

✓ LISTAS

1. Un nuevo tipo de dato
2. Es mutable
3. Tiene Indexación: hay un orden
4. Hay un número finito de posiciones
5. Todas las listas abren y cierran con [] y se separan con ,

Sintaxis

```
nombre_lista=[] #Crear lista vacía  
flores=["margarita", "rosa", "tulipan"]
```

```
1 nu=[]  
2 for i in range(1,11):  
3     nu[i]=i
```



```
-----  
-----  
IndexError                                Traceback (most recent  
call last)  
/tmp/ipython-input-875959369.py in <cell line: 0>()  
      1 nu=[]  
      2 for i in range(1,11):  
----> 3     nu[i]=i  
  
IndexError: list assignment index out of range
```

Próximos pasos: [Explicar error](#)

```
1 nu.clear()
2 for i in range(10):
3     nu.append(i+1)
```

```
1 nu
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
1 nombres=[]
2 for i in range(3):
3     nombres.append(input("Ingrese un nombre: "))
```

```
Ingrese un nombre: Carlos
Ingrese un nombre: Manuel
Ingrese un nombre: Harry
```

```
1 nombres
```

```
['Carlos ', 'Manuel ', 'Harry']
```

✓ Indexación en listas

*Recordar que python cuenta desde **0**

```
Lista=[Elemento1, Elemento2, Elemento3]
Lista[0]=Elemento1
Lista[3]=error fuera de rango
```

```
1 #Indexación negativa
2 Lista_num=[1,2,3,4,5,6,7,8,9,10]
3 Lista_num[-1]
```

```
10
```

```
1 Lista_num[-2]
```

```
9
```

```
1 Lista_num[len(Lista_num)-1]
```

```
10
```

```
1 Lista_num[::]
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
1 Lista_num[:2]
```

```
[1, 2]
```

✓ Listas por compresión

Son expresiones que generan listas Sintaxis:

```
nombre_lista=[expresión for variable in iteración if condición]
```

Ejemplos

```
1 #1. Convertir una lista de oraciones en una lista de palabras
2 oracion=["Aquí va una oración", "Aquí va otra oración", "Aquí
3 len(oracion)
```

```
3
```

```
1 palabras=[i.split() for i in oracion]
2 palabras
```

```
[['Aquí', 'va', 'una', 'oración'],
 ['Aquí', 'va', 'otra', 'oración'],
 ['Aquí', 'va', 'la', 'tercera', 'oracion']]
```

```
1 len(palabras[2])
```

```
5
```

```
1 #2. Obtener los pares de Lista_num
2 pares=[x for x in Lista_num if x%2==0]
3 pares
```

```
[2, 4, 6, 8, 10]
```

```
1 #3. Clasificar números en po
2 numeros=[-1,2,-3,4,-5,6,-7,8
3 cacion=[("positivo",x) if x>=
4 cacion
```

```
[('negativo', -1),
 ('positivo', 2),
 ('negativo', -3),
 ('positivo', 4),
 ('negativo', -5),
 ('positivo', 6),
 ('negativo', -7),
 ('positivo', 8),
 ('negativo', -9),
 ('positivo', 10)]
```

