

Laboratorio 4

Programación 1

Primer Semestre 2017

Metas del módulo

- Capacidad para definir estructuras de decisión (if/else)
- Capacidad para definir estructuras de iteración (while/for)
- Capacidad para definir funciones.

1. Dado el siguiente código fuente:

```
def ejercicio_1(cadena):  
    i = 0  
    while cadena[i] == " ":  
        i += 1  
    return cadena[i:]
```

- a) Describir qué hace la función
- b) Enumerar los parámetros que recibe
- c) Escribir la salida de las siguientes llamadas a la función:
 - i. `ejercicio_1("programación uno")`
 - ii. `ejercicio_1(" programación uno")`
 - iii. `ejercicio_1("programación uno ")`
 - iv. `ejercicio_1(" programación uno ")`

El módulo random:

En Python, muchas funciones están implementadas en módulos que son archivos conteniendo definiciones con código fuente Python. Para utilizar un módulo es necesario importar el módulo mediante la declaración "import" seguido del nombre del módulo.

El módulo "random" contiene la función `random.randint` donde `random.randint(1,6)` devuelve un número aleatorio entre 1 y 6 (incluidos) y que se puede utilizar para simular la tirada de un dado.

2. **Simular 10 tiradas de dos dados simultáneamente, dando el promedio de la suma de los resultados de los dos dados.**

Función “input”

En Python, utilizamos la función “input” para las entradas por teclado. La función “input” toma como argumento un mensaje para el usuario que debe ser una cadena de caracteres. El valor que devuelve “input” es una cadena de caracteres. Puede ser necesario convertir el resultado a un número entero o decimal:

```
#Ejemplo uso de input
entrada = input('Entra un valor en años: ')
print("En pesos, son ", int(entrada), '.')
```

3. Escribir una función que recibe una fecha dada por día mes y año e imprime por pantalla la fecha del día anterior.

Ej.: Si la fecha recibida es 01,01,2016, se debe imprimir 31/12/2015. Tener en cuenta que Enero, Marzo, Mayo, Julio, Agosto, Octubre y Diciembre tiene 31 días, Febrero 28 (no se tienen en cuenta año bisiesto) y el resto de los meses 30 días.

Ej.: 10/11/1977 → 1977-11-10

4. Escribir una función que dibuja un rectángulo en pantalla utilizando el caracter “x”. El tamaño del rectángulo está dado por la cantidad de “x” en la base y en la altura.

Ej.: si la función recibe 7 y 5 como parámetro, imprime:

```
xxxxxxx
x      x
x      x
x      x
xxxxxxx
```

5. Utilizando la función input y el módulo random, crear un juego que permita acertar un número al azar. El juego comienza seleccionando un número entre 1 y 100. Luego se pregunta al usuario que arriesgue un número. En caso que acierte se indica la cantidad de intentos necesarios. En caso contrario se indica si el número es mayor o menor al que debe adivinar. El usuario podrá arriesgar no más de 10 veces.

6. Imprimir el valor correspondiente a la constante Pi y las primeras 10 sumas parciales de:

$$\sum_{k=0}^{\infty} \left(\frac{1}{16} \right)^k \left(\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} + \frac{1}{8k+6} \right)$$

7. Crear funciones que permitan imprimir por pantalla las siguientes figuras recibiendo cómo parámetro la cantidad de caracteres en cada línea:

* ** *** **** ***** ***** ***** ***** ***** ***** *****	***** ***** ***** ***** ***** ***** ***** ***** ***** ***** *****	***** ***** ***** ***** ***** ***** ***** ***** ***** ***** *****	* ** *** **** ***** ***** ***** ***** ***** ***** *****
---	---	---	---

Desafíos

1. La sucesión de Fibonacci es definida por las siguientes reglas:

$$f_0 = 0$$

$$f_1 = 1$$

$$f(n) = f(n-1) + f(n-2) \text{ si } n > 2$$

Definir una función para calcular un término "n" de la sucesión de Fibonacci.

2. Escribir una función que tome como parámetro un número y verifique si el número es de buena suerte (sus dígitos suman 21)