
Semana # 11 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 mas 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 funcion

```
/*Ejemplo encontrar el valor de dos numeros.*/
```

```
#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;
}
```

```
//Definición de la función
```

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

```
/*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;  
}
```

```
/*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;
}
```

```
/*Escriba una función nombrada funpot() que eleve un número entero que
se le transmita a una potencia en número entero positivo y despliegue el resultado.
El número entero positivo deberá ser el segundo valor transmitido a la función.*/

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

void pedirDatos();
void funpot(int x, int y);

int numero, exponente;

int main() {
    pedirDatos();
    funpot(numero, exponente);

    system("pause");
    return 0;
}

void pedirDatos() {
    cout << "Digite el numero a elevar: ";
    cin >> numero;
    cout << "Digite el exponente de elevacion: ";
    cin >> exponente;
}

void funpot(int x, int y) {
    long resultado = 1;

    for (int i = 1; i <= y; i++) {
        resultado *= x;
    }

    cout << "El resultado de la elevacion es: " << resultado << endl;
}
```

```
/*Escriba un programa en C++ que devuelva la parte fraccionaria de
cualquier número introducido por el usuario. Por ejemplo, si se introduce el número
256.879, debería desplegarse el número 0.879.*/

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

void pedirDatos();
float devolucionFraccionaria(float n);

float numero;

int main() {
    pedirDatos();

    cout << "La parte fraccionaria del numero es: " << devolucionFraccionaria(numero) << endl;

    system("pause");
    return 0;
}

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

float devolucionFraccionaria(float n) {
    //por ejemplo si el numero es 45.567 en entero solo se guardaria 45
    int entero = n;
    //y por ultimo restamos el valor flotante menos el entero para tener la parte fraccionaria
    float resultado = n - entero;

    return resultado;
}
```