Funciones - Parametros por valor

- Los parametros se pasan a las funciones por valor, es decir, una copia de los valores originales son enviados.

Definicion

```
tipo_de_retorno nombre(listado_de_parametros){
    declaraciones
    sentencias
}
```

- El `tipo-de-retorno` es el tipo del valor que retorna la función.
  - El tipo void esusado para indicar que la función no retorna ningún valor.
- `Nombre` que le damos a la función y que usaremos en la invocación a la misma
- `Listado de parámetros`: son los valores que se pasan como entrada a la función, separados por comas.
- Luego, entre llaves, el cuerpo de la función (d`eclaraciones y sentencias`)
- `return`: para devolver un resultado al punto de llamada.

Invocacion o llamada

La invocación de una función se hace empleando el nombre de la función junto con los parámetros actuales, si los tuviera. En caso de no tener parámetros, simplemente se colocará el nombre seguido de ().

Una función puede ser invocada en el programa:

1. Dentro de una expresión del mismo tipo retornado por la función.
2. En una sentencia, cuando el tipo de retorno de una función es void (no retorna ningún valor).

Función Separador.

```
// Función SEPARADOR que no recibe nada y no devuelve nada:
#include<iostream.h>
#include<conio.h>
// Definición de la Función Separador
void separador()
{
cout<<endl<<"****************************************"<<endl;
}
int main(){
separador();
cout<<" *** FACULTAD DE INFORMATICA ***";
separador();
cout<<" *** PROGRAMACION II ***";
separador();
```

```
cout<<" *** Tema del dia: FUNCIONES ***";
separador();
// llamada a la Función Separador, constituye por sí misma una sentencia.
getch();
return(0);
}
```

Pasaje de Parámetros

- En la definición de la Función, se define cuántos y de qué tipo son los datos que espera recibir cuando ella sea invocada y con qué nombres se van a referenciar a esos en el cuerpo de la función, éstos son los parámetros formales.
- Al "llamar" a una Función, habrá que pasarle los valores con los que ésta trabajará: uno por cada parámetro formal declarado. Estos valores se conocen como parámetros actuales.

Función con 2 Parámetros y que devuelve un resultado

```
// Función para sumar dos números
int suma (int a, int b)
{
    int c;
    c = a + b;
    return c;
}
```

Programa que Calcula el perímetro de un rectángulo

```
// Usa Función DOBLE
#include<iostream.h>
#include<conio.h>

//Funcion
float doble ( float valor )
{
    float resu;
    resu = valor*2;
    return (resu);
}

int main(){
    float largo;
    float ancho;
    float perimetro;

    cout<<"Ingrese el largo: ";
    cin>>largo;
    cout<<endl;
    cout<<"Ingrese el ancho: ";
```

```
    cin>>ancho;

        perimetro = doble(largo) + doble(ancho);
        cout<<endl <<"El perimetro es: "<<perimetro;

    getch();
    return(0);
}
```

```
// Calcular el perímetro de un rectángulo
// la Función Perimetro llama a la Función Doble
#include<iostream.h>
#include<conio.h>

//Funciones
float doble ( float valor )
{
    float resu;
    resu = valor*2;
    return (resu);
}
float perimetro(float largo, float ancho)
{
    float contorno;
    contorno = doble(largo) + doble(ancho);
    return(contorno);
}

int main(){
    float largo;
    float ancho;

    cout<<"Ingrese el largo: ";
    cin>>largo;
    cout<<endl<<"Ingrese el ancho: ";
    cin>>ancho;
    cout<<endl<<"El perimetro es: "<< perimetro(largo,ancho);

    getch();
    return(0);
}
```

Alcance o Ámbito de una variable

Una variable declarada en una función, es LOCAL a la función, fuera de ella no puede ser usada.

- Variables locales se crean y destruyen cada vez que se ejecuta la función.

Por otro lado, las variables globales, son las que se definen por fuera de las funciones y pueden ser accedidas desde cualquier parte del programa.

Prototipo de una funcion

1. Se desea usar una función antes de definirla
2. La función se encuentra en la biblioteca de funciones, la manera de informar al compilador:
    - el tipo de resultado
    - cantidad de parámetros
    - tipos de parámetros

Nota: no hace falta informarle los nombres de los parámetros.

El Prototipo de la Función, incluye la definición de la cabecera de la función, omitiendo el cuerpo de ésta y terminando en ";".

Programa que lee un año y muestra si es o no bisiesto

```
// Un año es bisiesto cuando es (divisible por 4 y no por 100) o (es
divisible por 400).
#include <iostream.h>
#include <conio.h>

//Prototipo de la funcion
int bisiesto(int);

int main(){
    int anio;
    //Pedir al usuario el valor del anio
    cout << "Ingrese el anio: ";
    cin >> anio;

    //Llamada a la funcion
    if(bisiesto(anio)){
        cout << "Es bisiesto";
    }else {
        cout << "No es bisiesto";
    }

    getch();
    return 0;
}

//Funcion
int bisiesto(int a) //definición de la función
{
    if ( a%4==0 && a%100!=0 || a%400==0)
    return 1;
    else
    return 0;
}
```