

# Funciones

# Concepto básico de función

- El objetivo básico de las funciones es dividir el programa en pequeños fragmentos (bloques) especializados de código

```
int main()  
{  
    imprimir_mensaje_intro();  
    obtener_datos_entrada();  
    efectuar_proceso();  
    imprimir_datos_salida();  
  
    return 0;  
}
```

# Ventajas de utilizar funciones

1. Código más organizado, con bloques más pequeños, a modo de resumen de funcionalidad
2. Permiten separar el desarrollo de grandes proyectos, asignando las distintas partes a diferentes desarrolladores de un equipo
3. Permiten **reutilizar** código (llamadas desde varios puntos del programa)
4. Incluso pueden reutilizarse funciones de uso común en diferentes proyectos

# Formato de una función

- Sintaxis:

```
return_type function_name( formal_param_list )  
{  
    local variable declarations  
    statements  
    return value_of_return_type;  
}
```

# Ejemplo 1

```
int main(void)
{
    char quit;

    do
    {
        cout << "Loop again? ";
        cin >> quit;
    } while (quit != '\n');

    return 0;
}
```

# Ejemplo 2

```
void greetings(void);
```

```
int main(void)
{
    greetings();
    return 0;
}
```

```
void greetings(void)
{
    cout << "Hi there, welcome to my world." << endl;
    return;
}
```

# Ejemplo 3

```
void display_number(int a_number)
{
    cout << a_number << endl;
    return;
}

int main(void)
{
    int bob = 7;
    display_number(bob);
    return 0;
}
```

# Ejemplo 4

```
int get_number(void);

int main(void)
{
    int bob;
    bob = get_number();
    return 0;
}

int get_number(void)
{
    int value_to_get;
    cout << "Enter an integer: ";
    cin >> value_to_get;
    return value_to_get;
}
```



# Ejemplo 5

```
int get_number(void);
float calculate_average(const float val1, const float val2);

int main(void)
{
    float bob, sue, average;
    bob = get_number();
    sue = get_number();
    average = calculate_average(bob, sue);
    return 0;
}

float calculate_average(const float val1, const float val2)
{
    float avg;
    avg = (val1 + val2) / 2;
    return avg;
}
```

# Ejemplo 6

```
const float PI = 3.14159;
float cylinder_volume(const float, const float);

int main(void)
{
    cout << cylinder_volume(3, 5) << endl;
    cout << radius << endl;
    return 0;
}

float cylinder_volume(const float radius, const float height)
{
    return PI * radius * radius * height;
}
```

# Importante!

- Cada función debería encargarse de realizar ***una y solamente una*** tarea específica

```
int num, root;

do
{
    cout << "enter non-negative number: " << endl;
    cin >> num;

} while (num < 0);

root = square_root(num);
```