

Programmation C

Conclusion

Du code au programme

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int vie = 3;
    int game_over = 0;

    InitialiserLeJeu();
    AfficherMenuDemarage();
    ...
}
```

Code source



Programme exécutable

Le compilateur

**Langage Humainement
compréhensible**

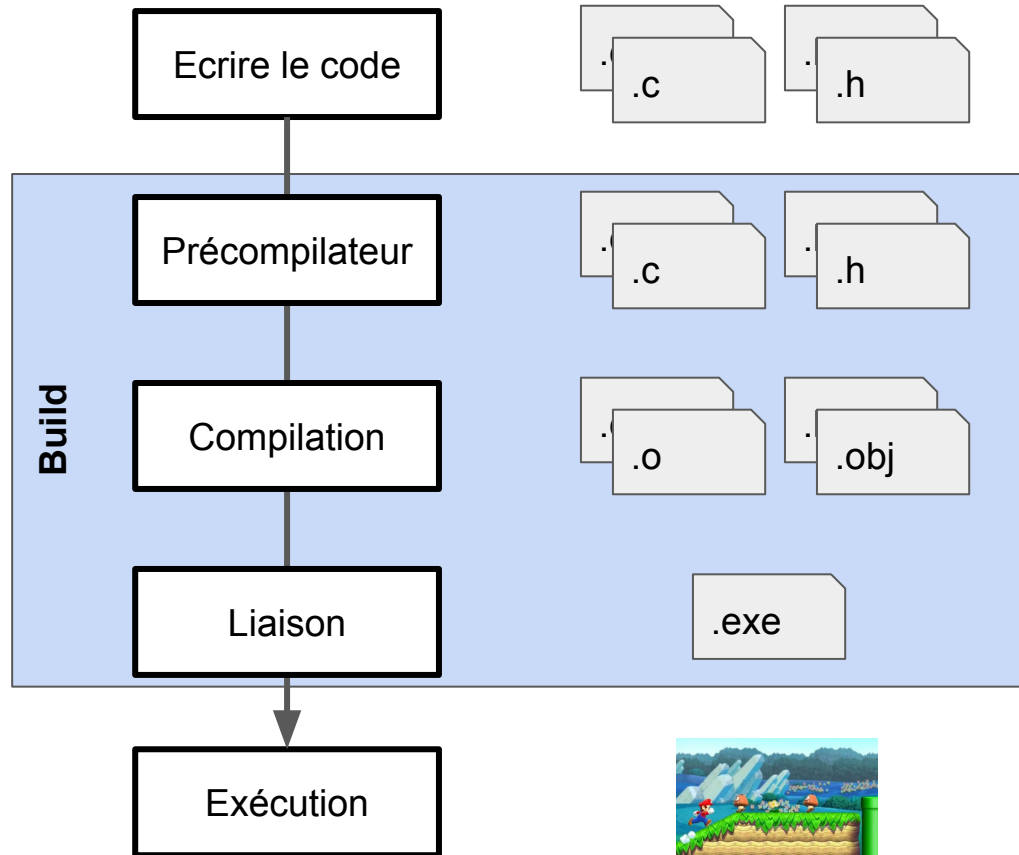
exemple:
Afficher "Hello" à l'écran

Compilateur

**Langage binaire compris
par les ordinateurs**

exemple:
011110011101101011100011

Création d'un Logiciel



Exécuter un programme

Disque dure

mon_programme.exe



Chargement

RAM (mémoire vive)

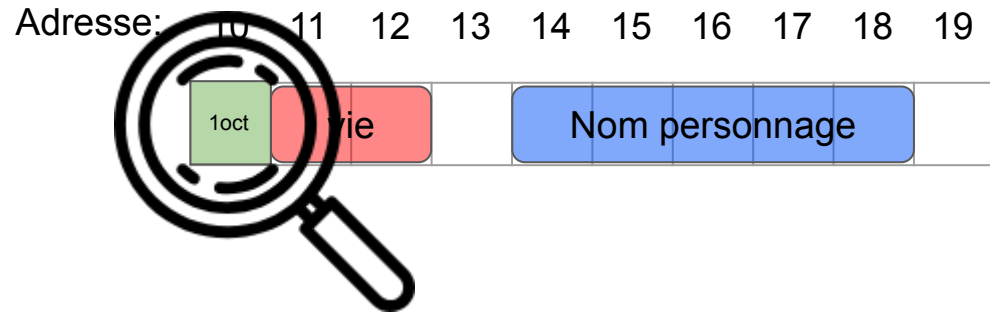
mon programme

autre programme



Gestion de la mémoire

MÉMOIRE



1oct

=

8 bits

ex: 1001 0111

Déclaration de variable

type

nom_de_variable

=

valeur

;

Exemple:

```
int ma_variable = 0;
```

```
int vie = 3;
```

ram



Les types short et long

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 conditions

Symbole	Signification	Exemple
==	est égale à	(password == 1852)
!=	est différent de	(password != 1852)
<	est inférieur à	(age_utilisateur < 18)
>	est supérieur à	(age_utilisateur > 18)
<=	est inférieur ou égale à	(age_utilisateur <= 0)
>=	est supérieur ou égale à	(age_utilisateur >= 0)

retourne 1(vrai) si la condition est respectée, sinon retourne 0(faux)

if, else if et else

```
if( condition )  
{  
    instructions  
}  
else if ( condition )  
{  
    instructions  
}  
...  
else  
{  
    instructions  
}
```

Structure

```
if(note == 0)  
{  
    printf("Null! \n");  
}  
else if(note < 10)  
{  
    printf("Peut mieux faire \n");  
}  
else if(note < 15)  
{  
    printf("Bien\n");  
}  
else  
{  
    printf("Tres bien! \n");  
}
```

exemple

note = 3
Peut Mieux faire

note = 0
Null!

note = 15
Tres bien !

structure switch

```
switch( variable )  
{  
    case valeur :  
        instructions  
        break;  
  
    ...  
  
    default :  
        instructions  
        break;  
}
```

structure

```
int numero_mois = 9;  
  
switch(numero_mois)  
{  
    case 1:  
        printf("Janvier\n");  
        break;  
    ...  
  
    case 12:  
        printf("Decembre\n");  
        break;  
  
    default:  
        printf("numero de mois invalide\n");  
        break;  
}
```

exemple

Septembre

La boucle for

```
for( init ; condition ; modification )  
{  
    instructions  
}
```

Structure

```
for(int var=0; var<5; var++)  
{  
    printf("%d\n", var);  
}
```

Exemple

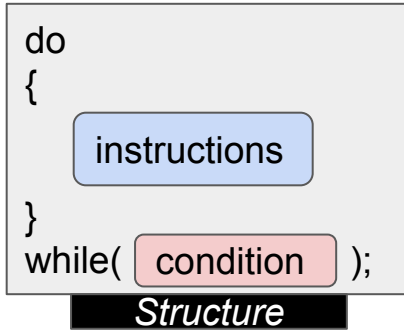
0
1
2
3
4

```
int var = 0;  
  
while( var < 5 )  
{  
    printf("%d\n", var);  
    var ++;  
}
```

Équivalent

0
1
2
3
4

La boucle do while

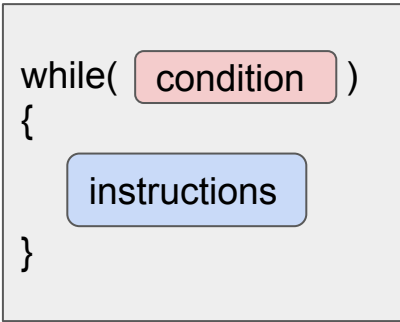


```
int var = 6;
do
{
    printf("%d\n", var);
    var ++;
}
while( var < 5 );
```

6

Exemple

La boucle while



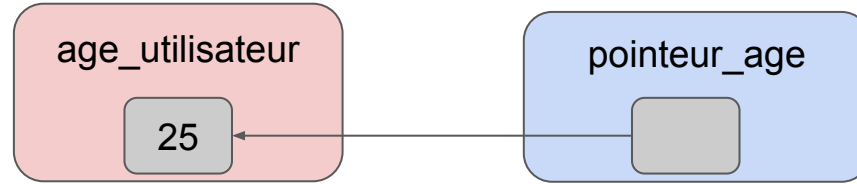
Structure

```
int var = 0;
while( var < 5 )
{
    printf("%d\n", var);
    var ++;
}
```

0
1
2
3
4

Exemple

Les pointeurs



Adresse: 10 11 12 13 14 15 16 17 18 19



ram

Déclaration de pointeur

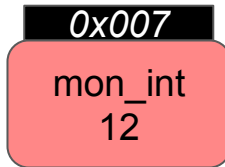
type * nom_de_pointeur = valeur ;

Exemple:

```
char* p_char = 0;  
int* p_int = NULL;
```

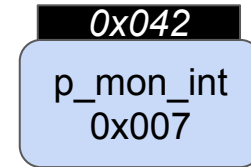


Utilisation des pointeurs



mon_int (contenue de mon_int)
-> 12

&mon_int (adresse de mon_int)
-> 0x007

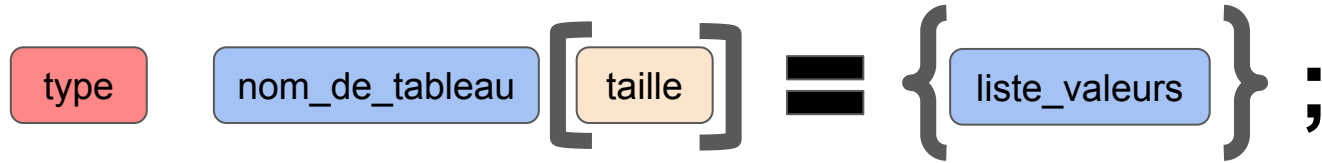


&p_mon_int (adresse de p_mon_int)
-> 0x042

p_mon_int (contenue de p_mon_int)
-> 0x007

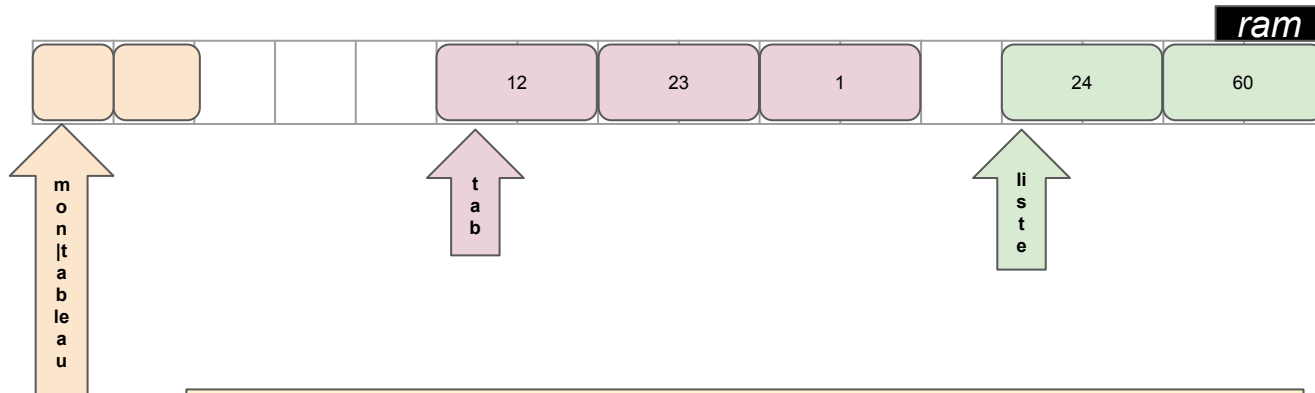
***p_mon_int (contenue de 0x007)**
-> 12

Déclaration Tableaux



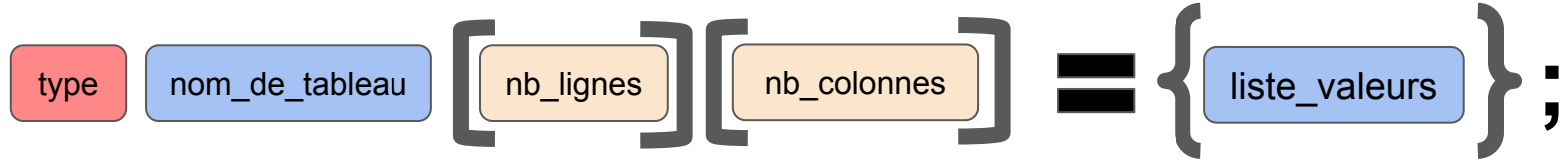
Exemple:

```
char mon_tableau[2];  
short tab[3] = {12, 23, 1};  
short liste[] = {24, 60}
```



Taille du tableau fixé à la déclaration et ne peut pas varier

Déclaration Tableaux 2D

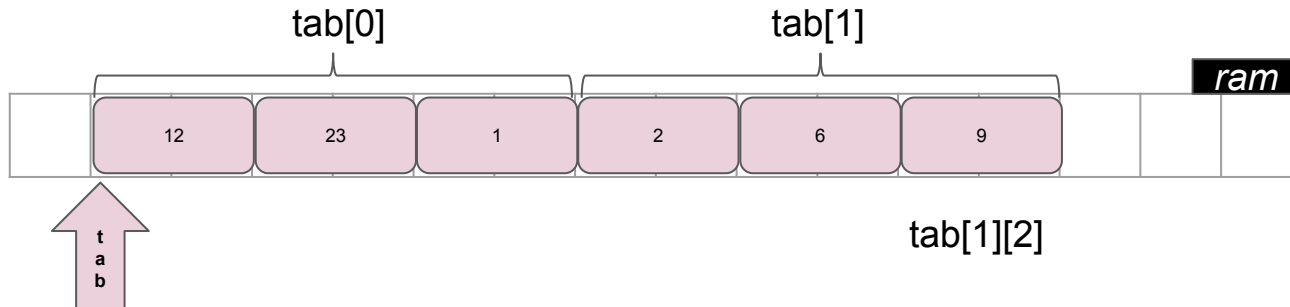


Exemple:

```
short tab[2][3] = { {12, 23, 1}, {2, 6, 9} };
```

```
tab[1][2] //retourne la valeur 9
```

		Colonnes		
		0	1	2
Lignes	0	12	23	1
	1	2	6	9

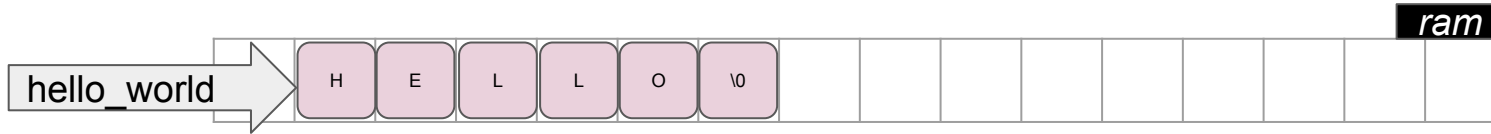


Les strings

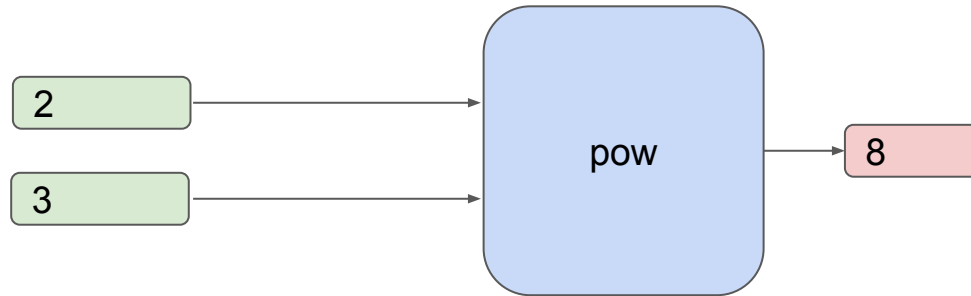
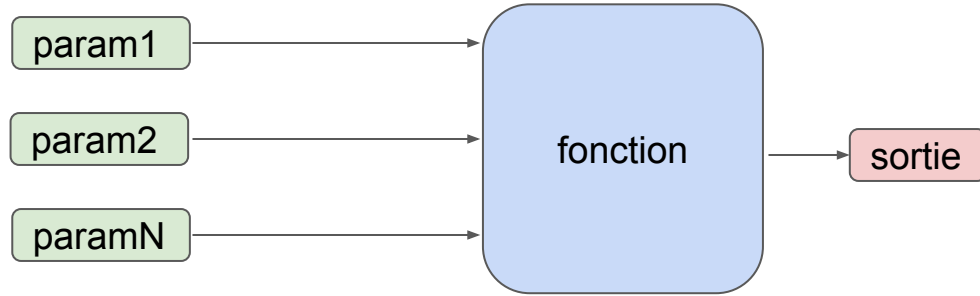
Example:

```
char hello_world[6] = { 'H', 'E', 'L', 'L', 'O', '\0' };  
printf("%s", hello_world) -> HELLO
```

H	E	L	L	O	\0
---	---	---	---	---	----



les fonctions



$$2^3 = 2 * 2 * 2 = 8$$

Les fonctions

Signature

```
type_retour nomFonction ( liste_params )  
{  
    instructions  
    return valeur;  
}
```

Structure

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

Exemple 1

```
void maFonction()  
{  
    ...  
}
```

Exemple 2

```
void afficherAge(int age)  
{  
    printf("tu a %d ans\n", age);  
    return;  
}
```

Exemple 3

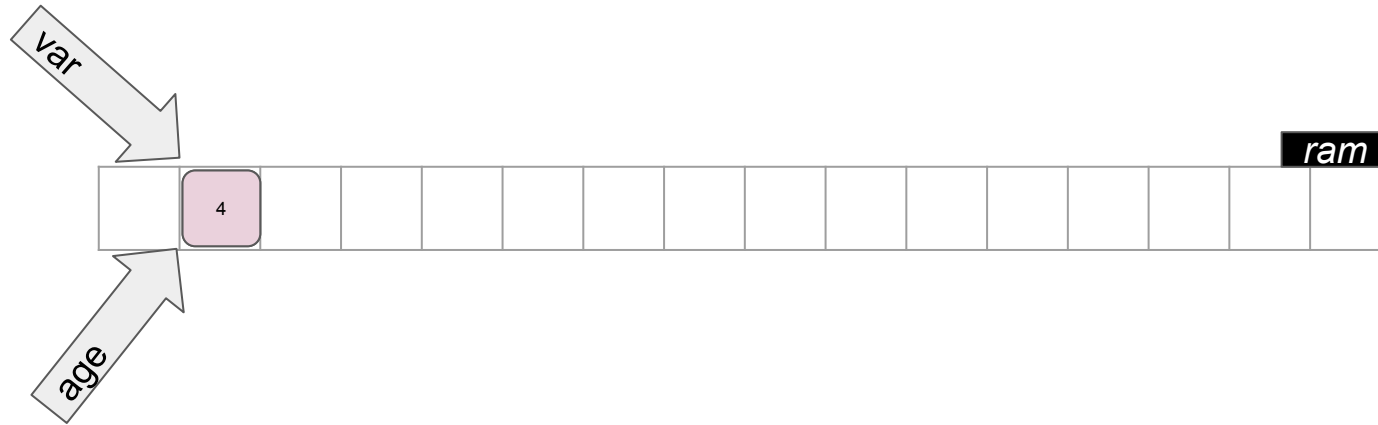
```
int somme(int var1, int var2)  
{  
    int total= var2 + var1;  
    return total;  
}
```

Exemple 4

Passage de paramètres par pointeur

```
int main()  
{  
    int age= 3;  
    anniversaire(&age);  
}
```

```
void anniversaire(int* var)  
{  
    *var++;  
    printf("%d", *var);  
}
```



Les defines

#define

NOM_DU_DEFINE

valeur

```
#define TAILLE_NOM 100
#define NOM_DEFAULT "Jon snow"
#define INIT_PV 5
#define NB_MAX_PV (INIT_PV *2)

int main()
{
    int pv = NB_MAX_PV;
    char nom[TAILLE_NOM] = NOM_DEFAULT;

    ....
    if(pv < NB_MAX_PV)
    ....
```

préprocesseur

```
int main()
{
    int pv = (5*2);
    char nom[100] = "Jon snow";

    ....
    if(pv < 10)
    ....
```

Nommage: tout en majuscule et mots séparés par des _

Les conditions de préprocesseur

#ifdef

NOM_DU_DEFINE

code_source

#else

code_source

#endif

#ifndef

NOM_DU_DEFINE

code_source

#else

code_source

#endif

```
#ifdef AFFICHER_HELLO
    printf("Hello\n");
#else
    pv++;
#endif
```

préprocesseur

AFFICHER_HELLO est défini

```
printf("Hello\n");
```

```
pv++
```

AFFICHER_HELLO n'est pas défini

Les macros

#define

NOM_MACRO

(params)

(expression)

```
#define DOUBLER(x) ( (x) * 2 )  
.....  
int main()  
{  
    int bonus = 20;  
    bonus = DOUBLER(bonus);  
}
```

préprocesseur

```
int main()  
{  
    int bonus = 20;  
    bonus = ((bonus) * 2);  
}
```

Les fichiers .h et .c

fichiers headers (.h)

- Prototypes des fonctions
- Enums

Publique

fichiers source (.c)

- Déclaration des fonctions
- Enums
- Variables globales

Privé

math.h

```
#define M_PI 3.1416....  
  
int pow(int,int);
```

math.c

```
int pow(int base, int exposant)  
{  
    ...  
}
```

Exemple