

Dans cette première leçon de l'Unité 3 vous allez étudier les conditions et une introduction à l'instruction **If** disponible en TI-Basic.

**Objectifs :**

- Étudier les conditions.
- Utiliser l'instruction 'simple' **If...** pour passer sous certaines conditions à une autre instruction.

Les structures conditionnelles **If...Then** sont utilisées pour passer à un bloc d'instructions seulement quand une *condition* est vraie ou fausse. Avant de voir l'ensemble des structures conditionnelles **If...Then**, ayons une idée de ce qu'est au juste une condition.

**Indication :** Les opérateurs de comparaison et les opérateurs logiques ont aussi leur ordre d'opérations et s'intègre ainsi dans l'ordre arithmétique des opérations.

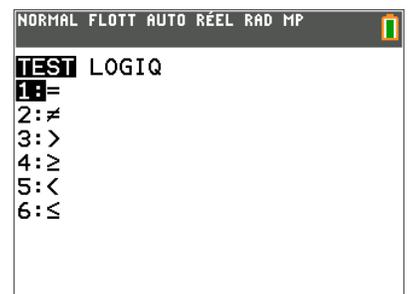
Voici la liste complète de l'ordre en TI-Basic :

Niveau de priorité Fonctions

1	Fonctions qui précèdent leur argument, comme $\sqrt{\quad}$ ou $\sin(\quad)$ , excepté la négation
2	Fonctions qui suivent leur argument, comme $^2$ ou $!$
3	$^$ et $\times\sqrt{\quad}$
3.5	Négation
4	$nPr$ et $nCr$
5	Multiplication, division, et multiplication implicite
6	Addition et soustraction
7	Les opérateurs de comparaison $=, \neq, <, >, \leq, \geq$
8	L'opérateur logique <b>and</b>
9	Les opérateurs logiques <b>or</b> and <b>xor</b>
10	Les conversions comme $\blacktriangleright$ <b>Frac</b>

**Conditions et le menu [tests]**

Les conditions sont des expressions qui peuvent être évaluées à 'vrai' ou 'faux'. Ces expressions sont soit vraies soit fausses ; elles ne peuvent pas être les deux à la fois ou ni l'un ni l'autre. Les opérateurs de comparaison et les opérateurs logiques se trouvent tous dans les menus de [tests] (**2nde** **math**). Le menu TEST contient les **opérateurs de comparaison**. Le menu LOGIQ contient les **opérateurs logiques**. Le signe égal (=) est utilisé pour former une condition, pas une affectation.



Ce document est mis à disposition sous licence Creative Commons



<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>

Exemples de quelques conditions :

$3 > 5$	$X * Y > 0$	$X = 5$ ou $Y = 5$
$X + 4 > X$	$B^2 - 4AC = 0$	$X/2 = \text{partEnt}(X/2)$
$X \neq Y$	$X > 0$ et $Y > 0$	$\text{non}(X > 0)$

```
NORMAL FLOTT AUTO RÉEL RAD MP
TEST LOGIC
1: et
2: ou
3: xor
4: non(
```

Conditions sur l'écran de calcul

Vous pouvez entrer des conditions directement dans l'écran de calcul pour voir comme elles sont traitées.

Observez que 1 représente *vrai (true)* et 0 représente faux (*false*).

*Note : lorsque vous utilisez une variable dans une condition, la calculatrice l'évalue en utilisant la valeur courante stockée dans la variable.*

```
NORMAL FLOTT AUTO RÉEL RAD MP
3>5
.....0
3≠5
.....1
X+1=X
.....0
X+1>X
.....1
```

**Indication :** Nous introduisons l'instruction **If** 'de base' ou 'simple' ci-dessous car c'est une façon pratique d'exécuter sous condition une seule instruction (si cela suffit). Mais l'instruction plus polyvalente **If...Then** examinée après est préférée, car il est plus clair à un lecteur du programme de voir ce que ce dernier tente de faire.  
La structure **:If** <condition> (sans **Then**) traite seulement l'instruction suivante quand la condition est vraie, sinon celle-ci est omise.

Programmer avec l'instruction **If... 'Simple'**

Essayez ce programme :

```
:Prompt A
:If A>0 (If est dans le menu CTL de [prgm]. '>' est dans le menu TEST de [tests])
:Disp "A POSITIF"
:Disp "A NEGATIF"
```

```
NORMAL FLOTT AUTO RÉEL RAD MP
prgmIFSTMT
A=?5
A POSITIF
A NEGATIF
.....Fait.
```

Lancez plusieurs fois le programme en entrant des nombres aussi bien positifs que négatifs et observez le résultat. Que peut-on en conclure ?

Quand la condition  $A > 0$  est vraie, l'instruction qui suit le **If** est exécutée, sinon elle est simplement omise. Par contre la seconde instruction qui affiche "A NEGATIF" est toujours exécutée, ce qui n'est pas correct ! Voir l'écran à droite. Nous allons réparer cela rapidement.

Le 'simple' **If...** est un moyen concis pour sauter **une** instruction basée sur une condition (quand elle est FAUSSE).

**Modifions le programme**

Corrigeons le programme ci-dessus en ajoutant un autre **If...**

1. Placez le curseur sur le second **Disp**.
2. Appuyez sur [insérer] puis sur  pour insérer une ligne vide.
3. Sur cette ligne vide ajoutez **If A<0**.

```
NORMAL FLOTT AUTO RÉEL RAD MP
PROGRAM: IFSTMT
:Prompt A
:If A>0
:Disp "A POSITIF"
:If A<0
:Disp "A NEGATIF"
```

Quittez et lancez le programme plusieurs fois en utilisant des nombres positifs, négatifs et également 0 !

Votre programme fonctionne-t-il correctement dans tous les cas ? Si ce n'est pas le cas, essayez de résoudre le problème.