

Funciones Incorporadas Basicas solo para listas

A	B	C	D
abs()	bin()	callable()	delattr()
aiter()	bool()	chr()	dict()
all()	breakpoint()	classmethod()	dir()
anext()	bytearray()	compile()	divmod()
any()	bytes()	complex()	
ascii()			

E	F	G	H
enumerate()	filter()	getattr()	hasattr()
eval()	float()	globals()	hash()
exec()	format()		help()
	frozenset()		hex()

I	L	M	N
id()	len()	map()	next()
input()	list()	max()	
int()	locals()	memoryview()	
isinstance()		min()	
issubclass()			

O	P	R	S
object()	pow()	range()	set()
oct()	print()	repr()	setattr()
open()	property()	reversed()	slice()
ord()		round()	sorted()
			staticmethod()
			str()
			sum()
			super()

--	--	--	--

T	V	Z	—
tuple()	vars()	zip()	import()
type()			

Claro, aquí tienes una lista de las funciones de Python que te pueden ser útiles específicamente para trabajar con listas:

- **len()** : Retorna la longitud de una lista.

```
lista = [1, 2, 3]
print(len(lista)) # Salida: 3
```

- **list()** : Convierte un iterable en una lista.

```
iterable = 'abc'
print(list(iterable)) # Salida: ['a', 'b', 'c']
```

- **sorted()** : Retorna una lista ordenada de los elementos de un iterable.

```
desordenada = [3, 1, 2]
print(sorted(desordenada)) # Salida: [1, 2, 3]
```

- **sum()** : Retorna la suma de todos los elementos de una lista.

```
numeros = [1, 2, 3]
print(sum(numeros)) # Salida: 6
```

- **max()** : Retorna el elemento más grande de una lista.

```
numeros = [1, 2, 3]
print(max(numeros)) # Salida: 3
```

- **min()** : Retorna el elemento más pequeño de una lista.

```
numeros = [1, 2, 3]
print(min(numeros)) # Salida: 1
```

- **map()** : Aplica una función a todos los elementos de una lista.

```
numeros = [1, 2, 3]
cuadrados = map(lambda x: x**2, numeros)
print(list(cuadrados)) # Salida: [1, 4, 9]
```

- **filter()** : Filtra los elementos de una lista que no cumplen con una condición.

```
numeros = [1, 2, 3, 4, 5]
pares = filter(lambda x: x % 2 == 0, numeros)
print(list(pares)) # Salida: [2, 4]
```

- **slice()** : Crea un objeto slice que se puede usar para seleccionar partes de una lista.

```
lista = [0, 1, 2, 3, 4]
corte = slice(1, 3)
```

```
print(lista[corte]) # Salida: [1, 2]
```

- **enumerate()** : Agrega un contador a una lista y la retorna como un objeto enumerado.

```
for indice, valor in enumerate(['a', 'b', 'c']):  
    print(indice, valor) # Salida: 0 a, 1 b, 2 c
```

- **zip()** : Combina los elementos de dos o más listas en una lista de tuplas.

```
a = [1, 2, 3]  
b = ['a', 'b', 'c']  
for i in zip(a, b):  
    print(i) # Salida: (1, 'a') (2, 'b') (3, 'c')
```

- **reversed()** : Retorna un iterador que recorre los elementos de una lista en orden inverso.

```
for i in reversed([1, 2, 3]):  
    print(i) # Salida: 3 2 1
```

Estas funciones son muy útiles para manipular y trabajar con listas en Python. Si necesitas más información o ejemplos, ¡no dudes en preguntar!