

*Nous aurons besoin de*



*L'Arduino Uno*



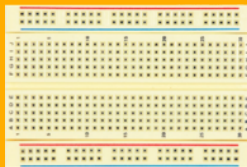
*1 LED*



*Le cable USB*



*L'ordinateur*



*La planche  
d'essai*



*1 résistance  
220  $\Omega$*



*Le programme  
FeuClignotant*



*L'éditeur de  
programme*

Le programme comporte 2 blocs principaux :

- setup qui dit ce qu'il faut faire 1 seule fois au démarrage
- loop qui dit ce qu'il faut faire de manière répétée à chaque cycle

Le programme commence par plusieurs ligne entre /\* et \*/.  
Ce sont des **commentaires**, c'est à dire des informations pour le programmeur.

La ligne qui commence par // est aussi un commentaire.  
Celui ci est sur une seule ligne.

La première ligne utilisable par l'Arduino est

```
void setup() { ... }
```

C'est une **fonction**, on y reviendra plus en détail plus tard.

Ce que fait la fonction est décrit entre les signes { }. Ces signes s'appellent des **accolades**.

`void loop() { ... }` est aussi une fonction.

Ces fonctions contiennent les instructions que l'on donne à l'Arduino. Ce sont aussi des fonctions mais celles-ci sont décrites dans du code qui sera envoyé avec notre programme. On n'a pas à les écrire. On appelle ce code fourni avec l'Arduino une **bibliothèque de fonctions**.

La LED s'éclaire quand il y a du courant et s'éteint quand il n'y en a plus. Pour la faire clignoter on va envoyer un courant, attendre un peu, puis ne plus envoyer le courant, attendre un peu. Et ensuite on recommence, c'est `loop` qui gèrera ça.

- `pinMode(13, OUTPUT)` indique que l'alimentation est faite par le pin13
- `digitalWrite(13, HIGH)` envoie la quantité maximale de courant sur le pin 13. La LED s'éclaire.
- `delay(1000)` attend 1s (1000 ms)
- `digitalWrite(13, LOW)` envoie la quantité minimale de courant sur le pin 13. La lampe s'éteint.

Les valeurs qui sont entre parenthèses après les fonctions s'appellent des **paramètres**. Par exemple, un paramètre permet d'utiliser `delay` pour attendre plus ou moins longtemps selon la valeur.

- `delay(1000)` attend 1s (1 000 ms)
- `delay(10000)` attend 10s (10 000 ms)