

Diccionarios con valores por defecto

Este script contiene unas notas breves y provisionales sobre este tema, por si facilita el uso de esta importante estructura de Python. Se agradece todo comentario hasta que pueda darlo por bueno.

Un ejemplo inicial

Los diccionarios se usan con frecuencia como colecciones de acumuladores. Ejemplo:

Contabilizar cuántas veces aparece cada letra de un texto

Para cada letra, tendremos un contador. Hagámoslo con diccionarios convencionales.

```
In [1]: ▶ # Con un diccionario clásico:

def cuenta_letras(texto):
    contadores = dict()
    for car in texto:
        if car.isalpha():
            if car in contadores:
                contadores[car] = contadores[car] + 1
            else:
                contadores[car] = 1
    return contadores

lluvia_proust = """
Un golpecito en el cristal, como si hubiesen tirado algo;
luego un caer ligero y amplio, como de granos de arena lanzados desde un
y por fin, ese caer que se extiende, toma reglas, adopta un ritmo
y se hace fluido, sonoro, musical, incontable, universal: llueve."
"""

frecuencias = cuenta_letras(lluvia_proust)
print(frecuencias)

{'U': 1, 'n': 17, 'g': 6, 'o': 22, 'l': 15, 'p': 4, 'e': 30, 'c': 9,
'i': 15, 't': 9, 'r': 14, 's': 13, 'a': 24, 'm': 6, 'h': 2, 'u': 10,
'b': 3, 'd': 10, 'y': 3, 'z': 1, 'v': 3, 'f': 2, 'q': 1, 'x': 1}
```

In [2]: `# Dos comprobaciones más:`

```
print(frecuencias["a"])
print(frecuencias["w"])
```

24

```
-----
----
KeyError                                Traceback (most recent call 1
ast)
Cell In[2], line 4
      1 # Dos comprobaciones más:
      3 print(frecuencias["a"])
----> 4 print(frecuencias["w"])
```

KeyError: 'w'

En efecto: la `z` no está presente en el texto y por lo tanto, tampoco en el diccionario. Por eso obtenemos un error al consultar su frecuencia.

En un *defaultdict*, un diccionario por defecto, se asigna un valor por defecto a los elementos ausentes. Esto tiene dos consecuencias. Para comentarlas, sigo refiriéndome a nuestro ejemplo de las letras.

1. Cuando se realiza el recorrido. Al realizar la contabilidad, no hace falta comprobar previamente si está la letra ya en el diccionario. El valor "inexistente" se tomará como el valor "por defecto", esto es, un cero.

Comprobemos esto:

In [3]: `# Con un diccionario con valores por defecto:`

```
from collections import defaultdict

def cuenta_letras(texto):
    contadores = defaultdict(int) # int pone el valor por defecto a cero
    for car in texto:
        if car.isalpha():
            contadores[car] = contadores[car] + 1
    return contadores

lluvia_proust = """
Un golpecito en el cristal, como si hubiesen tirado algo;
luego un caer ligero y amplio, como de granos de arena lanzados desde un
y por fin, ese caer que se extiende, toma reglas, adopta un ritmo
y se hace fluido, sonoro, musical, incontable, universal: llueve."
"""

frecuencias = cuenta_letras(lluvia_proust)
print(frecuencias)
```

```
defaultdict(<class 'int'>, {'U': 1, 'n': 17, 'g': 6, 'o': 22, 'l': 15,
'p': 4, 'e': 30, 'c': 9, 'i': 15, 't': 9, 'r': 14, 's': 13, 'a': 24,
'm': 6, 'h': 2, 'u': 10, 'b': 3, 'd': 10, 'y': 3, 'z': 1, 'v': 3, 'f':
2, 'q': 1, 'x': 1})
```

Y otra consecuencia:

2. Cuando se consulta un valor inexistente del diccionario, no se produce un error, sino que se añade al diccionario con el valor por defecto.

Comprobemos esto:

```
In [4]:  ▶ mi_dic = defaultdict(int)
          print(mi_dic)
          print(mi_dic["nuevo"])
          print(mi_dic)

defaultdict(<class 'int'>, {})
0
defaultdict(<class 'int'>, {'nuevo': 0})
```

Otros valores por defecto

Con enteros, hemos usado `int` para representar el `0` como valor por defecto.

Igualmente: Con reales, `float` representa `0.0`. Con listas, `list` representa `[]`. Con conjuntos, `set` representa `{}`.

```
In [5]:  ▶ # Con Listas:

dic_letras = defaultdict(list)
print(dic_letras)

for letra in "supercalifragilisticoexpialidoso":
    if letra in "aeiou":
        dic_letras["vocales"].append(letra)
    else:
        dic_letras["consonante"].append(letra)

print(dic_letras)

# Con conjuntos:

dic_letras = defaultdict(set)
print(dic_letras)

for letra in "supercalifragilisticoexpialidoso":
    if letra in "aeiou":
        dic_letras["vocales"].add(letra)
    else:
        dic_letras["consonante"].add(letra)

print(dic_letras)

defaultdict(<class 'list'>, {})
defaultdict(<class 'list'>, {'consonante': ['s', 'p', 'r', 'c', 'l', 'f', 'r', 'g', 'l', 's', 't', 'c', 'x', 'p', 'l', 'd', 's'], 'vocales': ['u', 'e', 'a', 'i', 'a', 'i', 'i', 'i', 'o', 'e', 'i', 'a', 'i', 'o', 'o']})
defaultdict(<class 'set'>, {})
defaultdict(<class 'set'>, {'consonante': {'f', 'l', 'p', 'd', 's', 'g', 't', 'r', 'x', 'c'}, 'vocales': {'a', 'u', 'i', 'o', 'e'}})
```

Defaultdict con defaultdict

Contabilicemos pares de letras consecutivas:

In [6]:  # Con tuplas:

```
dic_pares_letras = defaultdict(int)
print(dic_pares_letras)

frase = "superuperupereupercalicali...osososososososo"
for i in range(1, len(frase)):
    pre, post = frase[i-1], frase[i]
    dic_pares_letras[(pre, post)] += 1

print(dic_pares_letras)

defaultdict(<class 'int'>, {})
defaultdict(<class 'int'>, {('s', 'u'): 1, ('u', 'p'): 4, ('p', 'e'): 4, ('e', 'r'): 4, ('r', 'u'): 2, ('r', 'e'): 1, ('e', 'u'): 1, ('r', 'c'): 1, ('c', 'a'): 2, ('a', 'l'): 2, ('l', 'i'): 2, ('i', 'c'): 1, ('i', '.'): 1, (',', '.'): 2, (',', 'o'): 1, ('o', 's'): 7, ('s', 'o'): 7})
```

In [7]:  # Con diccionarios de diccionarios:

```
dic_pares_letras = defaultdict(lambda: defaultdict(int))
print(dic_pares_letras)

frase = "superuperupereupercalicali...osososososososo"
for i in range(1, len(frase)):
    pre, post = frase[i-1], frase[i]
    dic_pares_letras[pre][post] += 1

print(dic_pares_letras)

defaultdict(<function <lambda> at 0x0000025444D8F7E0>, {})
defaultdict(<function <lambda> at 0x0000025444D8F7E0>, {'s': defaultdict(<class 'int'>, {'u': 1, 'o': 7}), 'u': defaultdict(<class 'int'>, {'p': 4}), 'p': defaultdict(<class 'int'>, {'e': 4}), 'e': defaultdict(<class 'int'>, {'r': 4, 'u': 1}), 'r': defaultdict(<class 'int'>, {'u': 2, 'e': 1, 'c': 1}), 'c': defaultdict(<class 'int'>, {'a': 2}), 'a': defaultdict(<class 'int'>, {'l': 2}), 'l': defaultdict(<class 'int'>, {'i': 2}), 'i': defaultdict(<class 'int'>, {'c': 1, '.': 1}), '.': defaultdict(<class 'int'>, {'.'': 2, 'o': 1}), 'o': defaultdict(<class 'int'>, {'s': 7})})
```

O sea: tras la "s", viene la "u" una vez y la "o", siete veces...