



ISIS-1221 INTRODUCCIÓN A LA PROGRAMACIÓN

Nivel 1 – Laboratorio 3 Módulos y Consolas

Objetivo general

El objetivo general de este laboratorio es practicar la construcción y documentación de funciones, así como introducir los conceptos de interfaz por consola y de módulo.

Objetivos específicos

1. Familiarizarse con la lectura de documentación de funciones.
2. Implementar funciones a partir de la documentación.
3. Crear y usar un módulo.
4. Construir una interfaz basada en consola para interactuar con el usuario.
5. Aprender a conectar la interfaz por consola y los módulos.
6. Familiarizarse con la técnica de dividir y conquistar (dividir un problema en sub-problemas).
7. Practicar el llamado de funciones con parámetro y el llamado de funciones desde otras funciones (composición de funciones).

Preparación del ambiente de trabajo

1. Cree una carpeta para trabajar, poniéndole su nombre o login.
2. Descargue de BrightSpace el archivo con el “esqueleto” de la aplicación (n1-lab3-esqueleto.zip) y descomprímalo en su carpeta de trabajo. El esqueleto consiste en un conjunto de archivos que usted va a usar o a modificar.
3. Elimine el archivo comprimido que descargó.
4. Abra Spyder y cambie la carpeta de trabajo para que sea la carpeta con el esqueleto.

Actividad 1: Análisis de años

El primer caso representa un analizador de años. Este es un programa sencillo que le permitirá conocer el milenio, el siglo y la década a la cual pertenece el año de interés ingresado por el usuario. El programa está compuesto por dos archivos: `analisis.py` y `consola_analisis.py`, los cuales corresponden respectivamente al módulo de funciones y a la interfaz por consola del programa. **El propósito de esta actividad es MUY SIMPLE: leer y entender tanto la documentación como la implementación de una función dentro de un módulo y de una interfaz por consola.**

Módulo “`analisis.py`”

Abra el archivo `analisis.py` y revise la documentación de la función `analizar_anio`, la cual se muestra a continuación:

```
def analizar_año(año:int)-> str:

    """ Realiza el análisis del año recibido por parámetro.
    Parámetros:
        año (int) Año que se desea analizar. Debe ser un número entero
        positivo.
    Retorno:
        str: Mensaje con la forma "El año X hace parte del milenio Y, siglo Z,
        y década W."
    """
```

La documentación de una función corresponde a la especificación completa de la función y es el resultado del análisis del problema que se desea resolver con dicha función. La especificación contiene:

- Una descripción general de la función (primera línea).
- Los parámetros que recibe la función y para cada uno de estos: su nombre, su tipo (entre paréntesis) y una breve descripción. Si hay alguna restricción sobre los datos de entrada, debe ser descrita. En este caso, por ejemplo, que el año debe ser un entero positivo.
- El valor de retorno de la función, para el cual se especifica igualmente el tipo y una breve descripción.

Note que la documentación de una función en Python comienza y termina con triple comillas """. Esto implica que el texto que se encuentra encerrado por estas comillas no es interpretado y puede ser escrito en lenguaje natural. Por la misma razón, se visualiza en un color diferente (verde).

Una vez leída y entendida la documentación, proceda a leer su implementación, la cual se muestra a continuación:

```
#Las siguientes instrucciones permiten calcular el milenio
#el siglo y la década del año ingresado por el usuario
milenio = año//1000 + 1
siglo = año//100 + 1
decada = (año%100)//10 + 1

#Se construye el string de respuesta
respuesta = "El año " + str(año) + " hace parte del milenio " + \
            str(milenio) + ", siglo " + str(siglo) + " y década " + \
            str(decada) + "."

#Se retorna el string
return respuesta
```

Note que las líneas de comentarios en lenguaje natural (en este caso español) comienzan con el carácter "#". Estas líneas no son ejecutadas y nos sirven a nosotros como programadores a escribir un código que sea más fácil de entender por otras personas o por nosotros mismos un tiempo después. Note también el carácter "\" al final de las líneas que componen el string de respuesta. Ponemos este carácter cuando deseamos escribir en varias líneas y no se genere error.

Interfaz por consola “consola_analisis.py”

Pasemos ahora a estudiar el archivo `consola_analisis.py`, el cual contiene la interfaz basada en consola que maneja la interacción del programa con el usuario:

```
consola_analisis.py
1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Jul 22 21:47:01 2019
4
5 @author: Cupi2
6 """
7
8 import analisis as mod
9
10 def ejecutar_analizar_anio()->None:
11     anio = int(input("Ingrese el año que desea analizar: "))
12     resultado = mod.analizar_anio(anio)
13     print(resultado)
14
15 def iniciar_aplicacion()->None:
16     print("Bienvenido al analizador de años")
17     ejecutar_analizar_anio()
18
19 #PROGRAMA PRINCIPAL
20 iniciar_aplicacion()
21
```

La primera instrucción de la consola es la importación del módulo “analisis” donde se encuentra la función `analizar_anio`. Al importar el módulo lo renombramos como “mod”. Es importante tener presente que para que la importación funcione ambos archivos (módulo y consola) deben estar guardados en la misma carpeta.

Este archivo de interfaz por consola sigue el estándar que enseñamos en este curso. Esto es:

- El programa o bloque principal llama a la función `iniciar_aplicacion`.
- La función `iniciar_aplicacion`, como su nombre lo indica es el inicio del programa.
- Las demás funciones (en este caso hay una sola) se llaman `ejecutar_XX`, donde XX es alguna de las funciones de los módulos importados.

Actividad 2: Distancia entre dos puntos de la tierra

La superficie de la Tierra es curva, y la distancia entre grados de longitud varía con la latitud. Como consecuencia, encontrar la distancia entre dos puntos en la superficie terrestre es más complicado que simplemente usar el teorema de Pitágoras. Siendo (t_1 , g_1) y (t_2 , g_2) la latitud y longitud de dos puntos en la superficie de la Tierra, la distancia entre estos puntos, en kilómetros, es aproximada usando la fórmula: $d = r\Delta\sigma$, donde r , el **radio de la esfera**, es 6371.01 y $\Delta\sigma$ es el **ángulo central** entre los dos puntos, que se calcula de la siguiente manera:

$$\Delta\sigma = \arccos(\sin(t_1) * \sin(t_2) + \cos(t_1) * \cos(t_2) * \cos(g_1 - g_2))$$

Cree una función que reciba la latitud y longitud de dos puntos en la superficie de la Tierra y retorne la distancia entre estos, en kilómetros. El valor de retorno debe estar aproximado a dos cifras decimales.

Usted debe completar tanto el módulo con las funciones que hacen los cálculos, como la interfaz por consola con el programa principal.

En este laboratorio le estamos entregando un *esqueleto* de la solución, que es un programa que está parcialmente escrito y que usted debe completar, siguiendo las instrucciones descritas en este enunciado y llenando los espacios marcados dentro del código con el comentario `#TODO`. Las líneas donde aparece el comentario `#TODO` tienen además un chulito de color azul a la izquierda, para que se puedan encontrar fácilmente. **Hágalo paso por paso.**

Módulo “`distancia_puntos.py`”

Vamos a descomponer el problema de calcular la edad en 3 sub-problemas, y para cada uno de estos vamos a implementar una función que lo resuelva.

Sub-problema 1:

Pasar un número en grados a un número en radianes. Por ejemplo, 4.601615 grados equivalen a 0.08031333265915887 radianes.

El encabezado de la función es el siguiente:

```
def grados_a_radianes(grados:float)->float:
```

Usted debe completar la implementación de la función dentro del esqueleto que le fue entregado.

- La función `radians(...)` del módulo `math` puede serle de utilidad.
- Muchas funciones matemáticas no hacen parte del lenguaje python, sino que están definidas dentro de un módulo independiente llamado `math`. **Asegúrese de importarlo**, de lo contrario no podrá hacer esta conversión, ni hacer el cálculo de $\Delta\sigma$ en el siguiente sub-problema.

Sub-problema 2:

Calcular el ángulo central ($\Delta\sigma$) a partir de la fórmula dada. Recuerde que la fórmula es la siguiente:

$$\Delta\sigma = \arccos(\sin(t1) * \sin(t2) + \cos(t1) * \cos(t2) * \cos(g1 - g2))$$

El encabezado de la función es el siguiente:

```
def calcular_angulo (t1:float,g1:float,t2:float,g2:float)->float:
```

Usted debe completar la implementación de la función dentro del esqueleto que le fue entregado. Dos cosas importantes:

- Estas funciones del módulo `math` esperan parámetros en radianes, que debió haber convertido con el método `grados_a_radianes`.
- La idea es invocar las funciones `sin`, `cos`, y `acos` del módulo `math`. Por ejemplo, para calcular el seno de 0.5 radianes escribimos `math.sin(0.5)`.

Sub-problema 3:

Redondear a dos decimales un número flotante.

El encabezado de la función es el siguiente:

```
def redondear (numero:float)->float:
```

Usted debe completar la implementación de la función dentro del esqueleto que le fue entregado.

Interfaz por consola “consola_distancia_puntos.py”

Este archivo de interfaz por consola sigue el estándar que enseñamos en este curso. Esto es:

- El programa o bloque principal llama a la función iniciar_aplicacion.
- La función iniciar_aplicacion, como su nombre lo indica es el inicio del programa.
- Las demás funciones (en este caso hay una sola) se llaman ejecutar_XX, donde XX es alguna de las funciones de los módulos importados.

Usted debe completar la implementación de la consola dentro del esqueleto que le fue entregado.

Entrega

Cree un archivo comprimido .zip con los dos archivos distancia_puntos.py y consola_distancia_puntos.py. Entregue el archivo comprimido a través de Brightspace en el laboratorio del Nivel 1 designado como “L3: Módulos y consolas”. El nombre del archivo comprimido debe ser n1_l3_login.zip. Donde login es su usuario uniandes.