

Funciones con valores por defecto

Antonio Frías Delgado

En un ejercicio se pedía escribir una función a la que pasábamos un texto etiquetado por Freeling y devolvía tuplas (token, categoría). A veces nos interesaría la categoría completa y otras sólo la categoría principal, la primera letra de la misma. Podemos escribir dos funciones distintas, una función que devuelva la categoría completa y después reducirla a la primera letra, etc. Una opción razonable es escribir una función con valores por defecto. Ese tipo de funciones las han usado ya. Por ejemplo, los métodos de ordenación suelen tener un último valor por defecto: `reverse=False`. Si sólo le indicamos qué ordenar, hará un ordenamiento ascendente. Para que nos ordene de forma descendente tenemos que añadir: `reverse=True`. Podemos escribir nuestra función usando valores por defecto, digamos que lo que más vamos a usar es la categoría principal. Una posible implementación puede ser ésta:

```
def etiquetado_freeling(texto,categorias_reducidas=True):
    texto_segmentado=texto.split('\n')
    tuplas=[]
    for linea in texto_segmentado:
        if linea=='':
            continue
        words=linea.split()
        if categorias_reducidas==True:
            tuplas.append((words[0],words[2][0]))
        else:
            tuplas.append((words[0],words[2]))
    return tuplas
```

Supongamos que en la variable 'q' tenemos el texto del Quijote tal como resulta del etiquetado por Freeling. Podemos usar nuestra función así:

```
tuplas_simples=etiquetado_freeling(q)
tuplas_completas=etiquetado_freeling(q,categorias_reducidas=False)
O, abreviando:
tuplas_completas=etiquetado_freeling(q,False)
```