



# Bienvenidos al taller de C++

Introducción a C++  
Rony Hanco

Siguiente página →

# Agenda

- struct

User - Defined  
Datatype

**struct Books**

```
{  
char title[50];  
char author[50];  
char subject[100];  
int book_id;  
}book;
```

# struct

## Plain-Old-Data Classes POD

- Los PODs son simples contenedores.
- Piensa en ellos como una especie de matriz heterogénea de elementos de tipos potencialmente diferentes.
- Para acceder a cada miembro de la estructura se usa el punto(.).

```
#include<iostream>
using namespace std;
struct Estudiante{
    char codigo[9];
    char nombre[100];
    int ciclo_relativo;
    int creditos;
    bool matriculado;
};
int main(){
    Estudiante rony;
    rony.ciclo_relativo = 10;
    rony.creditos = 235;
    cout << rony.ciclo_relativo << endl;
    return 0;
}
```

# Union

- Es similar POD, a diferencia de él coloca a todos sus miembros en el mismo lugar.

```
#include<iostream>
using namespace std;
union Peso {
    char cadena[10];
    int entero;
    double punto_flotante;
};
int main(){
    Peso rony;
    rony.cadena[0] = '2';
    rony.cadena[1] = '0';
    rony.cadena[2] = '0';
    rony.entero = 200;
    rony.punto_flotante = 200.00;
    cout << rony.cadena << endl;
    cout << rony.entero << endl;
    cout << rony.punto_flotante << endl;
    return 0;
}
```



# struct con características de clases

- Métodos: Parecidas a las funciones declaradas de forma interna, pueden usar los miembros de la estructura.
- Control de acceso: Cada miembro puede ser declarado público o privado (por defecto público-private).
- Constructor: Es un método con el mismo nombre de la estructura(no retorna valor) y permite inicializar las variables internas.
- Clase: se puede usar "class" a cambio de "struct" haciendo que las variables internas sean por defecto privadas.

```
#include<iostream>
using namespace std;
struct Pokemon {
    string nombre;    string tipo;
    int ps;    int ataque;    int defensa;
    Pokemon(string nombre, string tipo, int ps, int ataque, int defensa) {
        this->nombre = nombre;
        this->tipo = tipo;
        this->ps = ps;
        this->ataque = ataque;
        this->defensa = defensa;
    }
    int get_nivel(){
        return this->nivel;
    }
    void set_nivel(int nivel){
        this->nivel = nivel;
    }
private:
    int nivel;
};

int main(){
    Pokemon caterpie("Caterpie","Insecto",3,2,3);
    cout << caterpie.nombre << endl;
    cout << caterpie.ataque << endl;
    cout << caterpie.get_nivel() << endl;
    return 0;
}
```

# Práctica

## Uso de struct

- Defina un struct para un triángulo con métodos para calcular el perímetro y área.
- Almacene y muestre los datos de una persona: edad, peso, nombre.