
Semana # 1 FUNCIONES

Hasta el día de hoy, todos nuestros programas los hemos realizado dentro de la función `int main()`.

A partir de ahora vamos a trabajar con funciones totalmente fuera del `main()` con la finalidad de no saturar la función como tal y hacer que nuestro programa se vea más elegante y ordenado.

¿Qué es una función?

Una función realiza una tarea concreta y puede ser diseñada, implementada y depurada de manera independiente al resto del código. Cada función realiza una sola tarea.

La declaración básica para la definición de una función en C++ es la siguiente:

```
Tipo nombre(tipo var1,tipo var2,...,tipo var n){  
Conjunto de instrucciones  
}
```

El tipo de dato (`int`, `char`, `float`, `bool`, `void`,....), también será el tipo de dato que me va a devolver dicha función.

Por ejemplo:

```
int numMax(int x, int y)  
  
double intercambio(double x, double y)  
  
void desplegar(float x, float y)
```

Como una regla de buena programación, el siguiente ordenamiento de instrucciones deberán formar la estructura básica alrededor de la cual se construyan todos sus programas en C++.

directivas del preprocesador (`# <Librerias>`)

prototipos de función (Es la forma en que le decimos al programa que esa función existe)

```
int main(){  
  
conjunto de instrucciones  
  
return valor  
}
```

Definiciones de función

```
Ejercicio 1.cpp ➤ X
Ejercicio 1 (Ámbito global)
#include<iostream>
#include<stdlib.h>
using namespace std;

//Prototipo de funcion
int encontrarMax(int x, int y);

int main() {
    int num1, num2;
    int mayor;
    cout << "Digite 2 numeros";
    cin >> num1 >> num2;

    mayor = encontrarMax(num1, num2);
    cout << "El mayor de los numeros es:" << mayor << endl;

    system("pause");
    return 0;
}

int encontrarMax(int x, int y) {
    int numMax;
    if (x > y) {
        numMax = x;
    }
    else {
        numMax = y;
    }
    return numMax;
}
```

```
Ejercicio2 (Ámbito global) main()
/*Ejercicio 2: Escriba una función llamada mult() que acepte dos números en punto
flotante como parámetros, multiplique estos dos números y despliegue el resultado.*/

#include<iostream>
#include<stdlib.h>
using namespace std;

//Prototipo de Funcion
void mult(float x, float y);
void pedirDatos();

float n1, n2;

int main() {
    pedirDatos();
    mult(n1, n2);

    system("pause");
    return 0;
}

void pedirDatos() {
    cout << "Digite 2 numeros: ";
    cin >> n1 >> n2;
}

void mult(float x, float y) {
    float multiplicacion = x * y;

    cout << "La multiplicacion es: " << multiplicacion << endl;
}
```

```
Ejercicio3.cpp X
Ejercicio3 (Ámbito global)

/*Ejercicio 3: Escriba una función llamada al_cuadrado() que calcule el cuadrado del
valor que se le transmite y despliegue el resultado. La función deberá ser capaz de
elevar al cuadrado números flotantes*/

#include<iostream>
#include<stdlib.h>
using namespace std;

void pedirDatos();
void al_cuadrado(float n);

float numero;

int main() {
    pedirDatos();
    al_cuadrado(numero);

    system("pause");
    return 0;
}

void pedirDatos() {
    cout << "Digite un numero: ";
    cin >> numero;
}

void al_cuadrado(float n) {
    float cuadrado = 0;

    cuadrado = numero * numero;

    cout << "El cuadrado del numero es: " << cuadrado << endl;
}
```

```
Ejercicio4.cpp X
Ejercicio4 (Ámbito global)

/*Ejercicio 4 Escriba una función nombrada tiempo() que tenga un parámetro en número
entero llamado totalSeg y tres parámetros de referencia enteros nombrados horas, min
y seg. La función es convertir el número de segundos transmitido en un número
equivalente de horas, minutos y segundos.*/

#include<iostream>
#include<stdlib.h>
using namespace std;

void tiempo(int, int&, int&, int&);

int main() {
    int totalSeg, horas, min, seg;

    cout << "Digita la cantidad de segundos: ";
    cin >> totalSeg;

    tiempo(totalSeg, horas, min, seg);

    cout << "\nHoras: " << horas << endl;
    cout << "Minutos: " << min << endl;
    cout << "Segundos: " << seg << endl;

    system("pause");
    return 0;
}

void tiempo(int totalSeg, int& horas, int& min, int& seg) {
    horas = totalSeg / 3600;
    totalSeg %= 3600;
    min = totalSeg / 60;
    seg = totalSeg % 60;
}
```