

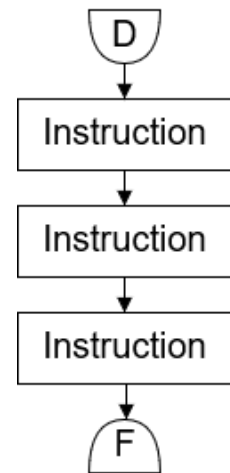
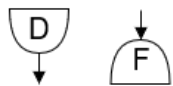
Logigrammes

Les logigrammes sont très souvent utilisés en informatique pour schématiser un algorithme avant de le coder. Les logigrammes portent parfois d'autres noms (ordinogramme par exemple) et peuvent être dessinés de nombreuses manières différentes. Il existe certaines normes plus ou moins reconnues et complexes...

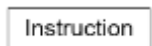
Nous allons présenter ici l'écriture que nous utiliserons dans nos fiches. Elle est simple mais suffisante à ce stade. Chaque type d'instruction a une forme différente et l'ordre dans lequel elles sont exécutées est représenté par des flèches.

Début et fin de programme

Le début et la fin d'un programme sont représentés par des demi-cercles. En programmation *séquentielle*, le programme commencera toujours par un « début » et finira toujours par une « fin ». Entre les deux, il y aura un nombre indéfini d'instructions.

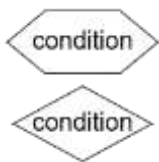


Instruction simple

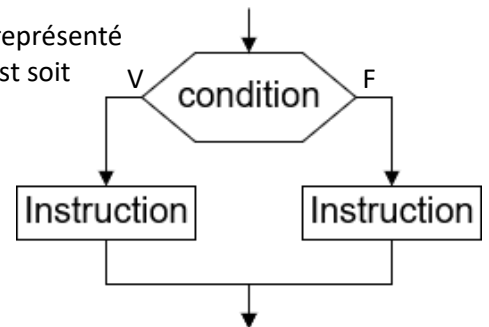


Une instruction simple est représentée grâce à un rectangle

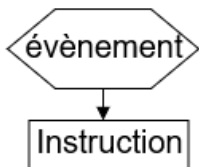
Condition



Les conditions se trouvent dans un hexagone (parfois représenté par un losange, plus facile à dessiner). Une condition est soit vraie (« alors... ») soit fausse (« sinon... »), il y a donc 2 chemins possibles depuis un hexagone/losange qui se rejoignent ensuite pour continuer le programme. Il se peut qu'un des deux chemins ne contienne aucune instruction (il n'y a pas toujours de « sinon... »), dans ce cas, le chemin doit être tracé mais ne contiendra aucune forme.



Évènement



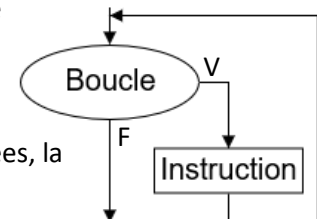
Les évènements n'existent pas en programmation *séquentielle* mais ils existent en programmation *évènementielle*. Il se peut en effet que quelque chose déclenche un évènement particulier.

Un évènement sera représenté par la même forme qu'une condition¹ mais sans flèche d'arrivée et sans sinon.

Boucle

Les boucles sont représentées par un cercle (ovale) dans lequel on inscrit une condition². Si la condition est vraie, on rentre dans la boucle, si elle est fausse, on sort de la boucle (ça pourrait être l'inverse, en fonction d'où se trouvent V et F).

Une fois que toutes les instructions contenues dans la boucle ont été exécutées, la flèche ramène juste avant le cercle, c'est-à-dire à la condition.



¹ Car c'est un cas particulier de condition sans sinon : par exemple « lorsqu'il fait clair, je me réveille » est pareil que « s'il fait clair, je me réveille » mais c'est un « si » qui n'apparaîtra pas à un moment précis de la suite d'instruction du programme mais sera vérifié continuellement.

² Il se peut qu'il n'y ait aucune condition (dans le cas d'une boucle infinie), dans ce cas, il n'y a pas d'obligation de mettre le cercle.