**Possíveis Erros em Python**

**Erros referentes a condicional:**

* Dois pontos  
  Em instruções compostas (*if, while, for*, etc) não use mais os parênteses (*brackets*), em python você deve usar dois pontos – :
* Usando o operador de atribuição em vez do operador de igualdade: Ao realizar uma comparação entre dois objetos ou valor, você apenas usa o operador de igualdade (==), e não o operador de atribuição (=) . O operador de atribuição coloca um objeto ou valor dentro de uma variável e não compara nada.
* Usando o operador lógico incorreto: A maioria dos operadores não apresenta desenvolvedores com problemas, mas os operadores lógicos fazem. Lembre-se de usar e determinar quando ambos os operandos devem ser True e quando qualquer um dos operandos pode ser True.

**Erros referentes a repetição:**

* Criando erros conta-por-um nos loops: Lembre-se de que um loop não conta o último número que você especifica em um intervalo. Então, se você especificar o intervalo [1: 11], você realmente obtém saída para valores entre 1 e 10.

**Erros referentes as funções:**

* Colocando chamadas de função na ordem errada ao criar instruções complexas: O Python sempre executa funções da esquerda para a direita. Então, a declaração MyString. faixa(). center (21, "\*") produz um resultado diferente do MyString. centro (21, "\*"). faixa(). Quando você encontra uma situação em que a saída de uma série de funções concatenadas é diferente do que você esperava, você precisa verificar a ordem das funções para garantir que cada função esteja no lugar correto.
* Atribuições com tipos mutáveis e imutáveis  
  Estude e tenha certeza de que realmente aprendeu o comportamento dos tipos de dados em python – *listas, dicionário*s*, strings e números*. É muito comum, por exemplo, fazer uma atribuição como: *lista = lista.append(alguma\_coisa)*. Com esta atribuição a única operação será atribuir *None* a *lista* e com isso perder completamente a referência para a sua lista de verdade.