



ISIS-1221 INTRODUCCIÓN A LA PROGRAMACIÓN

Nivel 1 – Laboratorio3 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 subproblemas).
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 x
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: Cálculo de la edad de una persona

Escriba un programa que calcule la edad de una persona a partir de su fecha de nacimiento y la fecha actual. Ambas fechas se dan en día, mes y año y la edad debe darse también de la misma forma. Suponga que todos los meses son de 30 días (es decir que un año tiene sólo 360 días). Por ejemplo, si la persona nació el 20 de noviembre de 1986 (día_nacimiento: 20, mes_nacimiento: 11, año_nacimiento: 1986) y hoy es 8 de febrero de 2019 (día_actual: 8, mes_actual: 2, año_actual: 2019), entonces la persona tiene 32 años, 2 meses y 18 días.

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. Esto 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 “calculo_edad.py”

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

Subproblema 1:

Transformar una fecha cualquiera (expresada en año, mes y día) a días (suponiendo que todos los meses tienen 30 días). Este subproblema será resuelto por una función llamada `fecha_a_días`. Por ejemplo: si el año es 1986, el mes es 11 y el día es 20, esta función devuelve el valor 715310, que es el resultado de: multiplicar el año 1986 por 360 (días) + el mes 11 por 30 (días) + 20 (días).

El encabezado de la función es el siguiente:

```
def fecha_a_días(anio: int, mes: int, dia: int)->int:
```

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

Subproblema 2:

Calcular la diferencia en días entre dos fechas cualesquiera `fecha1` y `fecha2` (expresadas en año, mes y día). Una restricción sobre estos datos de entrada es que la `fecha2` siempre es superior a la `fecha1`. Este subproblema será resuelto por una función llamada `fecha_a_días`. Por ejemplo: si la `fecha1` es el 20 de noviembre de 1986 y la `fecha2` es 8 de febrero de 2019, esta función devuelve el valor 11598, que es el resultado de: restar 726908 (`fecha2` pasada a días) y 715310 (`fecha1` pasada a días).

El encabezado de la función es el siguiente:

```
def diferencia_fechas (anio_fecha2: int, mes_fecha2: int, dia_fecha2: int,  
                      anio_fecha1: int, mes_fecha1: int, dia_fecha1: int)->int:
```

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

Subproblema 3:

Calcular la edad de la persona a partir de su fecha de nacimiento y la fecha actual. Este subproblema será resuelto por una función llamada `calcular_edad`, la cual recibe por parámetro las dos fechas (cada valor por separado: día, mes y año), llama las funciones anteriormente definidas para calcular la edad de la persona en días y calcula a partir de ese valor, cuántos años, meses y días tiene la persona. Por último retorna un string con esta información. Continuando con el ejemplo anterior, dado que la diferencia en días entre las dos fechas es 11598, la edad de la persona es 32 años, 2 meses y 18 días. La explicación es la siguiente:

- $11598 // 360$ da como resultado **32 (años)**
- $11598 \% 360$ da como resultado **78 (días)**
- $78 // 30$ da como resultado **2 (meses)**
- $78 \% 30$ da como resultado **18 (días)**

En este caso, el mensaje que debe retornar esta función es: **“La persona tiene 32 años, 2 meses y 18 días”**.

El encabezado de la función es el siguiente:

```
def calcular_edad (anio_actual: int, mes_actual: int, dia_actual: int,  
                  anio_nacimiento: int, mes_nacimiento: int,  
                  dia_nacimiento: int)->str:
```

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

Interfaz por consola “`consola_calculo_edad.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 calculo_edad.py y consola_calculo_edad.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.