



Unité 2 : Débuter la programmation en Python

Compétence 1 : Instruction conditionnelle

Dans cette première leçon de l'unité 2, vous allez découvrir comment écrire et utiliser une instruction conditionnelle en Python.

Objectifs :

- Écrire et utiliser une instruction conditionnelle.
- Réinvestir la notion de fonction en Python.

Dans un programme, il est particulièrement fréquent d'avoir à orienter l'exécution de celui-ci en fonction de **conditions** qui affectent les différentes variables.

Une condition est un énoncé qui peut être **vrai** ou **faux**.

Par exemple : $a = b$ ou bien $a \geq b$ mais aussi n est pair sont des conditions qui sont vérifiées selon les valeurs affectées à ces variables.

Dans un programme, on peut tester une condition et selon que celle-ci est vraie ou fausse, effectuer un traitement ou un autre. On parle alors de **traitement conditionnel**.

```

if condition :
    Instruction A
else :
    Instruction B

```

Conseil à l'enseignant : En langage Python, il n'y a pas d'instruction pour indiquer la fin de l'instruction conditionnelle. C'est l'indentation qui décale vers la droite les instructions A et B.

`elif` est la contraction de `else if`

Pour tester l'égalité de deux valeurs en langage Python, on utilise le signe « == »

Exemple :

Une société de location de voitures propose à ses clients le contrat suivant :

Un forfait de 66 € auquel s'ajoute 0.25 € par kilomètre au-delà de 70 km.

Votre travail consiste à écrire un script qui permette de calculer automatiquement le coût C du contrat en fonction de la distance parcourue.

Langage naturel

$X \leftarrow a$

Si ($0 < X$) et ($X < 70$)


Alors C prend la valeur 66

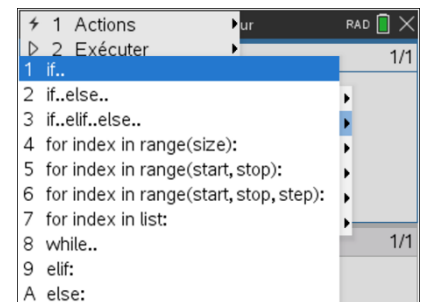
Sinon C prend la valeur $66 + 0.25X$

Fin Si

Conseil à l'enseignant : Prévoir éventuellement le cas où l'utilisateur saisit un nombre X négatif.

Mise en œuvre :

- Démarrer l'application.
- Commencer un nouveau script Python et le nommer « TARIF ». Valider en appuyant sur la touche **enter**.
- Partager la fenêtre en deux parties (éditeur ; console) comme vu dans l'unité 1.
- Appuyer sur la touche  puis choisir **4 Intégrés** et enfin **2 Contrôle**.
- Choisir le menu **3:if..elif..else...**





10 Minutes de Code

TI - NSPIRE™ CX II & TI - PYTHON

- Observer l'indentation du bloc conditionnel.
- Le compléter en utilisant l'algorithme proposé en langage naturel.
- Utiliser la touche `[tab]` afin d'accéder facilement à chaque bloc.

- Vous devriez obtenir le script ci-contre.
- L'instruction `<` et `<=` ainsi que `and` se trouve dans le menu **4 Intégrés** puis **3 Ops.** mais sont également accessibles en appuyant sur les touches `[ctrl]` `[=]`.

- Appuyer sur `[ctrl]` `[B]` afin d'enregistrer le script et de vérifier sa syntaxe.
- Exécuter le script `[ctrl]` `[R]` pour déterminer le coût d'un trajet avec une voiture louée à cette société.

UNITE 2 : COMPETENCE 1

NOTES DU PROFESSEUR

```

1.1 *Classeur RAD 2/8
* Tarif.py
def c(x):
  if ExprBooléen:
    bloc
  elif ExprBooléen:
    bloc
  else:
    bloc
  bloc
Shell Python 1/1
>>>|

```

```

1.1 *Classeur RAD 7/9
* Tarif.py
def c(x):
  if 0<x and x<70:
    c=70
  elif x<=0:
    c=0
  else:
    c=66+(x-70)*0.25
  return c
Shell Python 1/1
>>>|

```

```

1.1 *Classeur RAD 2/9
* Tarif.py
def c(x):
  if 0<x and x<70:
    c=70
  elif x<=0:
    c=0
Shell Python 5/5
>>>from Tarif import *
>>>c(20)
70
>>>|

```

Conseil à l'enseignant : `[ctrl]` `[B]` sauvegarde le script, mais n'enregistre pas le classeur. Appuyer sur `[doc]` afin d'enregistrer le classeur dans un dossier.

Appliquons nos connaissances : Fonction par morceaux

On considère la fonction affine par morceaux f définie par :

$$f(x) = \begin{cases} 2x + 1 & \text{si } x \leq -1 \\ -x + 2 & \text{si } x \in] -1 ; 0] \\ -3x + 2 & \text{si } x > 0 \end{cases}$$

Copier le script ci-contre et l'exécuter afin de compléter le tableau ci-dessous :

x	-4	-1.5	-0.5	-0.1	0.6	2.5	4.8	7.3
$f(x)$								

```

1.1 *Classeur RAD 8/8
* Fct.py
def f(x):
  if x<=-1:
    f=2*x+1
  elif x>-1 and x<=0:
    f=-x+2
  else:
    f=-3*x+2
  return f

```





10 Minutes de Code

TI - NSPIRE™ CX II & TI - PYTHON

Exécuter le script.

UNITE 2 : COMPETENCE 1

NOTES DU PROFESSEUR

```
1.1 1.2 *Classeur RAD 9/9
Shell Python
>>>#Running Fct.py
>>>from Fct import *
>>>f(-4)
-7
>>>f(0)
2
>>>f(3)
-7
>>>
```

Ce document est mis à disposition sous licence Creative Commons
<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>

