



Python Basico - Ejercicios

Ejercicios

Responda las preguntas o complete las tareas descritas en negrita a continuación, use el método específico descrito si corresponde.

¿Cuánto es 7 elevado a 4?

```
In [1]: 7**4
```

```
Out[1]: 2401
```

Divida esta cadena:

```
s = "Hola mi estimado!"
```

en una lista.

```
In [4]: s = "Hola mi estimado!"
```

```
In [5]: s.split()
```

```
Out[5]: ['Hola', 'mi', 'estimado!']
```

Dadas las variables:

```
planeta = "Tierra"  
diametro = 12742
```

Use .format() para visualizar la siguiente frase:

El diametro de la Tierra es 12742 kilometros.

```
In [6]: planeta = "Tierra"  
diametro = 12742
```

```
In [7]: print("El diametro de la {} es {} kilometros.".format(planeta,diametro))
```

El diametro de la Tierra es 12742 kilometros.

Dada esta lista anidada, use la indexación para obtener la palabra "hola"

```
In [8]: lst = [1,2,[3,4],[5,[100,200,['hola']],23,11],1,7]
```

```
In [9]: lst[3][1][2][0]
```

```
Out[9]: 'hola'
```

Dado este diccionario nido, tome la palabra "hola". Esté preparado, esto será molesto / complicado

```
In [10]: d = {'k1':[1,2,3,{'difícil':['oh','man','inicio',{'destino':[1,2,3,'hola']}]}]}
```

```
In [11]: d['k1'][3]['difícil'][3]['destino'][3]
```

```
Out[11]: 'hola'
```

¿Cuál es la principal diferencia entre una tupla y una lista?

```
In [12]: # La tupla es immutable
```

Cree una función que tome el dominio del sitio web de correo electrónico a partir de una cadena con el siguiente formato:

user@dominio.com

Entonces, por ejemplo, pasando "user@domain.com" retornaria: domain.com

```
In [13]: def dominioGet(email):  
         return email.split('@')[-1]
```

```
In [14]: dominioGet('user@dominio.com')
```

```
Out[14]: 'dominio.com'
```

Cree una función básica que devuelva Verdadero si la palabra 'perro' está contenida en la cadena de entrada. No se preocupe por casos extremos como una puntuación que se adjunta a la palabra perro, pero tenga en cuenta las mayúsculas.

```
In [15]: def ubicaDog(st):  
         return 'perro' in st.lower().split()
```

```
In [16]: ubicaDog('Esta el perro aqui?')
```

```
Out[16]: True
```

Cree una función que cuente el número de veces que aparece la palabra "perro" en una

cadena. Nuevamente ignore los casos extremos.

```
In [17]: def cuentaDog(st):  
         count = 0  
         for word in st.lower().split():  
             if word == 'perro':  
                 count += 1  
         return count
```

```
In [19]: cuentaDog('Este perro corre mas rapido que este otro perro no lo dudes!')
```

```
Out[19]: 2
```

Utilice expresiones lambda y la función filter () para filtrar las palabras de una lista que comiencen con la letra 's'. Por ejemplo:

```
seq = ['sopa', 'perro', 'salida', 'gato', 'excelente']
```

debe filtrarse solo:

```
['sopa', 'salida']
```

```
In [20]: seq = ['sopa', 'perro', 'salida', 'gato', 'excelente']
```

```
In [21]: list(filter(lambda word: word[0]=='s', seq))
```

```
Out[21]: ['sopa', 'salida']
```

Ejercicio Final

Conduces demasiado rápido y un policía te detiene. Escribe una función para devolver uno de los 3 resultados posibles: "Sin papeleta", "Papeleta NO GRAVE" o "Papeleta GRAVE". Si tu velocidad es de 60 o menos, el resultado es "Sin papeleta". Si la velocidad está entre 61 y 80 inclusive, el resultado es "Papeleta NO GRAVE". Si la velocidad es 81 o más, el resultado es "Papeleta GRAVE". A menos que sea tu cumpleaños (codificado como un valor booleano en los parámetros de la función), en tu cumpleaños, la velocidad puede ser 5 más alta en total de casos.

```
In [27]: def calcula_velocidad(velocidad, es_mi_cumple):  
         if es_mi_cumple:  
             speed = velocidad - 5  
         else:  
             speed = velocidad  
  
         if speed > 80:  
             return 'Papeleta GRAVE'  
         elif speed > 60:  
             return 'Papeleta NO GRAVE'  
         else:  
             return 'Sin papeleta'
```

```
In [25]: calcula_velocidad(81,True)
```

```
Out[25]: 'Papeleta NO GRAVE'
```

```
In [26]: calcula_velocidad(81,False)
```

```
Out[26]: 'Papeleta GRAVE'
```

Buen trabajo!