

Módulo 6 - Listas

Aula 6.4 - Compreensão - Exercícios

Instruções para submissão: Para cada questão, crie no VSCode um arquivo chamado `aula4_questaoX.py`, sendo `X` o número da questão. Faça commit de todos os arquivos para a pasta `modulo6` do seu repositório da disciplina no GitHub.

1) Escreva um script python que use compreensão de listas para criar as seguintes listas:

- Todos os números pares entre 20 e 50, ou seja, [20, 22, 24, ..., 48, 50]
- Os quadrados de todos os valores da lista [1, 2, 3, 4, 5, 6, 7, 8, 9]
- Todos os números entre 1 e 100 que sejam divisíveis por 7
- Para todos os valores em `range(0, 30, 3)`, escreva "par" ou "ímpar" dependendo da paridade do elemento (ex: ['par', 'ímpar', ... , 'ímpar'])

2) Solicite uma frase do usuário e usando compreensão de listas imprima:

- A lista de vogais da frase, ordenada alfabeticamente
- A lista de consoantes da frase (remova espaços em branco)

```
Digite uma frase: Eu gosto de programar em Python
Vogais: ['a', 'a', 'e', 'e', 'o', 'o', 'o', 'o', 'u']
Consoantes: ['E', 'g', 's', 't', 'd', 'p', 'r', 'g', 'r', 'm',
'r', 'm', 'P', 'y', 't', 'h', 'n']
```

3) Reescreva o código a seguir para construir a lista `pagamentos` usando compreensão de listas, ou seja, eliminando o laço de repetição.

```
horas_trabalhadas = [40, 37, 45, 40, 40, 48]
ganho_por_hora = 20
hora_extra = 25
pagamentos = []
for hora in horas_trabalhadas:
    pagamentos.append(ganho_por_hora * min(hora, 40) + \
                      hora_extra * max(0, hora-40))
print(pagamentos)
```

- 4) Reescreva o código a seguir para construir a lista **aprovados** usando compreensão de listas, ou seja, eliminando o laço de repetição.

```
alunos = ["Maria", "Jose", "Carla", "Sol"]
notas = [35, 50, 20, 80]
aprovados = []
for i in range(len(notas)):
    if notas[i] >= 60:
        aprovados.append(alunos[i])
```