

## PROCEDIMIENTOS Y ARCHIVOS EXTERNOS

### Procedimientos y funciones

Un procedimiento es un conjunto de instrucciones definidas al inicio de un programa y que pueden ser llamadas más de una vez durante la corrida del mismo. En la programación estructurada, los procedimientos son los elementos básicos para desarrollar programas más funcionales. Al ser llamadas o invocadas varias veces, minimizan el código, aumentan la legibilidad y fomentan la reutilización de código. A diferencia de otros lenguajes, en *Python*, un procedimiento no existe hasta que es invocado. Se usa la palabra reservada *def* para crear un procedimiento o definir una función, con la diferencia que la segunda devuelve un valor; en los siguientes dos ejemplos se usan procedimientos y funciones para entender su programación.

Para calcular los coeficientes binomiales del binomio de Newton  $(a+b)^n$  se usa la expresión:

$${}_nC_r = n! / ((n-r)!r!)$$

Así, el desarrollo de  $(a+b)^4$  es  $a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + b^4$ , donde los coeficientes 1, 4, 6, 4 y 1 se forman de  ${}_4C_0$ ,  ${}_4C_1$ ,  ${}_4C_2$ ,  ${}_4C_3$  y  ${}_4C_4$ .

El programa, usando procedimientos y funciones, se presenta a continuación:

```
"""Programa para calcular los coeficientes binomiales del desarrollo
del binomio de Newton (a+b)^n"""
import math as m
def fact(n):
    f=1
    for i in range(1,n+1):
        f=f*i
    return f
def CoefBin(n,r):
    coef=fact(n)/(fact(n-r)*fact(r))
    return coef
def Newton():
    print("Coeficientes binomiales del binomio de Newton")
    print()
    n=eval(input("Introduzca exponente n de (a+b)^n: "))
    coef_bin=[]
    for k in range(0,n+1):
        coef_bin.append(CoefBin(n,k))
    print("Coeficientes: ",coef_bin)
```

```
Newton()
```

```
Coeficientes binomiales del binomio de Newton
```

```
Introduzca exponente n de (a+b)^n: 4
```

```
Coeficientes: [1.0, 4.0, 6.0, 4.0, 1.0]
```

Se observa la definición de tres procedimientos, de los cuales dos corresponden a funciones: *fact()* y *CoefBin()* ya que regresan los valores de *f* y *coef*, respectivamente, y un procedimiento: *Newton()*, que se invoca en el cuerpo del programa para solicitar el exponente (*n*) y llamar a las funciones para obtener los factoriales y los coeficientes binomiales. Como ejercicio, modificar el programa anterior para obtener los coeficientes binomiales y generar el triángulo de *Pascal*:

```
1,1
1,2,1
1,3,3,1
1,4,6,4,1
1,5,10,10,5,1
....
```

El siguiente programa realiza el cálculo del límite líquido usando la ecuación de *Lambe*, que establece que  $LL=1/2*(w_1*(N_1/25)^{0.121}+w_2*(N_2/25)^{0.121})$ , donde  $w_1$  y  $w_2$  son las humedades registradas para  $5 \leq N_1 < 25$  y  $25 < N_2 \leq 35$ :

```

"""Cálculo del límite líquido LL con el método de Lambe"""
import math as m
def Humedad(wh,ws,t):
    w=(wh-ws)/(ws-t)
    return w
def Lambe(N1,W1,N2,W2):
    LL=1/2*(W1*(N1/25)**0.121+W2*(N2/25)**0.121)
    return LL
def LeerDatos(i):
    print("Lectura de datos para variable no. ",i)
    n1=eval(input("Introduzca el número de golpes (5<N1<25) y (25<N2<35): "))
    w1=eval(input("Introduzca el peso inicial del suelo + agua (gr): "))
    w2=eval(input("Introduzca el peso final del suelo + agua (gr): "))
    tara=eval(input("Introduzca la tara (gr): "))
    agua=w1-w2
    humedad=Humedad(w1,w2,tara)
    print("Número de golpes: ",n1)
    print("Humedad del suelo: {:.2%}".format(humedad))
    print("Cantidad de agua: {:.3f}".format(agua)," gr")
    print()
    return humedad,n1
def LL():
    W=[]
    for i in (1,2):
        W.append(LeerDatos(i))
    LimLiq=Lambe(W[0][1],W[0][0],W[1][1],W[1][0])
    print("El Límite Líquido es: {:.3%}".format(LimLiq))
LL()

```

```

Lectura de datos para variable no. 1
Introduzca el número de golpes (5<N1<25) y (25<N2<35): 12
Introduzca el peso inicial del suelo + agua (gr): 178.8
Introduzca el peso final del suelo + agua (gr): 162.7
Introduzca la tara (gr): 25.7
Número de golpes: 12
Humedad del suelo: 11.75%
Cantidad de agua: 16.100 gr

```

```

Lectura de datos para variable no. 2
Introduzca el número de golpes (5<N1<25) y (25<N2<35): 33
Introduzca el peso inicial del suelo + agua (gr): 145.5
Introduzca el peso final del suelo + agua (gr): 138.4
Introduzca la tara (gr): 30.1
Número de golpes: 33
Humedad del suelo: 6.56%
Cantidad de agua: 7.100 gr

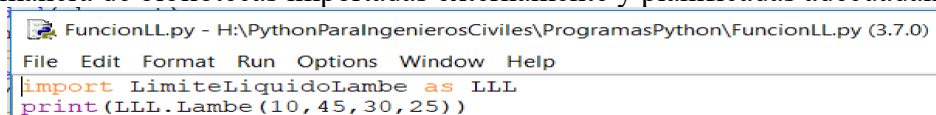
```

```

El Límite Líquido es: 8.766%

```

Una fortaleza del uso de funciones en un programa es que se pueden incluir y usar en otros programas a manera de bibliotecas importadas externamente y planificadas adecuadamente.



```

FuncionLL.py - H:\PythonParaIngenierosCiviles\ProgramasPython\FuncionLL.py (3.7.0)
File Edit Format Run Options Window Help
import LimiteLiquidoLambe as LLL
print(LLL.Lambe(10,45,30,25))

```