

Programación Científica 1

Taller 3: Funciones.

Profesores: Gilles Pieffet

Los programas del taller se pueden guardar en un archivo “ej3_funciones.py”, cada nuevo punto separado e indicado por un comentario:

```
##### Punto 1
... código Punto 1
```

```
##### Punto 2
... código Punto 2
```

1. **Función `raw_input()`.** Escribe un programa para calcular el volumen de una esfera ($V = \frac{4}{3}\pi r^3$) de radio r (básicamente el mismo problema que en el ejercicio 5 del taller 2), solo que esta vez r no está predefinido directamente adentro del programa pero hay que pedirlo de manera interactiva utilizando la función `raw_input()`. Para que el programa funcione de manera adecuada *tiene* que correrlo desde el terminal. Escribe el resultado con un formato de cadena para poder especificar las unidades: el radio r y el volumen V se pueden escribir en cm y cm³, respectivamente.

Utilice el programa para calcular el volumen de las esferas con radio $r = 1$ mm, 2 cm y 15 cm.

Nota 1: Cuidado con el tipo del valor de retorno de `raw_input()`.

Nota 2: Es una buena idea indicar cual es el dato que el programa está esperando.

2. Ahora vuelven a escribir el mismo programa, pero definiendo una función `volumen_esf()` que recibe como argumento el radio r y que tiene como valor de retorno el volumen V de la esfera. Utilice el valor de pi definido en el modulo `math`.
3. Escribe 2 funciones: la primera, `grados()`, para convertir ángulos de radianes a grados (la función recibe un ángulo en radianes y tiene como valor de retorno el mismo ángulo en grados) y la segunda, `radianes()` para convertir ángulos de grados a radianes (lo mismo que antes pero al revés). Verifique los resultados de las funciones con los ejemplos siguientes:

```
print grados(pi)
print grados(pi/2)
print grados(pi/6)
print radianes(180)
print radianes(360)
print grados(radianes(180))
print grados(radianes(30))
```

```
print radianes(grados(pi))
print radianes(grados(pi/2))
```

4. Python proporciona una función incorporada que se llama `len` que devuelva la longitud (es decir el numero de caracteres) de una cadena, así que `len('allen')` es igual a 5.

Escribe una función `alinear_derecha` que recibe una cadena s como argumento y imprime la cadena con suficientes espacios antes de tal manera que la ultima letra de la cadena se encuentra en la columna 70 del terminal.

```
>>> alinear_derecha('Gilles')
```

Gilles

Escribe los ejemplos siguientes para verificar que la función funciona correctamente:

```
alinear_derecha('Gilles')
alinear_derecha('Tim Burton')
alinear_derecha('James Rodriguez')
alinear_derecha('El conejito magico')
alinear_derecha(raw_input())
```

5. Un objeto función es un valor que se puede asignar a una variable o pasar como argumento. Por ejemplo `duplicar` es una función que recibe un objeto función como argumento y la llama dos veces.

```
def duplicar(f):
    f()
    f()
```

Aquí es un ejemplo que utiliza `duplicar` para llamar una función `imprimir_spam` dos veces:

```
def imprimir_spam():
    print 'spam'
```

```
duplicar(print_spam)
```

- a) Escribe este ejemplo y pruebalo.
- b) Escribe una versión mas general de `imprimir_spam` llamada `imprimir_2veces` que recibe una cadena como argumento y la imprime dos veces.
- c) Modificar `duplicar` de tal manera que reciba 2 argumentos, un objeto función y un valor y que llame la función 2 veces, pasandole el valor como argumento.
- d) Utilice la versión modificada de `duplicar` para llamar `imprimir_2veces` dos veces, pasando 'spam' como argumento.
- e) Definir una nueva función `cuadruplicar` que recibe una función como argumento y la llame 4 veces, pero de tal manera que el cuerpo de la función `cuadruplicar` solo contenga 2 instrucciones.