

Programación para Data Science

Unidad 3: Conceptos avanzados de Python

Ejercicios y preguntas teóricas

A continuación, encontraréis la parte que tenéis que completar en este modulo y las preguntas teóricas a contestar.

Pregunta 1

Las funciones **range** y **xrange** pueden utilizarse con la misma finalidad, pero su funcionamiento es distinto. Explicad qué beneficios ofrece el uso de la función **xrange** comparada con la función **range** y comentad para qué sirve la palabra reservada **yield**. Poned un ejemplo de uso de **range**, **xrange** y otro de **yield**. (1 punto)

Respuesta:

Respuesta

Pregunta 2

Las excepciones son errores detectados en tiempo de ejecución. Pueden y deben ser manejadas por el programador para minimizar el riesgo de que un determinado programa falle de forma no controlada. Escribid, en lenguaje Python, cómo generar y capturar las siguientes tres excepciones. (1 punto)

a) **ZeroDivisionError**

In []:

```
# Respuesta
```

b) **IndexError**

In []:

```
# Respuesta
```

c) **NameError**

In []:

```
# Respuesta
```

Ejercicio 1

Completad el código necesario para calcular el número de vocales y de consonantes respectivamente de un texto. (1 punto)

In [1]:

```
def contar_vocales_y_consonantes(texto):
    # Cuenta las vocales contenidas en el string texto y también las consonante
    num_vocales = 0
    num_consonantes = 0

    # Código que hay que completar.

    return num_vocales, num_consonantes

texto = "Orbiting Earth in the spaceship, I saw how beautiful our planet is. \
        People, let us preserve and increase this beauty, not destroy it!"

num_vocales, num_consonantes = contar_vocales_y_consonantes(texto)
print "El número de vocales es de %d" % num_vocales
print "El número de consonantes es de %d" % num_consonantes
```

El número de vocales es de 0

El número de consonantes es de 0

Ejercicio 2

Dada una molécula representada por un **string** del estilo C9-H8-O4 calculad su masa atómica. Por ejemplo, para una molécula C4-H3, la masa atómica sería de $4 \cdot 12.01 + 3 \cdot 1.007825$.

Haced una solución general accediendo al diccionario mediante la clave, que en este caso será el tipo de átomo. Por ejemplo, para la molécula C5-H3 deberíamos seguir estos pasos:

- Separar la molecula por los guiones (podemos hacerlo con la función split, por ejemplo).
- Para cada una de las partes, C5 y H3, encontrar el tipo de átomo: C y H (necesitaremos un bucle de algún tipo aquí).
- Acceder al diccionario de masas y para la clave que se corresponde con el tipo de átomo, obtener la masa.
- Una vez encontrada la masa, multiplicarla por el número de átomos encontrados.

Pista: para un string del tipo a = 'C15', a[0] nos proporcionará el tipo de átomo, C. a[1:] nos proporciona el string restante: '15'. Tened en cuenta que ha de convertirse a número decimal para poder multiplicarse.

(1.5 puntos)

In [2]:

```
# Masas atómicas
masas = {'H': 1.007825, 'C': 12.01, 'O': 15.9994, 'N': 14.0067, 'S': 31.972071,

def calcula_masa_atomica(molecula):
    """
    Calcula la masa atómica de una molécula
    """
    masa = 0.0

    # Código que hay que completar.

    return masa

print calcula_masa_atomica('H2-O')
print calcula_masa_atomica('H2-S-O4')
print calcula_masa_atomica('C2-H5-O-H')
print calcula_masa_atomica('H2-N-C6-H4-C-O-2H')
```

```
0.0
0.0
0.0
0.0
```

Ejercicio 3

Completad las siguientes funciones matemáticas y documentad el código también de cada función. Por último, escribid algún ejemplo de uso de cada una de las funciones. **(1.5 puntos)**

In [4]:

```
# Completad las siguientes funciones matemáticas
import math

def area_triangulo(base, altura):
    # Código que hay que completar.
    return 0.

def longitud_circulo(radio):
    # Código que hay que completar.
    return 0.

def tangente(angulo):
    # Código que hay que completar.
    return 0.

def area_rectangulo(lado1, lado2):
    # Código que hay que completar.
    return 0.

def volumen_esfera(radio):
    # Código que hay que completar.
    return 0.

def volumen_cubo(lado):
    # Código que hay que completar.
    return 0.

# Escribid aquí dos ejemplos de uso para cada una de las funciones anteriores,
# print 'El volumen del cubo de lado 3.5 es %f' % volumen_cubo(3.5)
# print 'El volumen del cubo de lado 12.3 es %f' % volumen_cubo(12.3)
```

Ejercicio 4

El siguiente ejercicio consiste en pasar un número en base 16 (hexadecimal, 0-9/A-F) a base 10 (decimal).

Dado un **string** que representa un número en hexadecimal, por ejemplo, AE3F, devolved el número natural correspondiente, en este caso, 44607. **(1.5 puntos)**

In [5]:

```
# Código que hay que completar: DEFINID UNA FUNCIÓN y escribid tres casos de us
```

Ejercicio 5

Uno de los algoritmos más básicos en criptografía es el cifrado César (https://es.wikipedia.org/wiki/Cifrado_C%C3%A9sar), que fue utilizado por Julio César para comunicarse con sus generales, y que consiste en dado un texto, por cada una de las letras del texto, añadirle un

desplazamiento para conseguir una nueva letra diferente de la original. Comprenderemos rápidamente su mecanismo mediante un ejemplo:

Si asignamos el número 1 a la primera letra del abecedario, A, 2 a la siguiente, B, etc., imaginad que tenemos el siguiente mensaje: ABC 123

Si aplicamos un desplazamiento de 3, buscaremos cuál es la letra en el abecedario que se corresponde: DEF 456

ABC se ha convertido en DEF porque hemos sumado un desplazamiento de 3. También podríamos aplicar otros tipos de desplazamiento como los negativos. Por ejemplo, para el desplazamiento -1 y el mensaje original ABC tendríamos un mensaje cifrado de: ZAB.

Escribid una función que dado un mensaje cifrado y un desplazamiento, calcule y devuelva el mensaje original. **(1.5 puntos)**

Nota. Debéis usar como alfabeto de entrada el **alfabeto español en mayúsculas**.

In [6]:

```
def descifrado_cesar(mensaje_cifrado, desplazamiento=1):
    """
    Descifra el mensaje utilizando el metodo de Cesar dado un desplazamiento
    """
    mensaje_original = ""

    # Código que hay que completar.

    return mensaje_original

# Añadir al menos dos ejemplos más.
print descifrado_cesar("QSPHSBNBDJPN QBSB EBUB TDJFNDF", 1)
```

Ejercicio 6

El paradigma de programación orientada a objetos es ampliamente utilizado en gran parte de las librerías que se escriben en Python. Es una forma útil de encapsular información de la que se ocupará el propio objeto donde se ha definido esa información. Debéis usar este concepto para: **(1 punto)**

a) Escribir en Python el código necesario para crear una clase Animal con dos atributos, nombre y onomatopeya ("miau", "guau", etc.), y un método, diAlgo, que debería imprimir por pantalla el nombre del animal y su onomatopeya. Por ejemplo, la salida podría ser algo así: "Terry acaba de decir miau."

In [8]:

```
# Respuesta
```

b) Poner un ejemplo de uso de la clase anterior.

In [9]:

```
# Respuesta
```

c) Crear una subclase y una superclase de la clase Animal.

In [10]:

```
# Respuesta
```

Podéis basaros en el material [Object Oriented Program Design \(http://life.bsc.es/pid/brian/python/#/7\)](http://life.bsc.es/pid/brian/python/#/7).

Ejercicio 7 (Opcional)

Una gran multinacional ha decidido cambiar su logo basado en los tres caracteres que más aparecen en el nombre de la empresa. En estos momentos están probando varias combinaciones de nombres de empresa y logos basados en la condición anterior. Dicho esto, vuestra tarea es crear una función que, dado el nombre de la empresa, devuelva los tres caracteres que más aparecen en el nombre.

Deberéis:

- Imprimir por pantalla los tres caracteres más comunes junto con el número de veces que aparecen.
- Ordenar los caracteres de forma descendiente basado en el número de veces que aparecen.
- Si los caracteres aparecen el mismo número de veces, deberéis ordenarlos en orden alfabético.

Por ejemplo, dadas las condiciones anteriores, GOOGLE crearía su logo usando las letras G, O y E.

Escribid una función en Python que, dada una cadena de caracteres, devuelva los tres caracteres que más aparecen en ella. La cadena que se os proporcionará será siempre en minúscula y tenéis que imprimir cada caracter en una línea separada. Deberéis mostrar el resultado para las siguientes empresas:

- china construction bank
- jpmorgan chase
- general electric

In [1]:

```
# Respuesta
```