

# L'instruction conditionnelle if

## Plan du T.P.

<b>1</b>	<b>Un logiciel de caisse automatique au cinéma</b>	<b>21</b>
<b>2</b>	<b>Les conditions</b>	<b>22</b>
2.1	Conditions	22
2.2	Opérateurs de comparaison	22
<b>3</b>	<b>L'instruction conditionnelle if en Python</b>	<b>23</b>
3.1	Manipuler	23
3.2	Verbaliser	23
3.3	Abstraire	23
3.4	Instruction alternative	24
<b>4</b>	<b>Cahier des charges</b>	<b>24</b>
<b>5</b>	<b>Programmation modulaire</b>	<b>25</b>
<b>6</b>	<b>Algorithmique et Informatique</b>	<b>27</b>

[Retour au sommaire](#)

## Quelques grandes figures de l'Histoire de l'Informatique



Cliquer une image au choix

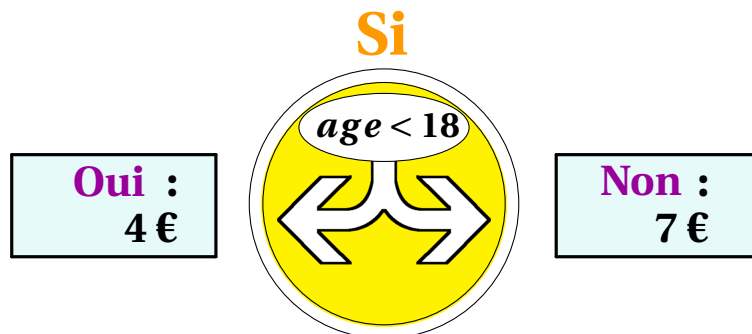
[Ada Lovelace](#) - [Alan Turing](#) - [John von Neumann](#)  
[Richard Stallman](#) - [Tim Berners-Lee](#) - [Linus Torvalds](#)

# 1 Un logiciel de caisse automatique au cinéma

## Exercice 1 :

Nous programmerons ici une version très simplifiée d'un tel logiciel. Il devra pouvoir :

- recevoir en entrée l'âge du client,
- choisir de renvoyer en retour le juste tarif à appliquer :



a. Compléter l'algorithme du logiciel en langage naturel :

<ul style="list-style-type: none"><li>• Si age &lt; 18 : .....</li><li>• Sinon : .....</li></ul>
--

b. De quel type d'objet informatique aurions-nous besoin pour stocker la valeur à renvoyer à l'utilisateur?

.....

c. Compléter à nouveau l'algorithme en utilisant une variable prix :

<ul style="list-style-type: none"><li>• Si age &lt; 18 : .....</li><li>• Sinon : .....</li></ul>
--

En Python, nous traduirons Si, par `if`. L'instruction `if` permet de sélectionner un traitement à appliquer, selon qu'une condition est vraie ou fausse. On parle ainsi d'instruction conditionnelle.

e. Quelle est la condition à tester dans l'algorithme ci-dessus?

.....

f. Quels sont les deux traitements possibles?

.....

.....

## 2 Les conditions

### 2.1 Conditions

#### Exercice 2 : les conditions

Pour chacune des instructions ou séquences d'instructions du tableau suivant, prévoir le résultat qu'affichera la console, puis vérifier en saisissant ces instructions dans la console.

<pre>&gt;&gt;&gt; 3 + 1 == 4</pre>	<pre>&gt;&gt;&gt; 3 + 1 == 5</pre>	<pre>&gt;&gt;&gt; 3 + 1 = 4</pre>
<pre>&gt;&gt;&gt; 5 &lt;= 3</pre>	<pre>&gt;&gt;&gt; 3.1 &lt; 17.3</pre>	<pre>&gt;&gt;&gt; -8 &gt; -7</pre>
<pre>&gt;&gt;&gt; "bonjour" == "bonjour"</pre>	<pre>&gt;&gt;&gt; "bonjour" == "Bonjour"</pre>	<pre>&gt;&gt;&gt; "bonjour" != "Bonjour"</pre>
<pre>&gt;&gt;&gt; x = 5 &gt;&gt;&gt; y = 3 &gt;&gt;&gt; x == y</pre>	<pre>&gt;&gt;&gt; x = 3 &gt;&gt;&gt; x = x + 1 &gt;&gt;&gt; x = 8 * x &gt;&gt;&gt; x == 36</pre>	<pre>&gt;&gt;&gt; x = 5 &gt;&gt;&gt; y = 3 &gt;&gt;&gt; a = x**2 - y**2 &gt;&gt;&gt; b = (x + y)(x - y) &gt;&gt;&gt; a == b</pre>

#### Définition :

- Une **condition** est un énoncé qui peut être **Vrai** ou **Faux**.
- Cela permet au programme, selon qu'une condition est vraie ou fausse, d'effectuer une action ou une autre. On parle d'**exécution conditionnelle**.

### 2.2 Opérateurs de comparaison

La condition testée peut contenir les opérateurs de comparaison suivants :

- **x == y** : teste si x est égal à y (→ en Python, "le conditionnel se conjugue en double égal"),
- **x != y** : teste si x est différent de y,
- **x > y** : teste si x est supérieur strictement à y,
- **x < y** : teste si x est inférieur strictement à y,
- **x >= y** : teste si x est supérieur ou égal à y,
- **x <= y** : teste si x est inférieur ou égal à y.

#### Attention :

- Il ne faut pas confondre l'**opérateur d'affectation** =, utilisé pour affecter une valeur à une variable (x=2), et l'**opérateur de comparaison** ==, utilisé pour tester l'égalité de deux valeurs ou deux variables (x == y).
- Un **procédé mnémotechnique** <sup>(1)</sup> : en Python, le conditionnel se conjugue en double égal.

#### Exercice 3 :

Nous allons avoir besoin de pouvoir tester si une condition, telle que : "age ≤ 18", est vraie ou fausse (**True** ou **False**, en Python). Prévoir le résultat des instructions suivantes et vérifier dans la console :

```
1 >>> age = 15
2 >>> age <= 18
3 .....
4 >>> age_papa = 45
5 >>> age_papa <= 18
6 .....
```

## 3 L'instruction conditionnelle if en Python

### 3.1 Manipuler

#### Exercice 4 :

Python exécute normalement les instructions de la première à la dernière ligne, sauf lorsqu'il rencontre une instruction conditionnelle `if`.

Cette instruction va servir d'aiguillage dans le programme, et lui permettre de suivre des chemins différents selon les circonstances : plus ou moins de 18 ans, par exemple.

a. Que renvoie la séquence d'instructions suivante ? Vérifier dans la console.

```
1 >>> age = 16
2 >>> if age < 18 :
3 ...     print("Vous êtes mineur")
```

b. Que renvoie la séquence d'instructions suivante ? Vérifier dans la console.

```
1 >>> age = 25
2 >>> if age < 18 :
3 ...     print("Vous êtes mineur")
```

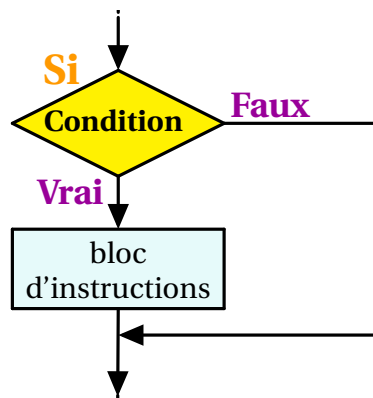
### 3.2 Verbaliser

Dans l'exercice ci-dessus question a., la condition `age < 18` est vraie, donc la ligne 3 a été lue et exécutée. Dans la question b., la condition est fausse, donc la ligne 3 n'a pas été lue et n'a pas été exécutée.

Le bloc d'instruction(s) à exécuter lorsque la condition est vraie doit être **indenté** (décalé de 4 espaces). La fin de l'indentation signale la fin de ce bloc d'instruction.

### 3.3 Abstraire

```
1 if condition :
2     instructions
```



(1). <https://fr.wikipedia.org/wiki/Mnémotechnique>

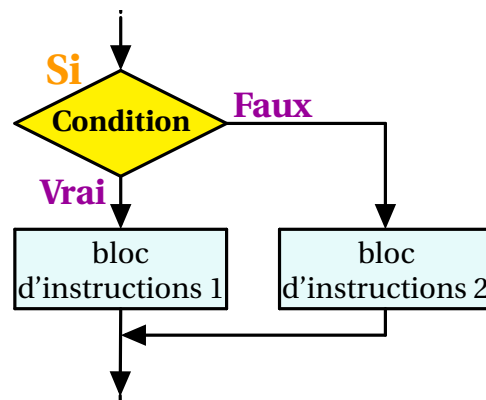
### 3.4 Instruction alternative

Dans le cas où la condition n'est pas satisfaite, nous pouvons programmer une **exécution alternative** à l'aide de l'instruction `else`.

#### Exercice 5 :

Prévoir l'effet des instructions suivantes. Vérifier dans la console.

```
1 >>> age = 25
2 >>> if age < 18 :
3 ...     print ("Vous êtes mineur")
4 ... else :
5 ...     print ("Vous êtes majeur")
```



## 4 Cahier des charges

Nous allons maintenant écrire des instructions conditionnelles à l'intérieur de nos fonctions. La fonction `print` que nous venons d'utiliser afin de découvrir et manipuler l'instruction conditionnelle `if` offre juste un affichage mais ne permet pas de renvoyer un résultat que nous pourrions ré-exploiter.

Nous utiliserons pour cela des **fonctions**, qui renvoient un résultat au moyen de l'**instruction** `return`.

#### Syntaxe d'une fonction :

```
1 def nom_de_la_fonction(argument1, argument2, ...) :
2     instructions
3     return resultat
```

#### Exercice 6 : Logiciel de caisse automatique

Nous sommes maintenant prêts à programmer notre logiciel.

Écrire une fonction `tarif(age)` :

- prenant en argument l'âge du client,
- et renvoyant, à l'aide de l'instruction `return`, le tarif appliqué..

## 5 Programmation modulaire

### Exercice 7 : abonnement

La compagnie de bus YESGO propose deux types de tarifs à ses usagers :

- tarif 1 : 1,30 € par trajet,
- tarif 2 : un abonnement mensuel de 12 €, puis 0,5 € par trajet.

Nous allons ici programmer plusieurs fonctions, puis les faire interagir entre elles pour choisir la formule la plus adaptée, selon le contexte.

### Manipuler

- Écrire une fonction `tarif_1(x)` :
  - prenant en argument le nombre  $x$  de trajets,
  - et renvoyant le prix à payer par la première formule.
- Écrire une fonction `tarif_2(x)` :
  - prenant en argument le nombre  $x$  de trajets,
  - et renvoyant le prix à payer par la seconde formule.
- Déterminer le rôle, et compléter la fonction suivante :

```
Fonction meilleur_tarif(x) :  
    Si tarif_1(x) < tarif_2(x) :  
        meilleur = 1  
    Sinon :  
        .....  
    Renvoyer .....
```

Rôle de cette fonction : .....

- Écrire en Python une fonction `meilleur_tarif(x)`.
- Quelle formule choisir si l'on doit réaliser 5 trajets ? 20 trajets ?

### Verbaliser

- Rendons-nous sur le site **Python Tutor** : <http://pythontutor.com/visualize.html#mode=edit>.
- Copier-coller l'ensemble de votre programme (les 3 fonctions), dans la fenêtre de code proposée.
- Ajouter en fin de programme un appel de la fonction `meilleur_tarif` : `meilleur_tarif(5)`
- Cliquer sur le bouton "Live Programming Mode", puis sur le bouton "« First".
- Vous allez maintenant pouvoir visualiser l'ordre dans lequel Python lit les lignes du programme, ainsi que les objets informatiques qu'il crée.

Pour cela, cliquer autant de fois que nécessaire sur le bouton "Forward >", pour **exécuter le programme pas à pas**.

- Ordre de lecture des lignes lors des appels :

```
meilleur_tarif(5) : .....  
meilleur_tarif(20) : .....
```

## Abstraire

Avec la programmation modulaire, nous découpons une tâche complexe en tâches plus simples. Chacune des **tâches simples** est ici réalisée par une **fonction spécifique**. L'interaction de ces différentes fonctions permet de résoudre un problème plus complexe.

En somme, en Informatique, l'expression **diviser pour mieux régner** prend un sens positif.

## Algorithme et triangles

### Exercice 8 :

On considère l'algorithme suivant où  $x_A, y_A, x_B, y_B, x_C$  et  $y_C$  sont des nombres réels :

```
d1 ← √((xB - xA)2 + (yB - yA)2)
d2 ← √((xC - xA)2 + (yC - yA)2)
Si d1 = d2 alors :
    reponse ← Vrai
Sinon :
    reponse ← Faux
FinSi
```

a. Déterminer par le calcul ce que contiendra la variable reponse à la fin de cet algorithme dans les deux cas suivants :

- $A(2; -2), B(1; 2)$  et  $C(-2; -1)$

.....

.....

.....

.....

.....

.....

- $A(-2; 4), B(1; 2)$  et  $C(-2; -1)$

.....

.....

.....

.....

.....

.....

b. Quel est le rôle de cet algorithme ?

.....

.....

c. Sur quelle droite remarquable se trouve le point A lorsque reponse contient la valeur Vrai ?

.....

## 6 Algorithmique et Informatique

### Algorithmique

Nous avons vu dans le chapitre 1, qu'il y a plus de deux millénaires, Euclide disposait déjà d'un algorithme lui délivrant le quotient et le reste d'une division euclidienne.

Al-Khwarizmi, un millénaire plus tard, vivait lui aussi dans une époque sans informatique. Cela ne l'empêchait en rien de concevoir des algorithmes puis de les mettre en œuvre mentalement ou par écrit.

Dans ce chapitre, nous avons écrit un algorithme dans le langage Python. Mais avant cela, nous avons conçu mentalement un algorithme, indépendant de Python, et qui aurait pu être verbalisé ainsi :

- Recevoir l'âge
- Comparer l'âge et 18
- Retourner le plus grand



#### Définition :

- Un **algorithme** <sup>(2)</sup> est un **processus systématique de résolution de problème**.



#### Définition :

- L'**Algorithmique** <sup>(3)</sup> est la science qui étudie et conçoit des algorithmes.

### Informatique

Nous avons conçu un algorithme, mais contrairement à Euclide ou Al-Khwarizmi, nous ne l'avons pas exécuté mentalement ou par écrit. Nous avons commencé par le traduire en Python, puis, nous avons confié à l'Informatique le soin de l'exécuter.

L'informatique offre ainsi à l'humain de 2019 des capacités de calculs, d'expérimentations et d'apprentissages infiniment supérieures à celles du cerveau seul.



#### Définition :

- **Implémenter** <sup>(4)</sup> un algorithme, c'est le traduire dans un langage de programmation.



#### Définition :

- Un **programme informatique** <sup>(5)</sup> est un ensemble d'instructions destinées à être exécutées par un ordinateur.



#### Définition :

- L'**Informatique** <sup>(6)</sup> est la science du traitement automatique de l'information par l'exécution de programmes informatiques sur des machines (ordinateurs, smartphones, robots ...).

(2). <https://fr.wikipedia.org/wiki/Algorithme>

(3). <https://fr.wikipedia.org/wiki/Algorithmique>

(4). <https://fr.wiktionary.org/wiki/implémenter>

(5). [https://fr.wikipedia.org/wiki/Programme\\_informatique](https://fr.wikipedia.org/wiki/Programme_informatique)

(6). <https://fr.wikipedia.org/wiki/Informatique>



### Exercice 9 : verbalisation

Nous venons de dire :



#### Définition :

- L'**Informatique** <sup>(7)</sup> est la science du traitement automatique de l'information par l'exécution de programmes informatiques sur des machines (ordinateurs, smartphones, robots ...).

a. Quelle information avez-vous traitée dans le logiciel de caisse automatique ?

.....

b. Comment avez-vous traité cette information ?

.....

.....

.....

c. Quel était le programme informatique ? (Rappeler uniquement la première ligne).

.....

d. Quelle était la machine ?

.....