Computação I - Python
Estrutura de repetição iteradora: *for*

Apresentação

Apresentado por: Carla A. D. M. Delgado

Produção: Instituto de Computação - UFRJ

Metodologia de referência: https://doi.org/10.5753/wei.2016.9683

Estrutura de repetição: for

- usado em situações onde o número de repetições é sabido antes do laço ser iniciado
- o laço **for** existe em várias linguagens de programação!
- em Python, o laço **for** opera sobre um dado iterável
  - Passagem por cada elemento do iterável

```
for <variavel> in <indexavel>:
    <sequência de comandos>
```

Estrutura de repetição: for

- Iterando sobre tuplas

```
1 def filtra_pares(t):
2     ''' retorna uma tupla com os
3     inteiros pares da tupla original
4     tuple --> tuple '''
5     pares = ()
6     for proximo in t:
7         if proximo%2 == 0:
8             pares = pares + (proximo,)
9     return pares
```

Estrutura de repetição: for

- Iterando sobre tuplas

```
1 def filtra_pares(t):
2     ''' retorna uma tupla com os
3     inteiros pares da tupla original
4     tuple --> tuple '''
5     pares = ()
6     for proximo in t:
7         if proximo%2 == 0:
8             pares = pares + (proximo,)
9     return pares
```

- no início de cada iteração do laço **for**, um elemento da tupla *t* é atribuído à variável *proximo*
- a execução do laço **for** termina quando o laço tiver sido executado uma vez para cada elemento de *t*

Estrutura de repetição: for

- Comparando com o laço **while**

```
1  def soma_paresV1(numeros):
2      ''' soma os numeros pares que estao na lista de entrada
3      list --> int '''
4      i=0
5      soma=0
6      while i<len(numeros):
7          if numeros[i]%2==0:
8              soma = soma + numeros[i]
9          i=i+1
10     return soma
```

Estrutura de repetição: for

- Comparando com o laço **while**

```
1  def soma_paresV1(numeros):
2      ''' soma os numeros pares que estao na lista de entrada
3      list --> int '''
4      i=0
5      soma=0
6      while i<len(numeros):
7          if numeros[i]%2==0:
8              soma = soma + numeros[i]
9          i=i+1
10     return soma
```

- Iterando sobre listas

```
1  def soma_paresV2(numeros):
2      ''' soma os numeros pares que estao na lista de entrada
3      list --> int '''
4      soma = 0
5      for num in numeros:
6          if num%2==0:
7              soma = soma + num
8      return soma
```

Estrutura de repetição: for

- Iterando sobre strings

```
1 def todasasvogais(texto):
2     '''retorna uma string com as vogais que apareceram em um texto,
3     na mesma sequencia que apareceram
4     str->str'''
5     vogais=""
6     for letra in texto:
7         if letra in 'AEIOUaeiou':
8             vogais=vogais+letra
9     return vogais
```

Estrutura de repetição: for

- E como funciona com dicionários?

Estrutura de repetição: for

- E como funciona com dicionários?

```
1 afinidades = {
2     'Leo': ['Sofia', 'Andrea'],
3     'Marcos': ['Andrea'],
4     'Sofia': ['Leo'],
5     'Alex': ['Andrea', 'Marcos'],
6     'Andrea': ['Marcos']
7 }
```

Estrutura de repetição: for

- E como funciona com dicionários?

```
afinidades = {
    'Leo': ['Sofia', 'Andrea'],
    'Marcos': ['Andrea'],
    'Sofia': ['Leo'],
    'Alex': ['Andrea', 'Marcos'],
    'Andrea': ['Marcos']
}
```

```
def quem_curtiu(afinidades, fulano):
    '''retorna a lista de pessoas que curtiram fulano
    de acordo com as afinidades mapeadas no dicion rio
    dici, str --> list'''

    curtiram=[]
    for nome in afinidades:
        if fulano in afinidades[nome]:
            curtiram=curtiram+[nome]
    return curtiram
```

Computação I - Python
Estrutura de repetição iteradora: *for*

Apresentação

Apresentado por: Carla A. D. M. Delgado

Produção: Instituto de Computação - UFRJ

Metodologia de referência: `https://doi.org/10.5753/wei.2016.9683`