



FUNCIONES O MÉTODOS

Algoritmia y Programación

CONTENIDO

- ❑ Funciones/métodos
 - ❑ Definición
 - ❑ Partes de una función
 - ❑ Ejemplos
- ❑ Ámbito de variables
 - ❑ Variables Locales
 - ❑ Variables Globales

DEFINICIÓN

Una función es una porción de código (sub-algoritmo) que forma parte de un algoritmo principal, el cual se encarga de resolver una tarea específica.

DEFINICIÓN

Una función es una porción de código (sub-algoritmo) que forma parte de un algoritmo principal, el cual se encarga de resolver una tarea específica.

❑ Divide y vencerás: Es una estrategia para resolver problemas grandes, dividiéndolo en problemas más pequeños



DEFINICIÓN

Los métodos reciben datos del programa principal, realizan operaciones y le devuelven los resultados.

Es como una oficina: El programa principal es el jefe, que da instrucciones a sus subordinados (métodos), ellos realizan una tarea, y cuando terminan le devuelven el resultado y control al jefe.



VENTAJAS DE USAR FUNCIONES

- ❑ Los problemas pequeños son más fáciles de entender, de desarrollar y de mantener (localizar errores).
- ❑ Se evita código innecesario, pues los métodos se escriben una sola vez, y pueden ser utilizados (llamados) desde diferentes partes del programa, las veces que sea necesario

PROBLEMA

- ⦿ Desarrolle un programa que lea los datos de 2 catetos (a, b) y determine cual de los 2 es mayor y muestre su hipotenusa.

PROBLEMA

- ⦿ Desarrolle un programa que lea los datos de 2 catetos (a, b) y determine cual de los 2 es mayor y muestre su hipotenusa.

1. Análisis del problema

Entrada: a, b

PROBLEMA

- ⦿ Desarrolle un programa que lea los datos de 2 catetos (a, b) y determine cual de los 2 es mayor y muestre su hipotenusa.

1. Análisis del problema

Entrada: a, b

Salidas : hipotenusa, mayor

Proceso: ??

PROBLEMA

- ⦿ Desarrolle un programa que lea los datos de 2 catetos (a, b) y determine cual de los 2 es mayor y muestre su hipotenusa.

1. Análisis del problema

Entrada: a, b

Salidas : hipotenusa, mayor

Proceso: determinar mayor de 2 números
calcular hipotenusa

PROBLEMA

- ⊙ Desarrolle un programa que lea los datos de 2 catetos (a,b) y determine cual de los 2 es mayor y muestre su hipotenusa.

1. Análisis del problema

Entrada: a, b

Salidas : hipotenusa, mayor

Proceso: entero determinarMayor(a, b :entero) {

a y b son datos necesarios
para calcular el mayor de
los dos números

PROBLEMA

- ⊙ Desarrolle un programa que lea los datos de 2 catetos (a,b) y determine cual de los 2 es mayor y muestre su hipotenusa.

1. Análisis del problema

Entrada: a, b

Salidas : hipotenusa, mayor

Proceso: entero determinarMayor(a, b :entero) {

m: entero

SI $a > b$

m = a

SINO

m = b

retornar m

}

PROBLEMA

- ⦿ Desarrolle un programa que lea los datos de 2 catetos (a,b) y determine cual de los 2 es mayor y muestre su hipotenusa.

1. Análisis del problema

Entrada: a, b

Salidas : hipotenusa, mayor

Proceso: real calcularHipotenusa(a, b :entero)

PROBLEMA

- ⦿ Desarrolle un programa que lea los datos de 2 catetos (a,b) y determine cual de los 2 es mayor y muestre su hipotenusa.

1. Análisis del problema

Entrada: a, b

Salidas : hipotenusa, mayor

Proceso: real calcularHipotenusa(a, b :entero)

a y b son datos necesarios
para calcular la hipotenusa

PROBLEMA

- ⊙ Desarrolle un programa que lea los datos de 2 catetos (a,b) y determine cual de los 2 es mayor y muestre su hipotenusa.

1. Análisis del problema

Entrada: a, b

Salidas : hipotenusa, mayor

Proceso: real calcularHipotenusa(a, b :entero) {
 h: real
 $h = \sqrt{a^2 + b^2}$
 retornar h
}

PROBLEMA

- ⊙ Desarrolle un programa que lea los datos de 2 catetos (a,b) y determine cual de los 2 es mayor y muestre su hipotenusa.

1. Análisis del problema

Entrada: a, b

Salidas : hipotenusa, mayor

Proceso: real calcularHipotenusa(a, b :entero) {

h: real

$h = \sqrt{a^2 + b^2}$

retornar h

}

¿Cómo se escribe un método en Java?

LLAMADO DE MÉTODOS

- ⦿ Desarrolle un programa que lea los datos de 2 catetos (a,b) y determine cual de los 2 es mayor y muestre su hipotenusa
- 2. Diseñar el algoritmo y escribirlo en pseudocódigo

Inicio

a,b,mayor : **entero**
hipotenusa : **real**

leer (a)

leer (b)

hipotenusa = calcularHipotenusa(a,b)

mayor = determinarMayor(a,b)

imprimir(hipotenusa, mayor)

Fin

**La variable hipotenusa
va a tomar el valor que
le devuelva el método
calcularHipotenusa(a,b)**

FUNCIONES: EJEMPLO

```
def calcularHipotenusa(a, b):  
    h = math.sqrt( (a*a) + (b*b) )  
    return h
```

```
def calcularMayor (a,b):  
    if (a > b):  
        return a  
    else:  
        return b
```

Parámetros / argumentos

Hay funciones que reciben algunos valores para poder hacer operaciones con ellos. Si son varios se separan con comas .

Return: Las funciones *pueden* devolver un valor

FUNCIONES: EJEMPLO

```
a= int (input ("Digite el valor del cateto a: "))
b= int (input ("Digite el valor del cateto b: "))

#Llamado a las funciones
h= calcularHipotenusa (a,b)
m= calcularMayor(a,b)

print ("El valor de la hipotenusa es: ", h)
print ("El valor mayor es : ", m)
```

[Ver hipotenusa.py](#)

Funciones: Ejemplo

1. Declarando las funciones

```
def suma (x,y):  
    return x + y
```

```
def resta():  
    o1 = int(input("Digite un número"))  
    o2 = int(input("Digite un número"))  
    resta = o1 - o2  
    print ("Resta ", resta)
```

```
def multiplicacion (x,y):  
    m = x * y;  
    print (" X * Y = ", m)
```

- Las funciones pueden o no *retornar* un valor
- Las funciones pueden o no recibir *parámetros*

Funciones: Ejemplo

1. Declarando las funciones

```
def suma (x,y):  
    return x+y
```

```
def resta():  
    o1 = 20  
    o2 = 3  
    resta = o1-o2  
    print ("Resta ",resta)
```

```
def multiplicacion (x,y):  
    m = x * y;  
    print (" X * Y = ", m)
```

```
def division (x, y):  
    if y == 0:  
        print (" Error ... ")  
    else:  
        divi = x / y  
        print ("Division ", divi)
```

Funciones: Ejemplo

2. Llamando a las funciones

```
sum = suma (5,2)
print ("Suma: ", sum)

resta()
multiplicacion (20, 14)
division (10, 2)
```

Ver ejemplo funciones.py

EJERCICIO

⊙ Se desea reforestar un bosque que mide un número n de hectáreas. Si la superficie del terreno excede a $1'000.000\text{m}^2$, entonces se siembra así:

% superficie	Tipo de árbol
70%	Pino
20%	Oyamel
10%	Cedro

Si la superficie del terreno es menor o igual a $1'000.000\text{m}^2$ entonces se siembra así:

% superficie	Tipo de árbol
50%	Pino
30%	Oyamel
20%	Cedro

Se desea saber el número de pinos, oyameles y cedros que se deben sembrar, si se sabe que en 10m^2 caben 8 pinos; en 15m^2 caben 15 oyameles, y en 18m^2 caben 10 cedros.

EJEMPLO

1. Análisis del problema

Entrada: n

Salidas : cant_pinos , cant_oyamel , cant_cedro

Proceso: hallarCantidades

EJEMPLO

1. Diseñar el algoritmo y escribirlo en pseudocódigo

Inicio

n: **entero**

leer(n)

hallarCantidades(n)

Fin

EJEMPLO

1. Diseñar el algoritmo y escribirlo en pseudocódigo

hallarCantidades(n: entero){

cant_pino, cant_oyamel, cant_cedro, hectarea=10000 : entero

SI (n*hectarea) > 1000000

cant_pino= (((n*hectarea) *0.7)/10)*8

cant_oyamel = (((n*hectarea) *0.2)/15)*15

cant_cedro= (((n*hectarea) *0.1)/18)*10

SINO

cant_pino= (((n*hectarea) *0.5)/10)*8

cant_oyamel = (((n*hectarea) *0.3)/15)*15

cant_cedro= (((n*hectarea) *0.2)/18)*10

Imprimir (cant_pino, cant_oyamel, cant_cedro);

}

EJEMPLO: 1. Declaración de la función

```
def hallarCantidades (n):
```

```
    hectarea=10000
```

```
    if ( (n*hectarea) > 1000000):
```

```
        cant_pino= (((n*hectarea) *0.7)/10)*8
```

```
        cant_oyamel = (((n*hectarea) *0.2)/15)*15
```

```
        cant_cedro= (((n*hectarea) *0.1)/18)*10
```

```
    else:
```

```
        cant_pino= (((n*hectarea) *0.5)/10)*8
```

```
        cant_oyamel = (((n*hectarea) *0.3)/15)*15
```

```
        cant_cedro= (((n*hectarea) *0.2)/18)*10
```

```
print ("cantidad de Pinos", cant_pino, " cantidad de Oyamel",  
cant_oyamel, " cantidad de Cedro", cant_cedro)
```

EJEMPLO: 2. Llamando la función

```
cant= int ( input ("Ingrese el número de hectáreas:")) ;  
  
hallarCantidades(cant);
```

[Ver ejemplo.py](#)

ÁMBITO DE VARIABLES

El ámbito de una variable es el contexto (espacio) dentro del programa en donde ella puede ser utilizada y reconocida.

VARIABLES GLOBALES Y LOCALES

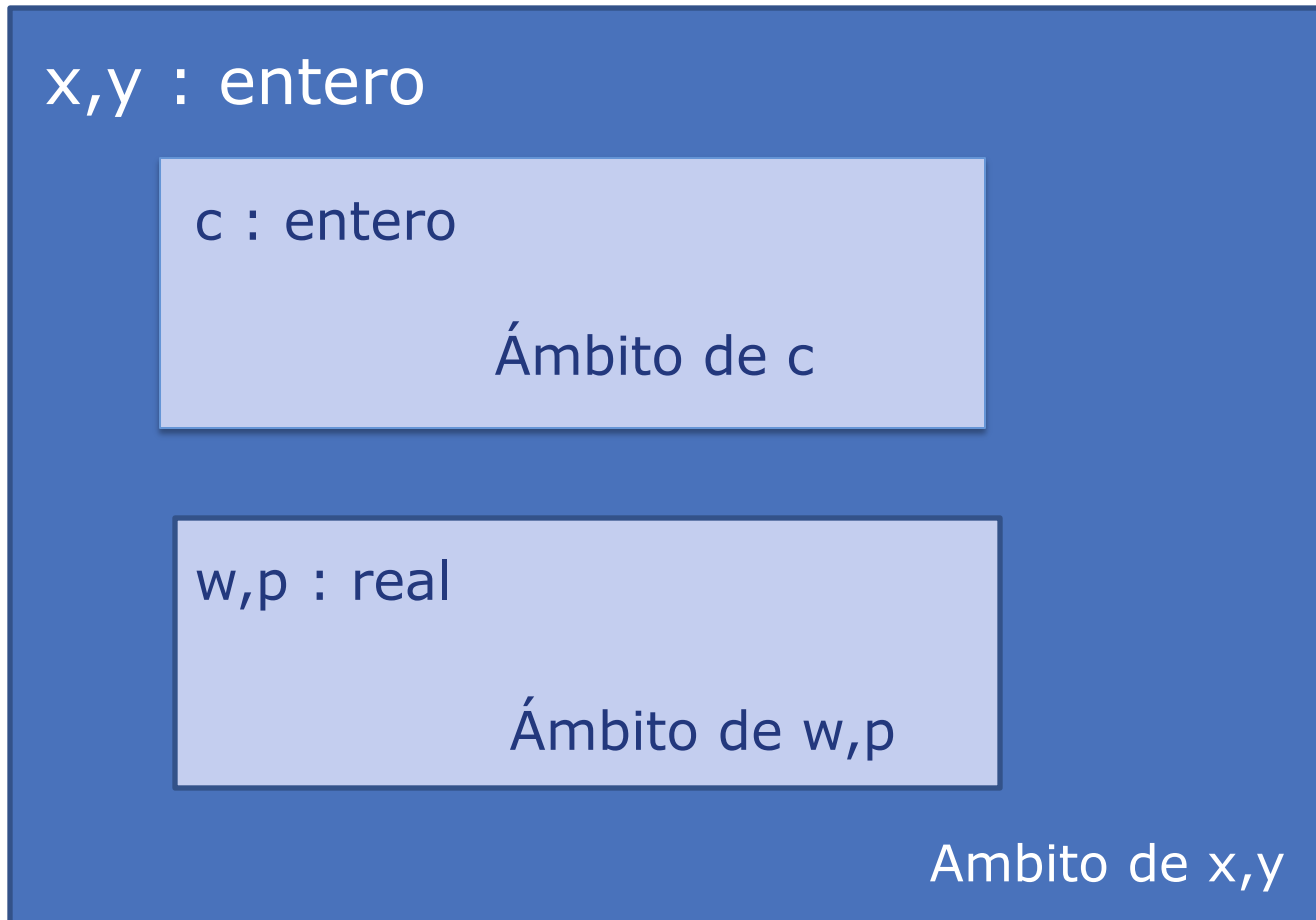
Las variables se clasifican en 2 tipos:

- Variables Locales
- Variables Globales
- **Variables Locales:** Son aquellas variables que se crean dentro de una función. Estas variables solamente son reconocidas dentro de esa función donde fueron creadas, y su valor se pierde al finalizar la ejecución de la misma.

VARIABLES GLOBALES Y LOCALES

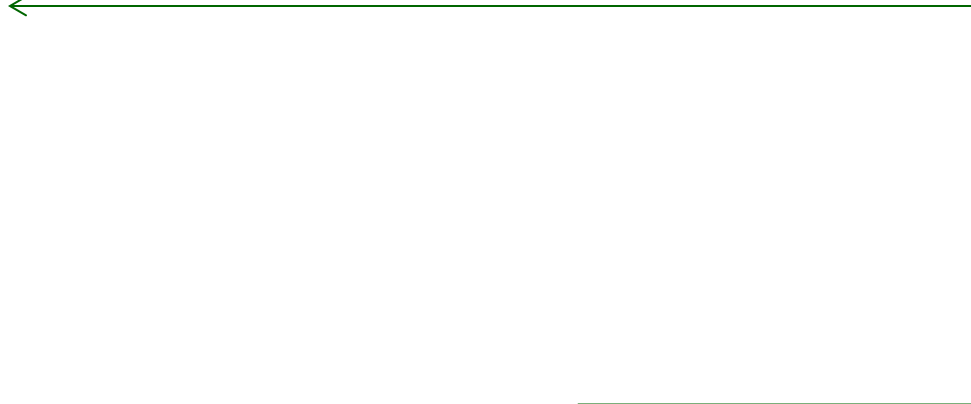
- **Variables Globales:** Son aquellas variables que se crean en el programa principal, por fuera de cualquier función. Éstas son reconocidas en todo lugar del programa.

VARIABLES GLOBALES Y LOCALES



VARIABLES GLOBALES Y LOCALES

```
def calcularHipotenusa(a, b):  
    h = math.sqrt( (a*a) + (b*b) )  
    return h
```



h es una Variable Local

EJEMPLO

- Desarrollemos el ejemplo de la reforestación, usando variables globales y métodos que no reciben parámetros

EJEMPLO

No se reciben parámetros

```
def hallarCantidades():
```

```
    hectarea=10000
```

```
    if ( (n*hectarea) > 1000000):
```

```
        cant_pino= (((n* hectarea) *0.7)/10)*8
```

```
        cant_oyamel = (((n* hectarea) *0.2)/15)*15
```

```
        cant_cedro= (((n* hectarea) *0.1)/18)*10
```

```
    else:
```

```
        cant_pino= (((n* hectarea) *0.5)/10)*8
```

```
        cant_oyamel = (((n* hectarea) *0.3)/15)*15
```

```
        cant_cedro= (((n* hectarea) *0.2)/18)*10
```

```
    print ("cantidad de Pinos", cant_pino, " cantidad de Oyamel", cant_oyamel,  
          " cantidad de Cedros, cant_cedro)
```

EJEMPLO

```
n = int ( input ("Ingrese el número de hectáreas:"))  
  
hallarCantidades();
```

La variable ***n*** es ***global*** porque se definió en el programa principal. Esta variable es ***visible*** dentro de cualquier función.

Ver Ejemplo Reforestación.py

EJERCICIOS

- ⦿ Usando funciones, desarrollar un programa que dado el peso, la altura y el sexo de un estudiante. Determine la cantidad de vitaminas que debe consumir, con base en los siguientes criterios:

Si es hombre, y su estatura es mayor a 1.60, y su peso es mayor o igual a 75 kilos, su dosis, será: 20% de la estatura y 80% de su peso, si la estatura es menor o igual a 1.60, la dosis será la siguiente: 30% de la estatura y 70% de su peso.

Si es mujer, y su estatura es mayor o igual a 1.55 y su peso es mayor o igual a 65 kilos, su dosis será: 25% de la estatura y 75% de su peso. Si el peso es menor a 65 kilos, será: 35% de la estatura y 65% de su peso.

EJERCICIOS

- Un comerciante se dedica a la venta de sillas. Vende tres tipos de sillas:

Tipo	Precio
1	\$5.000
2	\$7.000
3	\$10.000

Por cada cinco sillas compradas se obtiene un descuento, de acuerdo a la tabla

Tipo	Descuento
1	3%
2	5%
3	10%

El resto de sillas se cobran a precio normal. Diseñe un programa que lea el tipo de silla y la cantidad a comprar e imprima la cantidad, el precio unitario, el descuento y el precio total, de lo que debe cancelar el cliente por la compra.

Nota: El almacén sólo vende UN tipo de silla a cada cliente.