PYTHON – Compreensão de Listas

Introdução à Programação

SI2

Inicializando listas

- Não é possível atribuir a uma posição inexistente de uma lista
- >>> vetor = []
- >>> vetor [0] = 1
- Traceback (most recent call last):
- IndexError: list assignment index out of range
- Se uma lista vai ser usada como vetor, é conveniente iniciá-la
- >>> vetor = [0]*10
- >>> vetor [0] = 3
- >>> vetor
- [3, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Usando None

- No uso de estruturas de dados, às vezes é importante preencher uma posição com um valor “não válido˜.
- A melhor opção para esse uso é emprega o valor especial **None**
  - Não faz parte de tipo nenhum
  - É melhor que usar 0, [] ou uma string vazia
- Útil para criar uma lista “vazia” mas com um número conhecido de posições. Ex.:
- >>> lista = [None]*5
- >>> lista
- [None, None, None, None, None]

Criando Listas

- Crie uma lista com inteiros de 0 a 9 elevados ao quadrado

```
squares = []
for x in range(10):
  squares.append(x**2)
print squares
```

[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]

Criando Listas

- Crie uma nova lista com o tamanho de cada nome que pertence a uma dada lista. Como exemplo, use:
  `nomes = ['Anne', 'Amy', 'Bob', 'David', 'Carrie', 'Barbara', 'Zach']`

```
names = ['Anne', 'Amy', 'Bob', 'David', 'Carrie',
'Barbara', 'Zach']
lengths = []
for name in names:
   lengths.append(len(name))
print lengths
[4, 3, 3, 5, 6, 7, 4]
```

Criando Listas

- Crie uma lista com todos os possíveis pares de bebida e comida das listas `['água', 'chá', 'suco']` e `['presunto', 'ovos', 'queijo']`, respectivamente

```
possible_choices = []
for drink in ['water', 'tea', 'juice']:
   for food in ['ham', 'eggs', 'spam']:
      possible_choices.append([drink,food])
print possible_choices
```

```
[['water', 'ham'], ['water', 'eggs'], ['water', 'spam'], ['tea', 'ham'],
['tea', 'eggs'], ['tea', 'spam'], ['juice', 'ham'], ['juice', 'eggs'],
['juice', 'spam']]
```

Criando Listas

- Crie uma lista de coordenadas em um grid retangular, no formato (x,y), onde 0 < x < 10 e 0 < y < 15

```
coords = []
for x in range(5):
    for y in range(3):
        coordinate = (x, y)
        coords.append(coordinate)
print coords
```

```
[(0, 0), (0, 1), (0, 2), (1, 0), (1, 1), (1, 2), (2, 0),
(2, 1), (2, 2), (3, 0), (3, 1), (3, 2), (4, 0), (4, 1),
(4, 2)]
```

Filtrando Listas

- A partir de uma lista existente de nomes, crie uma nova lista contendo apenas os nomes que começam com a letra B:

```
names = ['Anne', 'Amy', 'Bob', 'David', 'Carrie',
'Barbara', 'Zach']
b_names = []
for name in names:
   if name.startswith('B'):
      b_names.append(name)
print b_names
['Bob', 'Barbara']
```

Filtrando Listas

- Crie uma lista com inteiros de 0 a 9, cujo quadrado é maior que 5 e menor que 50

```
special_squares = []
for x in range(10):
   square = x**2
   if square > 5 and square < 50:
      special_squares.append(square)
print special_squares
```

[9, 16, 25, 36, 49]

Daria pra fazer em uma linha só?

Criando Listas

```
[ x**2 for x in range(10) ]

[ len(name) for name in names ]

[ [drink,food] for drink in ['water', 'tea',
'juice'] for food in ['ham', 'eggs', 'spam'] ]

[ (x,y) for x in range(5) for y in range(3) ]
```

Compreensão de listas

- Funcionalidade muito poderosa da linguagem Python
  - Gera uma lista nova aplicando uma função para cada elemento da lista original.
  - Muito usado por programadores Python! (Economia de código!)
- A sintaxe da compreensão de lista usa-se de palavra-chaves:
  - [expression for name in list]

```
>>>s = [ x**2 for x in range(10) ]
>>> m = [len(x) for x in palavras]
```

python

Compreensão de listas

- Permite também o uso de filtros (determinam se uma determinada expressão deve ser executada sobre um membro da lista)
    - [expression for name in list if filter]

```
>>> x = [x**2 for x in s if x%2 == 0]
>>> m = [i for i in p if i>5]
```

- Você também pode aninhar compreensão de listas!
    - [expression for name in [expression for name in list]]

python

Filtrando Listas

```
[ name for name in names if name.startswith('B') ]
[ x**2 for x in range(10) if x**2 > 5 and x**2 < 50 ]

squares = [ x**2 for x in range(10) ]
special_squares = [ s for s in squares if s > 5 and s < 50 ]

[ s for s in [ x**2 for x in range(10) ] if s > 5 and s < 50 ]
```

Matrizes

- Listas podem ser usadas para guardar matrizes
- Por exemplo, podemos criar uma matriz identidade de 3x3 com o código

```
m = []
for i in range(3):
    m.append([0]*3)
    m[i][i] = 1
```

- Obs: Não é uma boa idéia iniciar uma matriz assim:

```
m = [[0]*3]*3
for i in range(3): m[i][i] = 1
print m
```

  - Resultado: [[1, 1, 1], [1, 1, 1], [1, 1, 1]]

EXERCÍCIOS

Exercícios

1. Ler uma lista de 5 números inteiros e mostre cada número juntamente com a sua posição na lista.
2. Ler uma lista de 10 números reais e mostre-os na ordem inversa.
3. Ler uma lista com 4 notas, em seguida o programa deve exibir as notas e a média.
4. Ler uma lista com 20 idades e exibir a maior e menor.

Exercícios

5. Inicialize uma lista de 20 números inteiros. Usando compreensão de listas, armazene os números pares em uma lista PAR e os números ímpares em uma lista IMPAR. Imprima as listas PAR e IMPAR.

6. Faça um programa que receba a temperatura média de cada mês do ano e armazene-as em uma lista. Em seguida, calcule a média anual das temperaturas e mostre a média calculada juntamente com todas as temperaturas acima da média anual, e em que mês elas ocorreram (mostrar o mês por extenso: 1 – Janeiro, 2 – Fevereiro, . . .).

Exercícios

7. Faça um programa que leia uma lista L com valores informados pelo usuário e gere uma nova lista cujos valores consistem do dobro dos valores lidos.

8. Faça um programa que crie uma matriz aleatoriamente. O tamanho da matriz deve ser informado pelo usuário.

Exercícios

9. Faça um programa que leia um número indeterminado de notas. Após esta entrada de dados, faça o seguinte:
   - Mostre a quantidade de notas que foram lidas.
   - Exiba todas as notas na ordem em que foram informadas.
   - Exiba todas as notas na ordem inversa à que foram informadas, uma abaixo do outra.
   - Calcule e mostre a soma das notas.
   - Calcule e mostre a média das notas.
   - Calcule e mostre a quantidade de notas acima da média calculada.

Exercícios

10. Utilizando listas faça um programa que faça 5 perguntas para uma pessoa sobre um crime. As perguntas são:
    - "Telefonou para a vítima?"
    - "Esteve no local do crime?"
    - "Mora perto da vítima?"
    - “Tinha dívidas com a vítima?"
    - "Já trabalhou com a vítima?“

    O programa deve no final emitir uma classificação sobre a participação da pessoa no crime. Se a pessoa responder positivamente a 2 questões ela deve ser classificada como "Suspeita“; entre 3 e 4 como "Cúmplice" e; 5 como "Assassino". Caso contrário, ele será classificado como "Inocente".

Exercícios

11. Uma empresa de pesquisas precisa tabular os resultados da seguinte enquete feita a um grande quantidade de organizações: "Qual o melhor Sistema Operacional para uso em servidores?" As possíveis respostas são:
    - 1- Windows XP 2- Unix 3- Linux 4- Netware 5- Mac OS 6- Outro
- Você deve desenvolver um programa em Python que leia as respostas da enquete e informe ao final o resultado da mesma. O programa deverá ler os valores até ser informado o valor 0 (zero), que encerra a entrada dos dados. Não deverão ser aceitos valores além dos válidos para o programa (0 a 6).
- Os valores referentes a cada uma das opções devem ser armazenados em uma lista. Após os dados terem sido completamente informados, o programa deverá calcular a percentual de cada uma das respostas e informar o vencedor da enquete.
  continua no próximo slide…

Exercícios

12. (Continuação) O formato da saída foi dado pela empresa, e é o seguinte:
    - Sistemas Operacionais - Votos %
        - Windows XP 1500 17%
        - Unix 3500 40%
        - Linux 3000 34%
        - Netware 500 5%
        - Mac OS 150 2%
        - Outro 150 2%
    - Total de 8800 votos
    - O Sistema Operacional mais votado foi o Unix, com 3500 votos, correspondendo a 40% dos votos.

Bibliografia

- Livro “Como pensar como um Cientista de Computação usando Python” – Capítulo 8
  - http://pensarpython.incubadora.fapesp.br/portal
- Python Tutorial
  - http://www.python.org/doc/current/tut/tut.html
- Dive into Python
  - http://www.diveintopython.org/
- Python Brasil
  - http://www.pythonbrasil.com.br/moin.cgi/DocumentacaoPython#head5a7ba2746c5191e7703830e02d0f5328346bcaac