

TEL-101 Ayudantía 4
Vicente Llanos Á - Thomas Soto F
(vicente.llanos@usm.cl - thomas.soto@usm.cl)
12 de mayo de 2021

1. Funciones

1. Introducción a las funciones.

Segmento de código que realiza instrucciones que pueden ser utilizadas en cualquier parte de un programa. Las funciones en Python nos ayudan a poder reutilizar código y no escribir de más en ciertos momentos de la preparación de algún software. Son capaces de trabajar con o sin parámetros.

2. Creación de las funciones.

- `def nombreFuncion():` # Creamos la función sin parámetros a trabajar.
- `def nombreFuncion(parámetro1,parámetro2):` # Utilización de dos parámetros.

3. Cosas claves dentro de las funciones.

- Utilización de la palabra **return**.
- Todas las variables creadas dentro de la función, viven en ese entorno y no globalmente.
- Para poder trabajar con el retorno de una función, debemos asignarlo a una variable
- Si la función no es llamada dentro del código, es como si no existiese.
- Los parámetros a utilizarse, se deben definir fuera de la variable y no necesariamente debe tener los mismos nombres de los parámetros.

```
1      #Ejemplo
2      def multiplicacion(n1,n2):
3          resultado = n1*n2
4          return resultado
5
6      #Si hacemos esto, creamos la función, pero en ningún momento la estamos llamando.
7      #Para su utilización, hacemos lo siguiente.
8
9      resultadoFuncion = multiplicacion(3,4)
10     print(resultadoFuncion)
```

4. Variables globales.

- Son aquellas variables definidas fuera de cualquier función. Tienen un ambito global, es decir, pueden ser utilizadas en cualquier parte del programa. Sin embargo, no pueden ser modificadas dentro de alguna función (en ese caso se crearía una variable local con el mismo nombre).

2. Ejemplos.

1. Área de un círculo.

```
1     PI = 3.14
2     def areaCirculo(radio):
3         area = PI*radio**2
4         return area
5
6     r = 3
7     print(areaCirculo(r))
8     #Muestra el valor de 3.14*9 = 28.26
```

2. Promedio de notas dado de una lista.

```
1     listaNotas = [1,2,3,4,5]
2
3     def promedio(lista):
4         suma = 0
5         for i in lista:
6             suma+=i
7
8         promedio = suma/len(lista)
9         return promedio
```

3. Promedio de notas dado de una lista. forma 2.

```
1     listaNotas = [1,2,3,4,5]
2
3     def promedio(lista):
4         return sum(lista)/len(lista)
```

4. Lista ordenada de menor a mayor.

```
1     listaDatos = [5,2,5,23,1,6,10]
2
3     def sortDatos(lista):
4
5         listaOrdenada = lista
6         listaOrdenada.sort()
7         return listaOrdenada
8
9     print(sortDatos(listaDatos)) #[1,2,5,5,6,10,23]
```

5. Fórmula de Herón.

```

1  def heron(a,b,c):
2
3      s = (a+b+c)/2
4      area = (s*(s-a)*(s-b)*(s-c))**(1/2)
5      return area
6
7  print(heron(13,15,14))

```

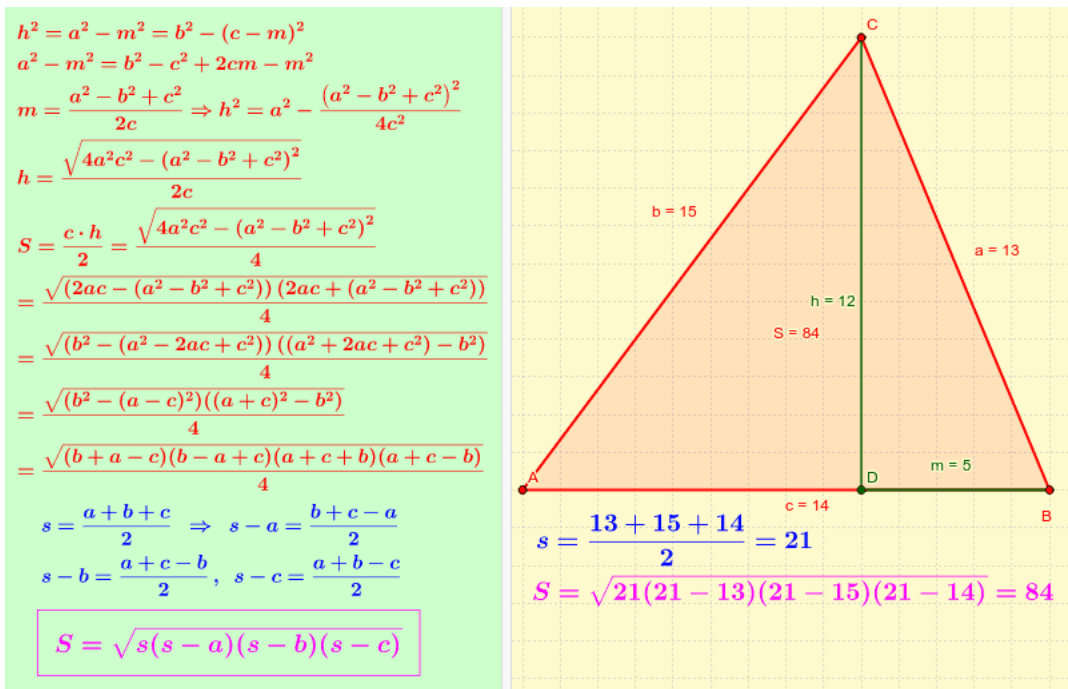


Figura 1: Fórmula de Herón. Demostración y resultado ejercicio.

3. Ejercicios.

1. Usted desea ir al cine a ver la próxima película del estudio Marvel lo antes posible para evitar los molestos spoilers. Para ello, la página del cine le muestra la lista sala con la disponibilidad de asientos. La sala del cine siempre será representada por una lista con n listas, cada una de ellas con n strings. Cada string puede tener los valores O que indica que el asiento está disponible o X que indica que está ocupado. El siguiente es un ejemplo para n igual a 5.

Nota: el valor de n no debería ser importante.

```
1  sala = [  
2  ["X", "X", "X", "O", "O"],  
3  ["X", "X", "X", "X", "O"],  
4  ["X", "O", "X", "O", "X"],  
5  ["X", "X", "X", "X", "O"],  
6  ["O", "O", "X", "O", "O"],  
7  ]  
8  sala2 = [  
9  ["X", "X", "X"],  
10 ["X", "X", "X"],  
11 ["X", "X", "X"],  
12 ["X", "X", "X"],  
13 ]
```

- Escriba la función `asientos_disponibles(sala)` que indique si la sala del cine tiene asientos disponibles. La función debe retornar `True` si la sala tiene asientos disponibles y `False` en caso contrario.
- Escriba la función `disponible(fila,columna,sala)` que reciba una sala del cine. La función debe retornar `True` si el asiento de la fila y columna ingresados está disponible y `False` en caso contrario. Si ingresa una fila o columna que no existan también deberá retornar `False`.
- Escriba la función `porcentaje_disponible(sala)` que indique porcentualmente la disponibilidad de una sala de cine.
- Usted desea ir con sus m amigos a ver la película y todos quieren ir a ver la película en la misma sala de cine. Debido al gran éxito en la pre-venta es posible que no queden muchos asientos. Escriba la función `hay_espacio_suficiente(m,sala)` que reciba un entero m con la cantidad de amigos (usted incluido) que quieren ir a ver la película y la sala del cine. La función debe retornar `True` si hay espacio suficiente para que todos puedan ver la película en la sala de cine o `False` en caso contrario :(
(