INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

Comment programmer ?

Programme 1 :

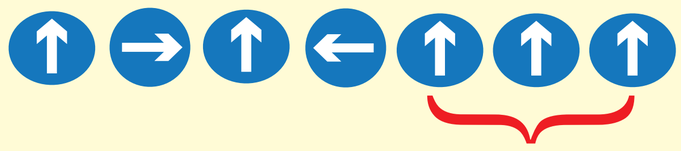
```
avance ( ) ;
tourne (90) ;
avance ( ) ;
tourne (-90) ;
avance ( ) ;
avance ( ) ;
avance ( )
```

Ce programme fait ceci :

Programme 2 :

```
avance ( ) ;
tourne (90) ;
avance ( ) ;
tourne (-90) ;
répète (3, avance)
```

Celui-là fait la même chose
mais il est plus court à écrire.



Comme le français, un langage de programmation a une grammaire.

Les règles de la grammaire doivent être suivies à la lettre car seuls les programmes bien écrits peuvent être traduits en langage machine par le compilateur.

Dans notre exemple, j'écris «avance (); tourne (90)» en suivant la règle de grammaire suivante:

Règle 1 : «Un programme est bien écrit si il commence par un programme bien écrit P suivi d'un point-virgule puis d'un programme bien écrit Q.»

Ainsi, le code source «avance () et puis tourne (90)» n'est pas un programme bien écrit.

Un langage de programmation propose des primitives.

Les programmes s'appuient sur des mécanismes primitifs pour construire des mécanismes plus complexes, sur lesquels reposent la construction de mécanismes encore plus complexes ... et ainsi de suite.

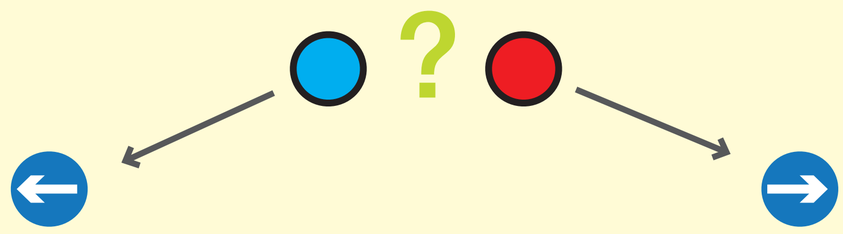
Dans notre exemple, quand j'écris «tourne (90)», je fais appel à une action primitive de mon langage. Il y a aussi des règles à suivre pour les utiliser :

Règle 2 : «Pour faire appel à une action primitive, j'écris le nom de cette action puis une parenthèse, puis une liste d'arguments séparés par des virgules et enfin, une parenthèse fermante.»

Programme 3 :

```
si
couleur_est (rouge)
alors
tourne (90)
sinon
tourne (-90)
```

Ce programme agit différemment
en fonction des cas :



Programme 4 :

```
tant que
non
couleur_est (blanc)
faire
 tourne (1)
```

Ce programme s'adapte à
de nombreuses situations !