Ejercicios Python 1

October 31, 2018

1 Ejercicios R

1.1 1. Escribe una función contar_letras(palabra,letra) que devuelva el número de veces que aparece una letra en una palabra.

```
In [5]: def contar_letras(palabra,letra):
            num_letras = 0
            for l in palabra:
                if(letra == l):
                    num_letras += 1

            return num_letras

        print(contar_letras("hola","a"))
        print(contar_letras("hola","e"))
```

```
1
0
```

1.2 2. Escribe una función eliminar_letras(palabra, letra) que devuelva una versión de palabra que no contiene el carácter letra.

```
In [20]: def eliminar_letras(palabra,letra):
             new_palabra = ''
             for l in palabra:
                 if l != letra:
                     new_palabra += l


             return new_palabra


         print(eliminar_letras("hola","a"))
         print(eliminar_letras("hola","o"))
```

```
hol
hla
```

1.3 3. Escribe una función buscar(palabra,sub) que devuelva la posición en la que se puede encontrar sub dentro de una palabra o -1 en caso de que no esté.

```
In [23]: def buscar(palabra,sub):
             sub_index = 0
             index = -1
             for i in range(len(palabra)):
                 if palabra[i] == sub[0]:
                     index = i
                     for j in range(1,len(sub)):
                         if i+j >= len(palabra):
                             index = -1
                         elif palabra[i+j] != sub[j]:
                             index = -1

             return index


         print(buscar("hola","la"))
         print(buscar("hola","lo"))
```

```
2
-1
```

1.4 4. Escribe una función num_vocales(palabra) que devuelva el número de vocales que aparece en la palabra.

```
In [25]: def num_vocales(palabra):
             vocales = 0
             vocals =  ['a','e','i','o','u']
             for l in palabra:
                 if l in vocals:
                     vocales += 1

             return vocales

         print(num_vocales("hola"))
         print(num_vocales("pepino"))
```

```
2
3
```

1.5 5. Escribe una función vocales(palabra) que devuelva las vocales que aparecen en la palabra.

```
In [28]: def vocales(palabra):
             vocales = []
```

```
            vocals = ['a','e','i','o','u']
            for l in palabra:
                if l in vocals:
                    vocales.append(l)
            return vocales

        print(vocales("hola"))
```

```
['o', 'a']
```

1.6 6. Escribe una función es_inversa(palabra1, palabra2) que devuelva True si una palabra es la misma que la otra pero ocn los caracteres en orden inverso. Por ejemplo 'absd' y 'dsba'

```
In [34]: def es_inversa(palabra1, palabra2):
            ini = 0
            fin = len(palabra2)-1
            inversa = True

            if len(palabra1) != len(palabra2):
                inversa = False
            else:
                while( ini < len(palabra1) and inversa):
                    if(palabra1[ini] != palabra2[fin]):
                        inversa = False
                    ini += 1
                    fin -= 1

            return inversa

        print(es_inversa("absd","dsba"))
```

```
True
```

1.7 7. Escribe una función comunes(palabra1, palabra2) que devuelva una cadena formada por los caracteres comunes a las dos palabras.

```
In [35]: def comunes(palabra1,palabra2):
            caracteres_comunes = ''
            for l in palabra1:
                if l in palabra2 and l not in caracteres_comunes:
                    caracteres_comunes += l
            return caracteres_comunes

        print(comunes("hola","olaa"))
```

```
ola
```

1.8 8. Escribe una función eco_palabra(palabra) que devuelva una cadena formada por palabra repetida tantas veces como sesa su longitud. Por ejemplo 'hola' -> 'holaholaholahola'

```
In [38]: def eco_palabra(palabra):
             eco = ''
             for veces in range(len(palabra)):
                 eco += palabra

             return eco

         print(eco_palabra("hola"))
```

```
holaholaholahola
```

1.9 9.Escribe una funicón palindromo(frase) que determiensi frase es un palíndromo. Es decir, que se lea igual de izquierda a derecha que de derecha a izquierda (sin considerar espacios).

```
In [44]: def palindromo(frase):
             es_palindromo = True
             mitad = (len(frase)-1)/2
             frase = frase.replace(" ","")
             ini = 0
             fin = len(frase)-1
             while( es_palindromo and ini < mitad
                  and fin > mitad):

                 if frase[ini] != frase[fin]:
                     es_palindromo = False

                 ini += 1
                 fin -= 1

             return es_palindromo

         print(palindromo("a cavar a caravaca"))
         print(palindromo("hola hola"))
```

```
True
False
```

1.10 10 Escribe una función orden_alfabético(palabra) que determina si las letras que forman palabra aparecen en orden alfabético. Por ejemplo 'abejo'

```
In [53]: def orden_alfabetico(palabra):
             palabra_ord = ''.join(sorted(palabra))
```

```
            return palabra_ord == palabra

        print(orden_alfabetico("abejo"))
        print(orden_alfabetico("hola"))
```

```
True
False
```

1.11 11. Escribe una función trocear(palabra, num) que devuelva una lista con trozos de tamaño num de palabra.

```
In [58]: def trocear(palabra,num):
            ini = 0
            trozos = []
            while( (ini+num) < len(palabra)):
                trozos.append(palabra[ini:ini+num])
                ini += num

            trozos.append(palabra[ini:])
            return trozos

        print(trocear("hola",1))
        print(trocear("hola",2))
        print(trocear("hola",3))
```

```
['h', 'o', 'l', 'a']
['ho', 'la']
['hol', 'a']
```

1.12 12. Un anagrama de una palabra pal1 es una palabra formada con las mismas letras que pal1 pero en orden disitinto. Escribe una función anagrama(palabra1,palabra2) que determine si es una anagrama. Ejemplo: marta - trama

```
In [60]: def anagrama(palabra1,palabra2):
            palabra1_ord = ''.join(sorted(palabra1))
            palabra2_ord = ''.join(sorted(palabra2))
            return palabra1_ord == palabra2_ord

        print(anagrama("marta","trama"))
        print(anagrama("patatas","hola"))
```

```
True
False
```