

1 Introduction

Langage informatique

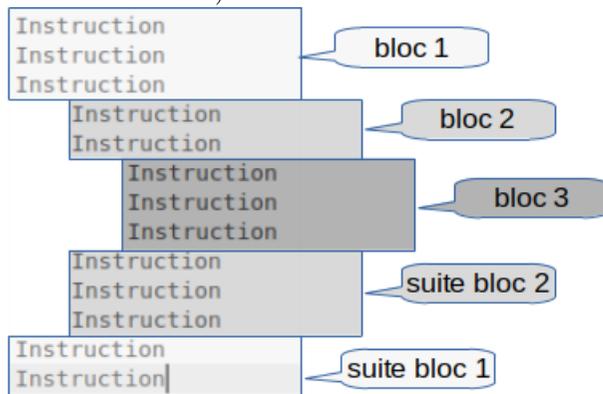
Un langage informatique permet de traduire un algorithme en une série d'instructions compréhensibles et exécutables par un ordinateur. Pour cela, les instructions écrites dans le langage informatique doivent respecter des spécificités et notamment une syntaxe plus précise et plus compliquée que celle d'un langage naturel.

Quelques principes de base à propos du langage python

- Par défaut, le langage python n'inclut pas bon nombre de fonctions mathématiques. C'est pour cela que certains scripts python commenceront par l'instruction suivante qui indique à python de charger les fonctions mathématiques standard :

```
from math import*
```

- Les débuts et fins de bloc d'instructions se marquent en langage python par un décalage horizontal (appelé indentation) et pas par des instructions particulières. Il faut donc faire très attention en python à ce que les instructions d'un même bloc soient écrites avec le même décalage horizontal par rapport à ce qui précède et suit ce bloc (on dit que ces instructions doivent avoir la même indentation).



2 Équivalences entre langage naturel et python

2.1 Affectation d'une valeur à une variable x

Langage naturel	Équivalent en python
x ← valeur	x = valeur

2.2 Lire la valeur d'une variable x entrée au clavier

Si x est un entier (et si l'algorithme ne manipule que des entiers) :

Langage naturel	Équivalent en python
LIRE x	x = int(input("x?"))

Si x est un réel :

Langage naturel	Équivalent en python
LIRE x	x = float(input("x?"))

Si on a besoin d'entrer une valeur résultant d'un calcul mathématique (fraction, racine carrée...) :

Langage naturel	Équivalent en python
LIRE x	from math import* x = eval(input("x?"))

2.3 Afficher la valeur d'une variable x

Langage naturel	Équivalent en python
AFFICHER x	print(x)

2.4 Afficher un message

Langage naturel	Équivalent en python
AFFICHER "Message"	print("message")

2.5 Les opérations mathématiques de base

Langage naturel	Équivalent en python
resultat ← a+b	from math import* resultat=a+b
resultat ← a-b	resultat=a-b
resultat ← a × b	resultat=a*b
resultat ← $\frac{a}{b}$	resultat=a/b
resultat ← a ²	resultat=a**2
resultat ← \sqrt{a}	resultat=sqrt(a)

Pour a et a entiers	Équivalent en python
reste de la division euclidienne de a par b	a%b
quotient de la division euclidienne de a par b	a//b

► Exercice n°1

Compléter le code du script python ci-dessous pour qu'il corresponde à l'algorithme en langage naturel.

Langage naturel	Équivalent en python
LIRE distance	distance=float(input("distance?"))
LIRE temps	temps=.....
vitesse ← $\frac{\text{distance}}{\text{temps}}$	vitesse=.....
AFFICHER vitesse	print(vitesse)

► Exercice n°2

Compléter le script python ci-dessous pour qu'il affiche la valeur de la diagonale d'un rectangle après avoir entré sa largeur et sa longueur :

Script python
<pre>from math import* largeur=float(input("largeur?")) longueur=..... diagonale=..... print(.....)</pre>

2.6 Instructions conditionnelles

• Avec un SI...ALORS

Langage naturel
<i>Instructions précédant le bloc si...alors</i> SI condition ALORS <i>Instructions exécutées si la condition est vérifiée</i> FIN_SI <i>Instructions suivant le bloc si...alors</i>

Équivalent en python
<div style="border: 1px solid gray; border-radius: 50%; padding: 2px; width: fit-content; margin-bottom: 5px;">Le : est indispensable</div> <pre>Instructions précédant le bloc si...alors if condition: Instructions exécutées si la condition est vérifiée Instructions suivant le bloc si...alors</pre> <div style="border: 1px solid gray; border-radius: 50%; padding: 2px; width: fit-content; margin-top: 10px;"> Toutes les instructions décalées (indentées) qui suivent la ligne du if seront exécutées si la condition est vérifiée </div>

► Exemple :

Langage naturel	Équivalent en python
LIRE x SI x>0 ALORS AFFICHER "x est strictement positif" FIN_SI AFFICHER "son opposé est" AFFICHER -x	<pre>x=float(input("x?")) if x>0: print("x est strictement positif") print("son opposé est",-x)</pre>

• Avec un SI...ALORS...SINON

Langage naturel
<i>Instructions précédant le bloc si...alors...sinon</i> SI condition ALORS <i>Instructions exécutées si la condition est vérifiée</i> SINON <i>Instructions exécutées si la condition n'est pas vérifiée</i> FIN_SI <i>Instructions suivant le bloc si...alors...sinon</i>

Équivalent en python
<div style="border: 1px solid gray; border-radius: 50%; padding: 2px; width: fit-content; margin-bottom: 5px;">Le : est indispensable</div> <div style="border: 1px solid gray; border-radius: 50%; padding: 2px; width: fit-content; margin-bottom: 5px; margin-left: 100px;">Toutes les instructions décalées (indentées) qui suivent la ligne du if seront exécutées si la condition est vérifiée</div> <pre>Instructions précédant le bloc si...alors...sinon if condition: Instructions exécutées si la condition est vérifiée else: Instructions exécutées si la condition n'est pas vérifiée Instructions suivant le bloc si...alors...sinon</pre> <div style="border: 1px solid gray; border-radius: 50%; padding: 2px; width: fit-content; margin-top: 10px;">Le : est indispensable</div> <div style="border: 1px solid gray; border-radius: 50%; padding: 2px; width: fit-content; margin-top: 10px; margin-left: 100px;">Toutes les instructions décalées (indentées) qui suivent la ligne du else seront exécutées si la condition n'est pas vérifiée</div>

• Syntaxe python pour les conditions

Situation	Syntaxe python correspondante
Si a égal à b (a et b entiers)	if a==b:
Si a est différent de b (a et b entiers)	if a!=b:
Si a est strictement supérieur à b	if a>b:
Si a est strictement inférieur à b	if a<b:
Si a est inférieur ou égal à b	if a<=b:
Si a est supérieur ou égal à b	if a>=b:

Cas particulier de la comparaison de deux réels de type float (qui ne sont pas représentés de façon exacte en informatique) :

Situation mathématique	Syntaxe pour python (>=3.5)
Si a égal à b (a et b flottants)	from math import*
Si a est différent de b (a et b flottants)	if isclose(a,b):
	if not(isclose(a,b)):

► Exercice n°3

Compléter le code du script python ci-dessous pour qu'il corresponde à l'algorithme en langage naturel.

Langage naturel	Équivalent en python
LIRE age	
SI age>=18 ALORS	age=int(input("age?"))
AFFICHER "majeur"
FIN_SI

► Exercice n°4

Parmi les quatre scripts python ci-dessous, déterminer le seul valide indiquant le plus grand de deux flottants.

Script 1	Script 2
a=int(input("a?")) b=int(input("b?")) if a>b print("le plus grand est a") else: print("le plus grand est b")	a=int(input("a?")) b=int(input("b?")) if a>b: print("le plus grand est b") else: print("le plus grand est a")
Script 3	Script 4
a=int(input("a?")) b=int(input("b?")) if a>b: print("le plus grand est a") else: print("le plus grand est b")	a=int(input("a?")) b=int(input("b?")) if a>b: print("le plus grand est a") else: print("le plus grand est b")

• Opérateurs ET/OU/CONTRAIRE dans les conditions

Situation	Syntaxe python correspondante
Si condition1 OU condition2	if condition1 or condition2:
Si condition1 ET condition2	if condition1 and condition2:
Si CONTRAIRE(condition)	if not(condition):

► Remarque : or, and et not doivent être en minuscules

► Exemple :

Langage naturel	Équivalent en python
LIRE x	x=float(input("x?"))
SI x>0 ET x<1 ALORS	if x>0 and x<1:
AFFICHER "x est dans]0;1["	print("x est dans]0;1[")
FIN_SI	

► Exercice n°5

On considère l'expression $A(x) = \frac{1}{(x-1)\sqrt{x}}$ où x est un réel.

Compléter le code du script python ci-dessous pour qu'il corresponde à l'algorithme en langage naturel.

Langage naturel	Équivalent en python
LIRE x	from math import*
SI x>0 ET x≠1 ALORS	x=.....
AFFICHER "A(x) existe"	if.....
SINON
AFFICHER "A(x) n'existe pas"	else:
FIN_SI

2.7 Boucles POUR...DE...A

• Rappel : ces boucles permettent de répéter des instructions un certain nombre (connu par avance) de fois.

Langage naturel
Instructions précédant le bloc pour...de...a
POUR i ALLANT_DE 1 A 10
Instructions qui seront répétées 10 fois
.....
FIN_POUR
Instructions suivant le bloc pour...de...a

Équivalent en python

ATTENTION : il faut ajouter 1 par rapport au langage naturel. range(1,11) indique tous les entiers entre 1 compris et 11 exclu.

Le : est indispensable

Instructions précédant le bloc pour...de...a

```
for i in range(1,11):
```

Instructions qui seront répétées 10 fois

.....

Instructions suivant le bloc pour...de...a

Toutes les instructions décalées (indentées) qui suivent la ligne du for seront exécutées le nombre de fois voulu

► **Exemple :**

Langage naturel	Équivalent en python
POUR i ALLANT_DE 1 A 10 AFFICHER 7×i FIN_POUR AFFICHER "script terminé"	for i in range(1,11): print(7*i) print("script terminé")

► **Exercice n°6**

Compléter le code du script python ci-dessous pour qu'il corresponde à l'algorithme en langage naturel dont le but est de calculer la somme $1 + 2 + 3 + \dots + 99 + 100$.

Langage naturel	Équivalent en python
somme ← 0 POUR i ALLANT_DE 1 A 100 somme ← somme+i FIN_POUR AFFICHER somme	somme=0 for i somme=..... print(somme)

► **Exercice n°7**

Parmi les quatre scripts python ci-dessous, déterminer le seul valide qui permet de calculer la somme $\sqrt{1} + \sqrt{2} + \sqrt{3} + \dots + \sqrt{49} + \sqrt{50}$.

Script 1	Script 2
from math import* somme=0 for i in range(1,51) somme=somme+sqrt(i) print(somme)	from math import* somme=0 for i in range(1,51): somme=somme+sqrt(i) print(somme)

Script 3	Script 4
from math import* somme=0 for i in range(1,50): somme=somme+sqrt(i) print(somme)	from math import* somme=0 for i in range(1,51): somme=somme+sqrt(i) print(somme)

2.8 Boucles TANT QUE

- Rappel : ces boucles permettent de répéter des instructions tant qu'une certaine condition est vérifiée. La condition en question s'exprime en python de la même façon qu'avec un if.

Langage naturel

Instructions précédant le bloc tant que

TANT_QUE condition FAIRE

Instructions qui seront répétées tant que la condition sera vérifiée

.....

FIN_TANT_QUE

Instructions suivant le bloc tant que

Équivalent en python

Le : est indispensable

Instructions précédant le bloc tant que

```
while condition:
```

Instructions qui seront répétées tant que la condition sera vérifiée

.....

Instructions suivant le bloc tant que

Toutes les instructions décalées (indentées) qui suivent la ligne du while seront répétées tant que la condition sera vérifiée

► **Exemple :**

Script permettant de déterminer le premier entier dont le cube dépasse 5000.

Langage naturel	Équivalent en python
n ← 0 TANT_QUE n ³ < 5000 FAIRE n ← n+1 FIN_TANT_QUE AFFICHER n	n=0 while n**3<5000: n=n+1 print(n)

► **Exercice n°8**

Compléter le code du script python ci-dessous pour qu'il corresponde à l'algorithme en langage naturel dont le but est de déterminer le premier entier n tel que le produit $1 \times 2 \times 3 \times \dots \times n$ dépasse 10000 .

Langage naturel	Équivalent en python
<pre>n ← 1 produit ← 1 TANT_QUE produit<10000 FAIRE n ← n+1 produit ← produit × n FIN_TANT_QUE AFFICHER n</pre>	<pre>n=1 produit=1 while n=..... produit=..... print(n)</pre>

► **Exercice n°9**

Parmi les quatre scripts python ci-dessous, déterminer le seul valide indiquant le plus petit entier n tel que $0,9^n \leq 0.001$

Script 1	Script 2
<pre>n=0 while 0.9**n<=0.001: n=n+1 print(n)</pre>	<pre>n=0 while 0.9**n>0.001: n=n+1 print(n)</pre>
Script 3	Script 4
<pre>n=0 while 0.9**n<=0.001 n=n+1 print(n)</pre>	<pre>n=0 while 0.9**n>0.001: n=n+1 print(n)</pre>

2.9 Les fonctions en programmation

• Rappel : une fonction en programmation est un bloc d'instructions qui ne sera exécuté que quand on l'appelle. Une fonction dépend en général d'un ou plusieurs paramètres et retourne une valeur dépendant de ces paramètres. L'utilisation d'une fonction n'a en général d'intérêt que si on l'appelle plusieurs fois.

Langage naturel
<pre>FONCTION mafonction(paramètres séparés par des virgules) DEBUT_FONCTION Instructions RETOURNER... FIN_FONCTION</pre>

Équivalent en python
<pre>def mafonction(paramètres séparés par des virgules): Instructions return...</pre> <p>Le : est indispensable</p> <p>Toutes les instructions définissant la fonction doivent être décalées (indentées)</p>

► **Exemple :**

Un cycliste et un piéton se déplacent à une vitesse respective de $18 \text{ km} \cdot \text{h}^{-1}$ et $6 \text{ km} \cdot \text{h}^{-1}$. Ce script donne le nombre de kilomètres parcourus par le cycliste et le piéton durant un même temps t (en heures).

Langage naturel	Équivalent en python
<pre>FONCTION distance(vitesse,temps) DEBUT_FONCTION RETOURNER vitesse × temps FIN_FONCTION DEBUT_ALGORITHME LIRE t AFFICHER distance(42,t) AFFICHER distance(6,t) FIN_ALGORITHME</pre>	<pre>def distance(vitesse,temps): return vitesse*temps t=float(input("t?")) print(distance(42,t)) print(distance(6,t))</pre>

► **Exercice n°10**

La formule de conversion entre la valeur d'une température en degrés fahrenheit et sa valeur en degrés celsius est la suivante :

$$\text{degrés Celsius} = \frac{5 \times (\text{degrés Fahrenheit}) - 160}{9}$$

Compléter le code du script python ci-dessous pour qu'il donne la valeur en degrés celsius d'une température de 100 et de 200 degrés fahrenheit.

Script python
<pre>def celsius(fahrenheit): return print(celsius(100)) print(.....)</pre>

3 Application

► **Exercice n°11**

Compléter le code du script python ci-dessous pour qu'il corresponde à l'algorithme en langage naturel.

Langage naturel	Équivalent en python
POUR i ALLANT_DE 1 A 100 AFFICHER i ³ FIN_POUR	for i print(.....)

► **Exercice n°12**

Que fait ce script python ?

Script python
<pre>from math import* i=0 while i*sqrt(i)<500: i=i+1 print(i)</pre>

Réponse :

► **Exercice n°13**

La valeur d'une voiture était de 20000 euros en 2018. On suppose qu'elle est multipliée par 0,9 chaque année. Compléter le script python ci-dessous pour qu'il indique la première année à partir de laquelle la valeur de la voiture sera inférieure ou égale à 12000 euros.

Script python
<pre>annee=2018 valeur=20000 while valeur=valeur*0.9 annee=annee+1 print(annee)</pre>

► **Exercice n°14**

Un ticket de tramway coûte 1 euro sans abonnement. Avec un abonnement annuel de 30 euros, le ticket ne coûte plus que 0,75 euros. Déterminer parmi les quatre scripts python ci-dessous le seul valide qui affiche, après avoir entré le nombre de tickets achetés n, si prendre un abonnement est rentable :

Script 1	Script 2
<pre>n=int(input("n?")) if 30+0.75*n<n: print("abonnement rentable") else: print("abonnement pas rentable")</pre>	<pre>n=int(input("n?")) if 30+0.75*n<n: print("abonnement rentable") else: print("abonnement pas rentable")</pre>
Script 3	Script 4
<pre>n=int(input("n?")) if 30+0.75*n<n: print("abonnement rentable") else: print("abonnement pas rentable")</pre>	<pre>n=int(input("n?")) if 30+0.75*n>n: print("abonnement rentable") else: print("abonnement pas rentable")</pre>

► **Exercice n°15**

Durant une année un individu a acheté n DVD valant 15 euros pièce et p livres valant 22 euros pièce. Compléter le script python ci-dessous pour qu'il indique si cet individu a dépensé plus de 300 euros en dvd et livres ou non :

Script python
<pre>def depense_totale(n,p): return n=int(input("nombre de dvd?")) p=int(input("nombre de livres?")) if depense_totale(n,p)..... print("plus de 300 euros") else: print("pas plus de 300 euros")</pre>

► **Exercice n°16**

Un automobiliste roule à 90 km · h⁻¹ pendant 2 heures, puis roule à 120 km · h⁻¹. Compléter le script python ci-dessous pour qu'il indique le nombre de kilomètres parcourus en fonction du temps de parcours exprimé en heures :

Script python
<pre>def distance(temps): if temps<=2: return else: return t=float(input("temps?")) print (distance(t))</pre>

4 Corrigé des exercices

► Exercice n°1

Langage naturel	Équivalent en python
LIRE distance LIRE temps $vitesse \leftarrow \frac{distance}{temps}$ AFFICHER vitesse	distance=float(input("distance?")) temps=float(input("temps?")) vitesse=distance/temps print(vitesse)

► Exercice n°2

Script python
<pre>from math import* largeur=float(input("largeur?")) longueur=float(input("longueur?")) diagonale=sqrt(largeur**2+longueur**2) print(diagonale)</pre>

► Exercice n°3

Langage naturel	Équivalent en python
LIRE age SI age>=18 ALORS AFFICHER "majeur" FIN_SI	age=int(input("age?")) if age>=18: print("majeur");

► Exercice n°4

Seul le script 3 est valable.

Dans le script 1, il manque : à la fin du if

Dans le script 2, le message affiché est incohérent avec la condition

Dans le script 4, la dernière ligne est mal indentée

► Exercice n°5

Langage naturel	Équivalent en python
LIRE x SI x>0 ET x≠1 ALORS AFFICHER "A(x) existe" SINON AFFICHER "A(x) n'existe pas" FIN_SI	<pre>from math import* x=float(input("x?")) if x>0 and x!=1: print("A(x) existe") else: print("A(x) n'existe pas")</pre>

► Exercice n°6

Langage naturel	Équivalent en python
somme ← 0 POUR i ALLANT_DE 1 A 100 somme ← somme+i FIN_POUR AFFICHER somme	<pre>somme=0 for i in range(1,100): somme=somme+i print(somme)</pre>

► Exercice n°7

Seul le script 2 est valable.

Dans le script 1, il manque : à la fin du for

Dans le script 3, le range n'est pas bon

Dans le script 4, l'avant dernière ligne est mal indentée

► Exercice n°8

Langage naturel	Équivalent en python
n ← 1 produit ← 1 TANT_QUE produit<10000 FAIRE n ← n+1 produit ← produit × n FIN_TANT_QUE AFFICHER n	<pre>n=1 produit=1 while produit<10000: n=n+1 produit=produit*n print(n)</pre>

► Exercice n°9

Seul le script 4 est valable.

Dans le script 1, la condition du while ne correspond pas à la question

Dans le script 2, l'avant dernière ligne est mal indentée

Dans le script 3, il manque : à la fin du while et la condition du while ne correspond pas à la question

► Exercice n°10

Script python
<pre>def celsius(fahrenheit): return (5*fahrenheit-160)/5 print(celsius(100)) print(celsius(200))</pre>

► Exercice n°11

Langage naturel	Équivalent en python
POUR i ALLANT_DE 1 A 100 AFFICHER i ³ FIN_POUR	<pre>for i in range(1,101): print(i**3)</pre>

► **Exercice n°12**

Il indique le plus petit entier positif i tel $i * \sqrt{i} \geq 500$.

► **Exercice n°13**

Script python

```
annee=2018
valeur=20000
while valeur>12000:
    valeur=valeur*0.9
    annee=annee+1
print(annee)
```

► **Exercice n°14**

Seul le script 1 est valable.

Dans le script 2, il manque : à la fin du else

Dans le script 3, le else est mal indenté

Dans le script 3, la condition du if ne correspond pas à la question

► **Exercice n°15**

Script python

```
def depense_totale(n,p):
    return 15*n+22*p

n=int(input("nombre de dvd?"))
p=int(input("nombre de livres?"))
if depense_totale(n,p)>300:
    print("plus de 300 euros")
else:
    print("pas plus de 300 euros")
```

► **Exercice n°16**

Script python

```
def distance(temps):
    if temps<=2:
        return 90*t
    else:
        return 180+120*(t-2)

t=float(input("temps?"))
print (distance(t))
```