “#”, será o primeiro caractere na linha anterior quando a função tiver um comentário.

Imprima os nomes de todas as funções sem comentários, junto com o nome do arquivo e o número da linha onde a definição da função está localizada. O usuário fornecerá os nomes de um ou mais arquivos Python como parâmetros de linha de comando. Se o seu programa encontrar um arquivo que não existe ou não pode ser aberto, ele deve exibir uma mensagem de erro apropriada antes de prosseguir e processar os arquivos restantes (neste caso utilizar conceito de exceção).

**10.** A distância entre duas strings é uma medida de sua similaridade - quanto menor a distância, mais semelhantes as strings são em relação ao número mínimo de operações de inserção, exclusão e substituição necessárias para transformar uma string na outra.

Considere as palavras (strings) *camisa* e *amigas*. A primeira string pode ser transformada na segunda com as seguintes operações: Remova a letra *c*, substitua o *s* por *g*, insira a letra *s* no final da string. Este é o menor número de operações que podem ser realizadas para transformar *camisa* em *amigas*. Como resultado, a distância de edição é 3.

Escreva uma função recursiva que calcule a distância de edição entre duas strings. Use o seguinte algoritmo:

```
Leia duas strings s e t
Se tamanho de s é 0, então
    Retorne o tamanho de t
Senão Se o tamanho t for 0, então
    Retorna o comprimento de s
Senão
    Custo = 0
    Se o último caractere em s não for igual ao último
    caractere em t, então
        Custo = 1
    Defina d1 igual à distância de edição entre todos os
    caracteres, exceto o último em s, e todos os caracteres
    em t, mais 1
    Defina d2 igual à distância de edição entre todos os
    caracteres em s, e todos caracteres exceto o último em t,
    mais 1
    Defina d3 igual à distância de edição entre todos os
    caracteres, exceto o último em s, e todos os caracteres,
    exceto o último em t, mais custo
    Retorne o mínimo de d1, d2 e d3
```

Use sua função recursiva para escrever um programa que lê duas strings do usuário e exibe a distância de edição entre elas.