



Table des matières

III Instructions conditionnelles	1
1 Si ... alors... sinon	1
2 if test1 : instructions1 elif test2 : instructions2	5

Troisième partie

Instructions conditionnelles

III.1. Si ... alors... sinon



if test
effectue (une fois) les instructions indentées qui suivent lorsque le test est vérifié.



Indentation pour les blocs

- Dans le langage Python, on peut passer des lignes pour plus de clarté, ce qui n'est pas pris en compte lors de l'exécution du programme. Par contre, vous ne pouvez pas ajouter un espace en début de ligne comme bon vous semble, car cela a une signification.
- Indentation
On appelle *indentation* ce décalage d'un cran d'une ou de plusieurs lignes d'un programme. Elle permet de délimiter un bloc d'instructions dans une boucle ou lors d'une exécution conditionnelle.
- Deux points ":"
La ligne précédent l'indentation se finit toujours par deux points. Quand vous appuyez sur la touche après avoir tapé « : », l'indentation est automatiquement effectuée en même temps que le passage à la ligne.
- On peut aussi appuyer sur la touche de tabulation *Tab*, à gauche de la touche "A" pour gérer l'indentation, mais les deux points : sont toujours nécessaires.

1. Le programme ci-dessous cherche à résoudre l'équation : $ax = b$.

```
def soleq(a:float,b:float)->float:  
    '''solution éventuelle de l'équation ax=b'''  
    if a!=0: # si a est différent de 0  
        return b/a
```

Exécutez-le dans la console de droite avec deux valeurs de votre choix. Il permet la résolution de l'équation $ax = b$ avec un test pour savoir si a est bien différent de zéro.

La condition « a différent de zéro » s'écrit « $a!=0$ ». Aucun affichage n'est proposé ici.

2. Si on entre `soleq(0,2)` observer le résultat produit. Pour compléter, on va utiliser la structure « `if... else` »



if test : ... else : ...

if test : ... else : ... : Effectue les instructions indentées lorsque le test est vérifié, sinon effectue les instructions alternatives indentées.

Remarques :

- Le "else" est aligné avec le "if" qui lui correspond.
- Taper ":" puis appuyer sur la touche "entrée" pour passer à la ligne, provoque l'indentation automatique.

```
def soleq(a:...,b:...)->:
    """
    if a!=0: # si a est différent de 0
        return b/a
    else:
        if b==0:
            return "R"
        else:
            return "pas de solution"
```



Exercice 1

On cherche maintenant à résoudre une **équation de type $ax + b = c$** .

1. Commencer par résoudre, à la main, l'équation $2x + 3 = 4$.
2. Proposer une fonction nommée soleq2(a,b,c) renvoyant la solution de cette équation.
3. Vérifier votre programme avec l'équation de la question 1.



Les Conditions du test

- **a!=0** : La condition « *a* différent de zéro » s'écrit « *a!=0* ».
- **a == 0** : La condition « *a* égal à zéro » s'écrit « *a == 0* », (avec deux fois le symbole =).
- **< et >** : Ces symboles désignent les inégalités strictes habituelles.
- **<= et >=** : Ces combinaisons de symboles désignent les inégalités larges \leq et \geq habituelles.
- **and et or**
 - **and** : Permet d'effectuer une instruction si deux tests sont vérifiés simultanément.
 - **or** : Permet d'effectuer une instruction si au moins un test sur deux est vérifié.

III.1.1 Un programme de calcul



Exercice 2

Voici un programme de calcul :

- Choisir un nombre entier;
- Si il est pair, le diviser par 2;
- Sinon le multiplier par 3 et ajouter 1.
- Renvoyer le résultat.

1. Appliquer ce programme de calcul à 8 et à 11 (à la main).
2. Écrire un algorithme sous forme d'une fonction de paramètre *n* correspondant à ce programme de calcul. On pourra compléter le programme ci-après.
3. Que se passe-t-il lorsqu'on entre un décimal? Pourquoi?
4. Élaborer une stratégie pour éviter cette erreur et modifier le programme en conséquence.

**Aide**

$a \% b$: Renvoie le reste de la division euclidienne de a par b . Et si un nombre est pair, le reste de sa division euclidienne par 2 est nul ...

Par exemple $17 = 7 \times 2 + 3$ donc le reste de la division euclidienne de 17 par 7 est 3. Donc :

```
>>>17%7  
3
```

```
def programme(n:int)->...:  
    ''' ... '''  
    if ...  
        return ...  
    else:  
        return ...
```

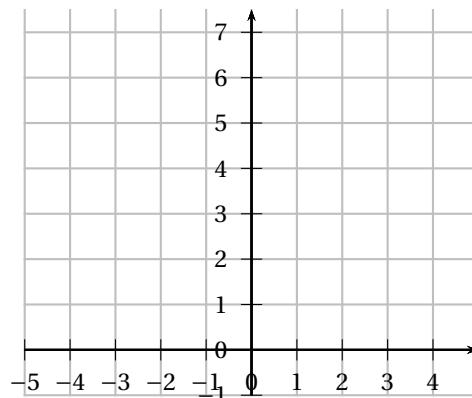
III.1.2 Une fonction définie par morceaux et Géogébra

Exercice 3

1. Écrire un programme utilisant une fonction qui permet de calculer l'image d'une valeur demandée, par la fonction f définie sur \mathbb{R} par :

$$f_{20}: x \mapsto f_{20}(x) = \begin{cases} -2x & \text{si } x < 0 \\ x^2 & \text{si } x \geq 0 \end{cases}$$

```
def f20 (x: ...) -> ...:
    if ...
        return ...
    else:
        return ...
```



2. Compléter alors la tableau de valeurs suivant et tracer \mathcal{C}_f dans le repère ci-contre.

x	-3	-1	0	0,5	1	1,5	2	2,5
$f_{20}(x)$								

III.1.3 Des photocopies

Exercice 4

Un magasin de reprographie propose un tarif dégressif. Les 20 premières photocopies sont facturées à 10 centimes et les suivantes à 8 centimes.

1. Calculer à la main le coût de 15 puis de 30 photocopies.
2. Écrire un algorithme utilisant une fonction qui renvoie le montant de la facture en euros pour un nombre de photocopies donné.
3. Écrire l'expression de cette fonction en fonction des valeurs de x .

$$facture: n \mapsto facture(n) = \begin{cases} \dots & \text{si } \dots \\ \dots & \text{si } \dots \end{cases}$$

III.2. if test1 : instructions1 elif test2 : instructions2



if test1 : instructions1 elif test2 : instructions2

if test1 : instructions1 elif test2 : instructions2 : Effectue les instructions1 indentées lorsque le test1 est vérifié, sinon effectue le test2 et, si celui-ci est vérifié, effectue les instructions2 indentées.

Remarques :

- *On peut enchaîner autant de "elif" que nécessaire.*
- *il faut terminer une série de "elif" par un "else" afin d'être sûr de traiter tous les cas.*



Exercice 5

1. On considère le programme ci-dessous qui calcule l'image d'un nombre x par une fonction h définie par morceaux sur \mathbb{R} . Écrire l'expression de cette fonction en fonction des valeurs de x .

$$h: x \longmapsto h(x) = \begin{cases} \dots & \text{si } \dots \\ \dots & \text{si } \dots \\ \dots & \text{si } \dots \end{cases}$$

2. Exécuter ce programme (dans la console) pour différentes valeurs de x puis tracer la courbe représentative de cette fonction sur l'intervalle $[-5; 5]$ dans un repère de votre choix. *Vous pourrez utiliser le logiciel Geogébra.*

```
def h(x:float) -> float:
    '''image de x par h'''
    if x<0:
        return 2*x+3
    elif x<2: # Donc x >=0 et x<2 soit 0 <= x < 2
        return 3-x
    else: # cas où x >= 2
        return x**2-3
```

III.2.1 Résoudre $x^2 = a$



Exercice 6

- 1 Compléter le programme ci-dessous pour résoudre l'équation $x^2 = a$.

```
from math import sqrt

def resolxcarre(a:...) -> ...:
    """
    ...
    if a<0:
        return ...
    elif ...:
        return ...
    else:
        return ...
```