Laços

* Um laço é, assim como uma condição, uma estrutura de desvio de fluxo de execução.
* Entretanto, um laço realiza desvio de fluxo para um ponto anterior do código, fazendo com que alguns comandos já executados **sejam executados novamente**.
* Por isso, laços são chamados de **estruturas de repetição**.
* Python possui dois tipos de laço:
  + For
  + While

For

* Sintaxe:

*for <variável> in <conjunto>:*

*bloco\_de\_codigo*

* A instrução **for** realiza uma quantidade de repetições associada à quantidade de elementos em um conjunto fornecido.
* Existe uma variável associada à execução do **for**.
* A cada repetição do bloco de código, essa variável assume um dos elementos do conjunto.
* Dessa forma, sabe-se a quantidade de vezes que o bloco de código será repetido.
* Que elementos podem fazer o papel de conjunto?
  + Strings - os caracteres são os elementos;
  + Listas - estrutura a ser estudada mais adiante;
  + **range - faixas numéricas**.

Range

* Sintaxe:

*range(start, stop, step)*

* Parâmetros:
  + start: primeiro valor que a variável recebe.
  + stop: último valor que a variável recebe e que dispara o fim das repetições.
    - **O BLOCO DE CÓDIGO NÃO CHEGA A SER EXECUTADO COM ESTE VALOR**.
  + step: valor acrescentado à variável a cada iteração do laço
* Exemplos:
  + range(5,10,1) = {5,6,7,8,9}
  + range(10,20,2) = {10,12,14,16,18}
  + range(0,15,3) = {0,3,6,9,12}
  + range(10,4,-1) = {10,9,8,7,6,5}
* Os parâmetros **start** e **step** são opcionais.
  + Se o range tiver somente dois parâmetros, são o **start** e o **stop**.
    - Neste caso, o **step** será 1.
  + Se o range tiver somente um parâmetro, é o **stop**.
    - Neste caso, o **start** = 0 e o **step** = 1.
* range(0,6) = range(0,6,1) = {0,1,2,3,4,5}
* range(7) = range(0,7,1) = {0,1,2,3,4,5,6}