

Programmation C

15 - Types personnalisés

Les types de base

Type	Taille (octets)	Valeur min	Valeur max	Affichage
unsigned short int / unsigned short	2	0	65535	%d
short int / short	2	-32768	32767	%d
unsigned long int / unsigned long	4	0	4294967295	%ld
long int / long	4	-2147483648	2147483647	%ld
unsigned int	2 / 4	0 / 0	65535/ 4294967295	%d
int	2 / 4	-32768 / -2147483648	32767 / 2147483647	%d
unsigned char	1	0	255	%c
signed char	1	-128	127	%c
char	1	-128 / 0	127 / 255	%c
float	4	-3.4e38	3.4e38	%f
double	8	-1.7e308	1.7e308	%f

Les enums

enum

NOM_ENUM

{

CLE

valeur numérique

,

CLE

valeur numérique

,

...

...

};

si pas de valeurs alors suite numérique partant de 0

Défini dans le .h

```
enum REPONSE
```

```
{
```

```
    OUI,  
    NON
```

```
};
```

```
enum NIVEAU
```

```
{
```

```
    FACILE 1,  
    MOYEN 2,  
    DIFFICILE
```

```
};
```

```
int main()
```

```
{
```

```
    printf("%d \n", OUI);
```

```
    printf("%d", MOYEN);
```

```
}
```

0
2

Exemple

Créer une structure

struct

st_nom

{

liste de variables ;

};

Déclaration de structure en dehors des fonctions

Définie dans le .h

```
struct st_personnage
```

```
{
```

```
    char nom[100];
```

```
    int age;
```

```
    float taille;
```

```
};
```

```
struct st_point_2d
```

```
{
```

```
    int x;
```

```
    int y;
```

```
};
```

Exemple

Utiliser les structures

Déclarer une variable

```
struct nom_structure nom_variable = {valeurs} ;
```

Accès aux éléments d'une structure

```
nom_variable.nom_element;
```

```
struct st_personnage
{
    char nom[100];
    int age;
};

int main()
{
    struct st_personnage ned = {"Ned Stark", 36};
    struct st_personnage khaleesi;

    khaleesi.age = 16;
    strcpy( khaleesi.nom, "Daenerys Targaryen");
}
```

Exemple

typedef et structure

struct

st_nom

{

liste de variables

};

typedef struct

st_nom

nom type

typedef struct

st_nom

{

liste de variables

}

nom type

;

Exemple

```
typedef struct st_personnage
{
    char nom[100];
    int age;
} Personnage;
```

```
typedef struct st_personnage perso
```

```
int main()
{
    struct st_personnage ned = {"Ned Stark", 36};
    Personnage khaleesi;

    strcpy( khaleesi.nom, "Daenerys Targaryen");
    khaleesi.age = 16;
}
```

Typedef et enums

typedef enum

NOM

{

CLE

valeur

,

CLE

valeur

,

...

...

,

}

nom type ;

si pas de valeurs alors suite numérique partant de 0

Défini dans le .h

```
typedef enum JOURS
{
    LUNDI 1,
    MARDI 2,
    MERCREDI 3,
    JEUDI 4,
    VENDREDI 5,
    SAMEDI 6,
    DIMANCHE 7,
} Jours;
```

```
int main()
{
    Jours today = SAMEDI;

    printf("%d", today);
    printf("%d", LUNDI);
}
```

Exemple

6
1

Includes multiples

contenu

```
#include "module"  
#include "module"  
#include "module"
```

contenu

contenu

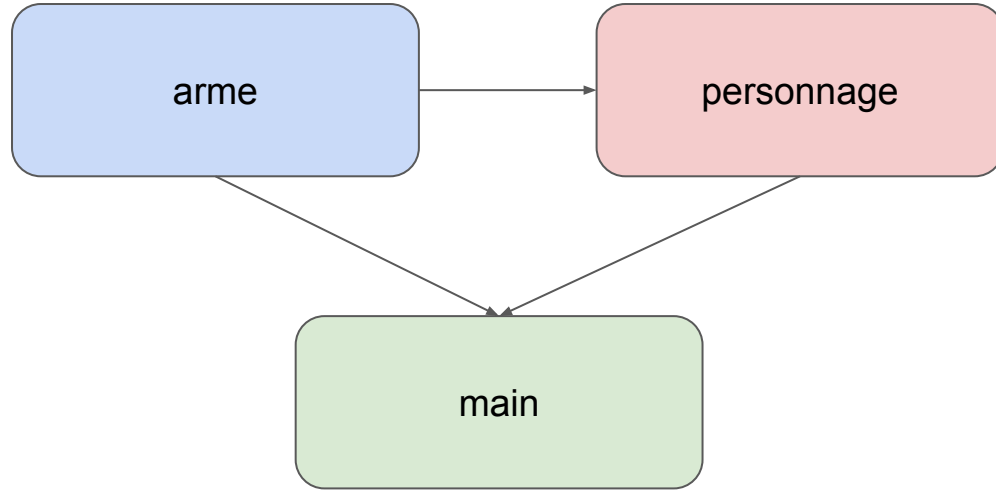
contenu

```
#ifndef MODULE_H_INCLUDED  
#define MODULE_H_INCLUDED  
    contenu  
#endif // MODULE_H_INCLUDED
```

```
#include "module"  
#include "module"  
#include "module"
```

contenu

Includes multiples



structures et pointeurs

Annotation pointeur classique:

`(*pointeur).variable`

=

Annotation simplifié:

`pointeur->variable`

Exemple

```
typedef struct st_personnage
{
    int taille;
    int age;
}Personnage;

int main()
{
    Personnage ned;
    Personnage* perso = &ned;

    (*perso).taille = 180;
    perso->age = 36;
}
```