<C++>

Struct | enum

Dorian.H Mekni

15 | APRIL | 2020

Struct -> Rappel

- Les structures peuvent intégrer des variables, des tableaux ainsi que des fonctions
- Mettre les membres qui prennent le plus de place en mémoire en premier
- l'allocation mémoire se construit de manière continue comme un empilement, d'où l'idée de placer en premier les données qui nécessitent le plus d'espace mémoire ne premier

C++

struct | Architecture

```
struct  Persona
        {

std:: string name;
      int age;


        };
```

```
int main()
      {

/* struct */ Persona p{ };

      return 0;

      }
```

Il n’est pas obligatoire en C++ de faire appel à un typedef pour licencier le préfixe de type struct

C++

struct | usage

Il n’est pas obligatoire en C++ de faire appel à un typedef pour licencier le préfixe de type struct

```cpp
#include <iostream>


struct Persona
{
    std:: string name;
    int age;
};



int main()
{

    Persona p{"Dorian", 36};

    std::cout << "His name is " << p.name << std::endl;

    return 0;

}

```

```
His name is Dorian
Program ended with exit code: 0
```

C++

struct | usage -> options

Une autre manière de déclarer un attribut | struct

```cpp
#include <iostream>


struct Persona
{
    std:: string name;
    int age;
};



int main()
{

    Persona p{};

    p.name = "Dorian";

    std::cout << "His name is " << p.name << std::endl;

    return 0;

}
```

```
His name is Dorian
Program ended with exit code: 0
```

C++

struct | doublon

Créer un doublon en plaçant le dénomination
de la variable créée entre les bracelets de la nouvelle.

```cpp
#include <iostream>


struct Persona
{
    std:: string name;
    int age;
};



int main()
{

    Persona p{"Dorian", 36};

    Persona p2{p};


    std::cout << "His name is " << p.name << std::endl;
    std::cout << "His name is " << p2.name << std::endl;

    return 0;

}

```

```
His name is Dorian
His name is Dorian
Program ended with exit code: 0
```

C++

struct | using

Le typedef est remplacé par using en C++ et il s'utilise de la manière la suivante :

```cpp
#include <iostream>

using type = struct Persona;

struct Persona
{
    std:: string name;
    int age;
};



int main()
{

    Persona p{"Dorian", 36};

    type p2{p};


    std::cout << "His name is " << p.name << std::endl;
    std::cout << "His name is " << p2.name << std::endl;

    return 0;

}
```

```
His name is Dorian
His name is Dorian
Program ended with exit code: 0
```

C++

enum | ->

The enum in C++ se construit sur le modèle hérité du C et
peut s'incorporer au sein d'une structure
et plus tard réutilisée dans notre fonction main()comme c'est le cas dans notre example ->

```cpp
#include <iostream>

enum Bool
{
    FALSE,// = 0
    TRUE // = 1
};

struct Guessing
{
    std:: string question;
    Bool b;

};

int main()
{

    Guessing newQuestion{"Are you a happy?", Bool::TRUE};

        if(newQuestion.b == Bool::TRUE)
            std::cout << "Yes, I'm happy !" << std::endl;
        else
            std::cout << "so so." << std::endl;
    return 0;

}
```

```
Yes, I'm happy !
Program ended with exit code: 0
```

C++