

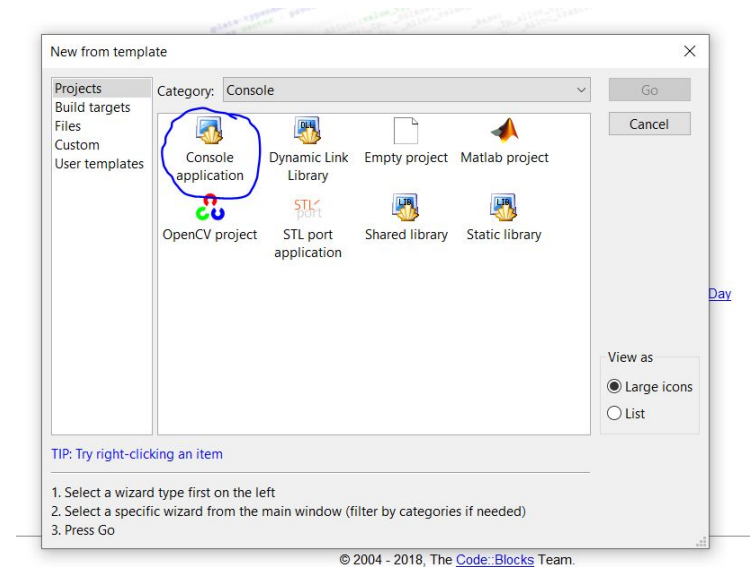
Apprendre à coder en langage C

Le langage C est un programme avec console.

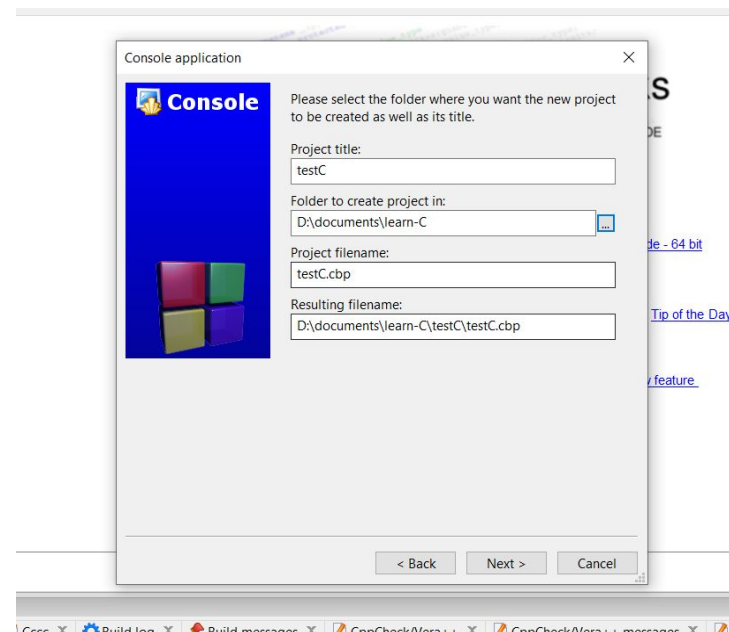
Le C est un langage *compilé* (par opposition aux langages interprétés). Cela signifie qu'un programme C est décrit par un fichier texte, appelé *fichier source*. Ce fichier n'étant évidemment pas exécutable par le microprocesseur, il faut le traduire en langage machine. Cette opération est effectuée par un programme appelé *compilateur*.

Après avoir téléchargé: Code Block: <http://www.codeblocks.org/downloads/26#windows>

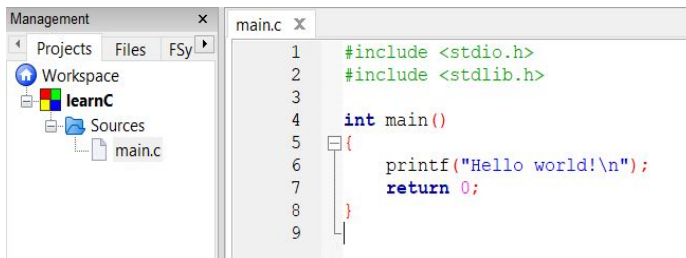
Dans l'interface crée un new projet choisir projet->console->console application



Créer un dossier puis donner un nom à votre projet



Le dossier est généré avec un fichier main.c il s'agit du code source



inclure les librairie->**directives de préprocesseur**

init main() equivalent à int main(int argc, char *argv[])

-> le programme commence par main

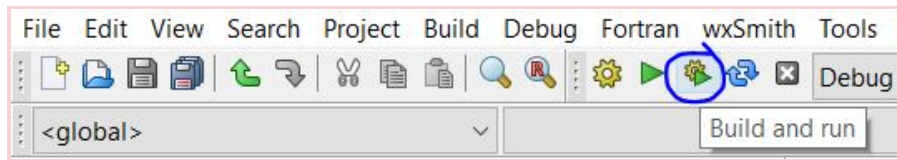
-> printf affiche le message

-> return 0 = tout s'est bien passé

Très important : toute instruction se termine **obligatoirement** " ; "

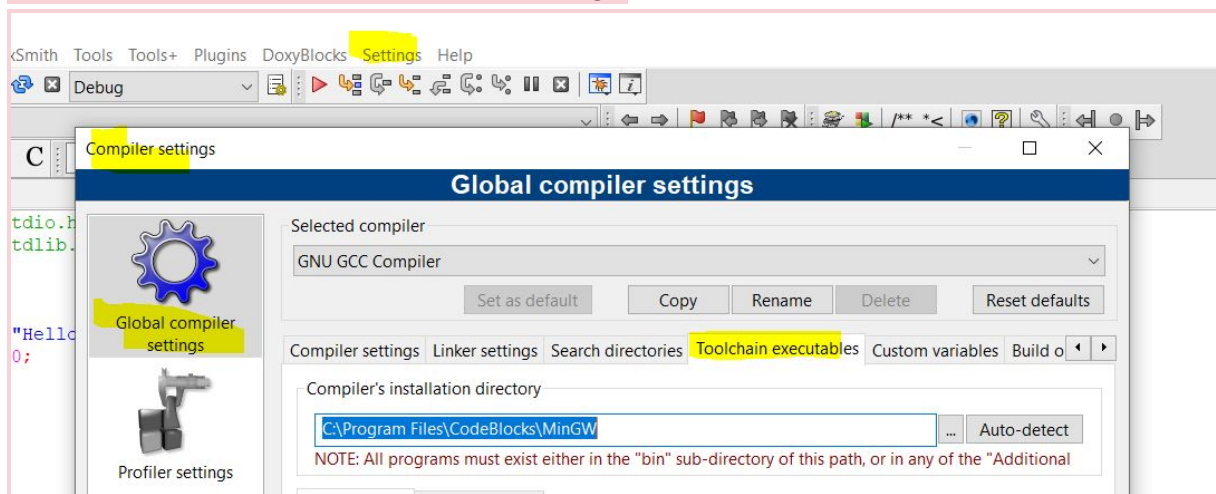


POUR EXÉCUTER LE CODE POUR LE COMPILER cliqué sur



Si il y a une erreur vérifier si la ressource mingw est installer

Vérifier le chemin d'accès vers le dossier Mingw



SI IL Y A PAS ERREUR LE PROMPT EST OUVERT AVEC EXÉCUTION DES INSTRUCTION

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     printf("Hello world!\n");
7     return 0;
8 }
9
10

```

Process returned 0 (0x0) execution time : 0.010 s
Press any key to continue.

- \n: retour à la ligne (= « Entrée ») ;
- \t: tabulation.

printf("Bonjour\n");

```

C:\Users\Mateo\Projets\test\bin\Release\test.exe
Bonjour
Process returned 0 (0x0) execution time : 0.096 s
Press any key to continue.

```

Nom du type	Minimum	Maximum
signed char	-127	127
int	-32 767	32 767
long	-2 147 483 647	2 147 483 647
float	-1 x1037	1 x1037
double	-1 x1037	1 x1037

CRÉER DES VARIABLES:

pour un nombre **entier**, on utilisera le plus souvent int;

pour un nombre **flottant**, on utilisera généralement double.

Créer la variable -> int mavariable;

Affecter une valeur à la variable -> mavariable=5

Crée + affecter-> int mavariable=5

Il n'y a pas de valeur par défaut cette valeur peut être n'importe quoi

```
const int NOMBRE_DE_VIES_INITIALES = 5;
```

Bonne pratique les constantes sont en MAJUSCULE

- AFFICHER LE CONTENU D'UNE VARIABLE

Format	Type attendu
"%d"	int
"%ld"	long
"%f"	float
"%f"	double

Afficher la variable appeler son format si c init %d puis savoir qu'elle variable affichier via , nom variable

```
int cascovid = 200;
```

```
printf("Il y a %d cas covid en 2 jours \n",  
cascovid);
```

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

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

```
int main(int argc, char *argv[])  
{  
    int cascovid = 200;  
  
    printf("Il y a %d cas covid en 2 jours \n", cascovid);  
  
    return 0;  
}
```

Avec 1 + plusieurs variable

```
printf("Vous avez %d vies et vous etes au niveau n° %d", nombreDeVies,  
niveau);
```

- RECUP + SAISIR

Pour demander à l'utilisateur d'entrer quelque chose dans la console, on va utiliser une autre fonction toute prête : **scanf**.

```
int main(int argc, char *argv[])  
{  
    int age = 0; // On initialise la variable à 0  
  
    printf("Quel age avez-vous ? ");
```

```
scanf("%d", &age); // On demande d'entrer l'âge avec scanf
printf("Ah ! Vous avez donc %d ans !\n\n", age);
```

```
return 0;
}
```

● FONCTION MATH

Nom fonctions	Utilisations	Exemples
fabs	Cette fonction retourne la valeur absolue d'un nombre, c'est-à-dire $ x $	double absolu = 0, nombre = -27; absolu = fabs(nombre); // absolu vaudra 27
ceil	On arrondit en fait toujours au nombre entier supérieur	double dessus = 0, nombre = 52.71; dessus = ceil(nombre); // dessus vaudra 53
pow	calculer la puissance d'un nombre	Par exemple, « 2 puissance 3 » (que l'on écrit habituellement 2^3 sur un ordinateur), c'est le calcul $2 * 2 * 2$, ce qui fait 8 double resultat = 0, nombre = 2; resultat = pow(nombre, 3); // resultat vaudra $2^3 = 8$
sqrt	racine carrée	double resultat = 0, nombre = 100; resultat = sqrt(nombre); // resultat vaudra 10
rand()	génère un nombre au hasard	srand(time(NULL)); nombreMystere = (rand() % (MAX - MIN + 1)) + MIN;

- L'**incrément** est l'opération qui consiste à ajouter 1 à une variable. On écrit variable++.
- La **décrément** est l'opération inverse : on retire 1 à une variable. On écrit donc variable--.
- Pour augmenter le nombre d'opérations connues par votre ordinateur, il faut charger la **bibliothèque mathématique** (c'est-à-dire `#include <math.h>`).
- Cette bibliothèque contient des **fonctions mathématiques plus avancées**, telles que la puissance, la racine carrée, l'arrondi, l'exponentielle, le logarithme, etc.

```

int main(int argc, char *argv[])
{
    int nbr1 = 0;
    int nbr2=0;
    int resultat=0;

    printf("Entre votre 1er chiffre");

    scanf("%d", &nbr1);

    printf("Entre votre 2er chiffre");

    scanf("%d", &nbr2);

    resultat= nbr1/ nbr2;

    printf("Total  %d + %d =%d",nbr1,nbr2,resultat);

    return 0;
}

```

- ATTENTION LES CARACTÈRE SPÉCIAUX SONT PAS PRISE EN COMPTE

Ajouter ceci dans le code

```

#include <windows.h>
#include <stdio.h>
#include <locale.h>

int main()
{
    setlocale (LC_ALL,"");

```

LES BOUCLES

Boucles : While

C'est aussi simple que cela.whilesignifie « Tant que ». On dit donc à l'ordinateur « Tant que la condition est vraie, répète les instructions entre accolades ».

Ex:

```
int compteur = 0;
while (compteur < 10)
{
    printf("La variable compteur vaut %d\n", compteur);
    compteur++;
}
```

Boucle: FOR

```
int compteur = 0;

while (compteur < 10)
{
    printf("Salut les Zeros !\n");
    compteur++;
}
```

- **CRÉER UNE FONCTION**

Pour déclarer une fonction on utilise la fonction **triple**. Reçoit un nombre entier de type **int** et qu'elle renvoie un nombre entier aussi de type **int**.

```
int triple(int nombre)
{
    int resultat = 0;
    resultat = 3 * nombre; // On multiplie le nombre fourni par 3
    return resultat;      // On retourne la variable resultat qui vaut le triple de nombre
}
```