

Funciones Avanzadas

Programación

<http://progra.usm.cl>

UTFSM



Departamento de Informática
Universidad Técnica Federico Santa María

Uso de módulos

Dos maneras de importar objetos:

```
from math import exp, cos, pi
from random import randrange, choice

print exp(5.5)
print cos(pi / 2)
print randrange(10)
print choice(['Lunes', 'Martes', 'Viernes'])
```

```
import math
import random

print math.exp(5.5)
print math.cos(math.pi / 2)
print random.randrange(10)
print random.choice(['Lunes', 'Martes', 'Viernes'])
```

Creación de módulos

programa.py

```
from circulos import area, perimetro

radio = float(raw_input('Radio:'))
print 'El area es', area(radio)
print 'El perimetro es', perimetro(radio)
```

circulos.py

```
pi = 3.1415926535897931

def area(r):
    return pi * r ** 2

def perimetro(r):
    return 2 * pi * r
```

Funciones con múltiples valores de retorno

```
def convertir_segundos(segundos):  
    horas = segundos / (60 * 60)  
    minutos = (segundos / 60) % 60  
    segundos = segundos % 60  
    return horas, minutos, segundos
```

```
>>> h, m, s = convertir_segundos(9814)  
>>> h  
2  
>>> m  
43  
>>> s  
34
```

Funciones sin valor de retorno

```
def imprimir_datos(nombre, apellido, rol,
                   dia, mes, anno):
    print 'Nombre completo:', nombre, apellido
    print 'Rol:', rol
    print 'Fecha de nacimiento:',
    print dia, '/', mes, '/', anno

imprimir_datos('Perico', 'Los Palotes',
               '201101001-1', 3, 1, 1993)
imprimir_datos('Yayita', 'Vinagre',
               '201101002-2', 10, 9, 1992)
imprimir_datos('Fulano', 'De Tal',
               '201101003-3', 14, 5, 1990)
```

Parámetros con valores por omisión

```
def f(a, b=2):  
    return a + b
```

```
>>> f(10)
```

```
12
```

```
>>> f(10, 15)
```

```
25
```

Funciones como parámetros

```
def sumar(n, f):  
    s = 0  
    for i in range(n):  
        s = s + f(i)  
    return s  
  
def identidad(x):  
    return x  
  
def cuadrado(x):  
    return x ** 2  
  
def cubo(x):  
    return x ** 3  
  
print sumar(1000, identidad)  
print sumar(1000, cuadrado)  
print sumar(1000, cubo)
```

Funciones recursivas

$$\begin{aligned}n! &= 1 \cdot 2 \cdot 3 \cdot \dots \cdot (n - 2) \cdot (n - 1) \cdot n \\ &= (n - 1)! \cdot n\end{aligned}$$

```
def factorial(n):  
    if n == 0:  
        return 1  
    return factorial(n - 1) * n
```