

Ejemplo - Iterar lista por sus elementos

Al iterar directamente una lista en un ciclo `for`, el iterador `i` toma el valor del **elemento recorrido** en cada iteración, **no su índice**.

Se tiene una lista heterogénea (que tiene elementos con distintos tipos de dato), y se desea contar los elementos strings **accediendo a su posición específica**.

Un primer enfoque sería implementar un loop clásico.

```
lista_heterogenea = ['Lechugas', 'Tomates', 5, 10,
                    True, False, True, 'Papas',
                    5.1, 45.2, 1, 2, 0]

count_str = 0
for i in lista_heterogenea:
    if type(lista_heterogenea[i]) is 'str':
        count_str += 1
    else:
        pass
```

```
-----

TypeError                                Traceback (most recent call last)

<ipython-input-25-f67ebd4371c7> in <module>()
      5 count_str = 0
      6 for i in lista_heterogenea:
----> 7     if type(lista_heterogenea[i]) is 'str':
      8         count_str += 1
      9     else:
```

```
TypeError: list indices must be integers or slices, not str
```

En este ejemplo, Python arrojará un `TypeError: list indices must be integers or slices, not str`. Este error indica que el índice de la lista es inválido.

Esto se debe a que se está utilizando un string, el elemento que se está recorriendo en la iteración, como índice de la posición del elemento.

Para resolver este problema, se necesita encontrar una función que entregue tanto la *posición* como el *elemento en sí* de la lista. Para ello se debe utilizar `enumerate`.