

Quand on écrit un programme informatique, il faut être très rigoureux car un simple caractère mal placé peut perturber l'ordinateur du robot. Si le format de votre code n'est pas bon, il ne va pas aller plus loin et va simplement vous indiquer que vous avez commis une erreur.

Quand ça arrive, il faut d'abord **tenter de comprendre le message d'erreur** et ensuite **relire attentivement la zone de code concernée**, en particulier pour s'assurer qu'il ne manque pas de symbole tel qu'une parenthèse ou un guillemet.

Voici quelques exemples de messages d'erreur.

Attention aux points-virgules !

Si on oublie le point-virgule, cela produit une erreur :

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Bonjour" << endl
}
```

↳ *line 7:1: error: expected ';' before '}' token*

Si on traduit ce texte, il nous dit justement qu'on a oublié le « ; ».

Attention aux guillemets !

Si on ne place pas correctement les guillemets, cela produit une des erreurs suivantes :

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Bonjour << endl;
}
```

↳ *line 6:12: warning: missing terminating " character*
line 6:4: error: missing terminating " character
In function 'int main()':
line 7:1: error: expected primary-expression before '}' token

```
#include <iostream>
using namespace std;

int main()
{
    cout << Bonjour tout le monde << endl;
}
```

↳ *In function 'int main()':*
line 6:12: error: 'Bonjour' was not declared in this scope

Conclusion

Il faut être précis et ne rien oublier. Si l'on rencontre une erreur, on pensera à bien tout vérifier. Les messages décrivent très souvent la source exacte du problème, en anglais : il faut les lire et essayer de comprendre leur signification.

Un message d'erreur vous donnera aussi toujours le numéro de la ligne qui a causé le souci. Ce numéro n'est pas toujours exact : l'ordinateur peut n'être dérangé qu'après votre erreur. Il vous faudra donc souvent regarder avant l'emplacement indiqué.