08-Listas

September 23, 2018

```
In [4]: from IPython.display import Image
```

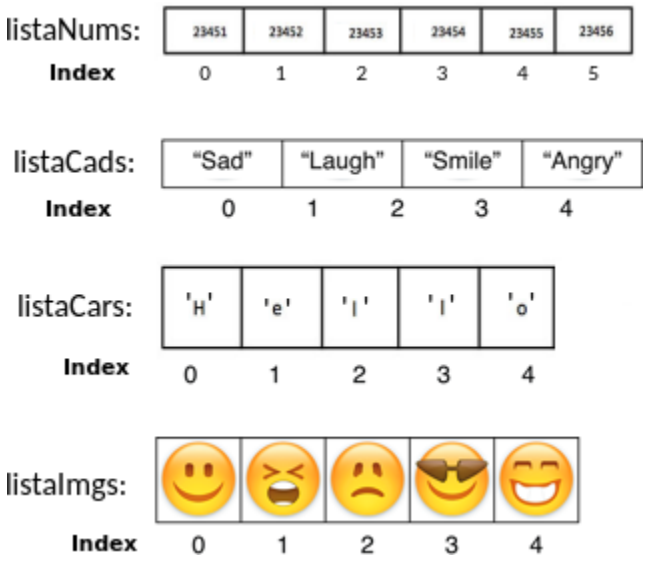
1 Listas

- Estructuras que agrupan uno o más objetos.
- La estructura mas importante y más usada
- Símbolo []
- Tipo **list**
- Son mutables, es decir puede modificarlas (adicionar, eliminar, cambiar elementos)
- Generalmente, contienen un solo tipo de elemento (int, float, string, ...)
- Pero pueden contener cualquier tipo de objeto

1.1 Ejemplos

```
In [6]: Image(filename='images/ejemplosListas.png')
```

Out[6]:



1.2 Creación de listas

```
In [ ]: lsVacia = [] ## Lista vacia
        lsNumeros = [11, 22, 55, 33] ## Lista con elementos de tipo entero
        lsSecuencias  = ["acgtt","ttgaataa", "gagattataggaa", "agatat"]
        lsNombres = ["Maria", "Jose", "Janeth", "Mary"]
```

1.3 Acceso

- A través del operador [] y la posición (indice), así [indice].
- A través del operador de rango: [inicio:final]

1.3.1 Ejemplo

```
In [9]: ls = [1, 2, 3, 11, 55, 99, 0.5, 'a', 'b']

In [75]: ls [4] ## Acceso a la posición 4

Out[75]: 55

In [76]: ls [4: 8] ## Rango de elementos

Out[76]: [55, 99, 0.5, 'a']
```

1.4 Asignación

```
In [10]: ls = [1, 2, 3, 11, 55, 99, 0.5, 'a', 'b']
         print (ls)

[1, 2, 3, 11, 55, 99, 0.5, 'a', 'b']


In [78]: ls [2] = 333
         print ls

[1, 2, 333, 11, 55, 99, 0.5, 'a', 'b']


In [80]: ls [3] = "aaaa"
         print ls

[1, 2, 333, 'aaaa', 55, 99, 0.5, 'a', 'b']


In [81]: a = ls [2]
         b = ls [5]
         print a, b
         print a + b

333 99
432
```

1.5 Funciones Integradas de Listas

```
In [ ]: lista.append (elemento) ## Adición de elemento
        lista.count (elemento)  ## Conteo de un elemento
        lista.extend (otraLista) ## Pegar otra lista
        lista.index (elemento) : posición ## Retorna posición de un elemento
        lista.insert (elemento, posición) ## Inserta un elemento en una posición
        lista.pop (posición) : elemento   ## Elimina el elemento en la posición y retorna el e
        lista.remove (elemento) ## Remueve la primera aparición del elemento
        lista.reverse () # Invierte la lista
        lista.sort ()  # Ordena la lista ascendentemente (reverse=True, para descendente)
```

1.6 Ejemplos

```
In [13]: ls = [1, 2, 3, 11, 55, 99, 0.5, 'a', 'b']
         print (ls)

[1, 2, 3, 11, 55, 99, 0.5, 'a', 'b']


In [14]: help (ls.append)

Help on built-in function append:

append(...) method of builtins.list instance
    L.append(object) -> None -- append object to end



In [19]: ls.append (5555)

In [18]: ls.remove ('a')


        ---------------------------------------------------------------------------

        ValueError                                Traceback (most recent call last)

        <ipython-input-18-6bee0de2843f> in <module>()
    ----> 1 ls.remove ('a')


        ValueError: list.remove(x): x not in list


In [20]: ls.remove ('b')
         ls

Out[20]: [1, 2, 3, 11, 55, 99, 0.5, 5555, 5555]
```

```
In [22]: ls.reverse()
         print (ls)

[1, 2, 3, 11, 55, 99, 0.5, 5555, 5555]


In [23]: ls.sort()
         print (ls)

[0.5, 1, 2, 3, 11, 55, 99, 5555, 5555]


In [25]: ls.sort (reverse=True)
         print(ls)

[5555, 5555, 99, 55, 11, 3, 2, 1, 0.5]
```