

# Unidade 1 - Atividade Objetiva 01

## PERGUNTA 1

Marque os identificadores válidos em Python.

XYZ  
\_\_nome\_aluno

## PERGUNTA 2

Sobre os comentários, marque as alternativas corretas.

Comentários podem utilizar tanto o símbolo cerquilha (#) quanto a combinação de 3 aspas (""").

## PERGUNTA 3

No código a seguir `x = "1978"`, um valor está sendo atribuído à variável `x`. Ao executar o comando `type(x)`, que tipo será retornado?

`str`

## PERGUNTA 4

Marque entre as alternativas abaixo, os tipos de dados corretos para cada informação.

CPF - `str`  
Altura de uma pessoa em centímetros - `int`  
Data de nascimento - `str`

## PERGUNTA 5

Suponha a seguinte declaração de uma lista:

```
lista_aninhada = [ [1,2,3],[4,5,6,7],[8,9],[ [10,11,12], [13, 14, 15] ] ]
```

Que instrução retornará o número 13?

```
lista_aninhada[3][1][0]
```

## PERGUNTA 6

Suponha que seja declarado o seguinte dicionário:

```
dic = { "MG": {"Área": 586522, "População": 21040662, "Capital": "Belo Horizonte"},  
        "SP": {"Área": 248222, "População": 45538936, "Capital": "São Paulo"},  
        "RS": {"Área": 281730, "População": 11329605, "Capital": "XYZ"} }
```

Qual comando devemos utilizar para alterar a capital do Rio Grande do Sul de XYZ para Porto Alegre?

```
dic["RS"]["Capital"] = "Porto Alegre"
```

## PERGUNTA 7

Em relação às estruturas de dados aninhadas (uma estrutura dentro de outra), pode-se afirmar que Python só aceita o aninhamento entre estruturas do mesmo tipo (por exemplo, dentro de uma lista só podemos ter outra lista, dentro de um dicionário só pode haver outro dicionário).

Falso

#### PERGUNTA 8

A função `reduce()` possui esse nome pelo fato de reduzir uma lista com `n` elementos para apenas 1 elemento.

Falso

Pergunta 9A função `reduce()` reduz uma lista a um único valor agregado (por exemplo, a soma de todos os números), mas ela não interfere na ordenação e nem na quantidade de elementos.

#### PERGUNTA 9

Com a função `zip()` é possível combinar elementos de um dicionário com elementos de uma lista.

Exemplo:

```
dic = {"MG": "Minas Gerais", "SP": "São Paulo", "AC": "Acre", "MA": "Maranhão"}
regioes = ["SE", "SE", "NO", "NE"]
estados = list(zip(dic, dic.values(), regioes))
```

Verdadeiro

#### PERGUNTA 10

A função `filter(lambda x: x%3 == 0, numeros)`, aplicada à lista `numeros = [n for n in range(1, 16)]`, retornará uma lista apenas com os números não divisíveis por 3, no intervalo fechado entre 1 e 15, uma vez que ela elimina os elementos para os quais a condição `x%3 == 0` será avaliada como `True`.

Verdadeiro

#### PERGUNTA 11

Comprehension em Python é uma forma concisa e sofisticada de gerar listas, onde podemos gerar uma lista iterando por outra lista ou por uma faixa de valores e ainda testar se cada elemento satisfaz uma determinada condição.

O exemplo abaixo vai gerar uma lista dos anos bissextos entre os anos 1900 e 2020. A lista gerada está correta?

```
bissextos = [ano for ano in range(1900, 2021, 4) if (ano % 400 == 0) or (ano % 4 == 0 and ano % 100 != 0)]
```

Verdadeiro

O exemplo gera todos os anos múltiplos de 4 no intervalo entre 1900 e 2020, e testa de cada um desses anos gerados é bissexto, verificando de ele é divisível por 400 ou divisível por 4 mas não por 100.

#### PERGUNTA 12

Marque as opções que geram corretamente uma lista com os anos bissextos entre 1900 e 2020.

- `bissextos = list(filter(lambda ano: (ano % 400 == 0) or (ano % 4 == 0 and ano % 100 != 0), range(1900, 2021, 4)))`
- `bissextos = [ano for ano in range(1900, 2021, 4) if (ano % 400 == 0) or (ano % 4 == 0 and ano % 100 != 0)]`