

NOM :

Prénom :

Note : /15

- Ce document est un **TP noté** où des programmes sont à écrire en Python.
- La fiche de la semaine dernière est autorisée mais non indispensable.

PARTIE 1: RÉVISIONS : TABLEAU D'AVANCEMENT DES VARIABLES /2

Bob a écrit l'algorithme suivant. Compléter le tableau des variables afin de suivre l'évolution des valeurs de x , y et z

| INSTRUCTIONS | x | y | z |
|---------------------------|-----|-----|-----|
| $y \leftarrow 20$ | / | 20 | / |
| $x \leftarrow y \times 3$ | | | |
| $x \leftarrow x + y$ | | | |
| $z \leftarrow y - 10$ | | | |
| afficher (z) | ... | | |

PARTIE 2: DÉCOUVERTE DU TEST CONDITIONNEL « SI... SINON... » /4**EX 1. EXERCICE D'APPLICATION**

On considère le programme ci-dessous qui permet de calculer le prix (en euros) d'affranchissement d'une lettre en fonction de sa masse (en grammes).

```
masse = float(input("donne-moi un décimal (masse de la lettre en g) : "))

if masse < 30 :
    prix = 1.15
else :
    prix = 1.95

print(prix)
```

1. Quel sera le prix d'affranchissement d'une lettre pesant 40g? 10g? 30g?
2. Que signifie l'instruction **print**?
3. Que signifie l'instruction **input**?
4. Que signifie l'instruction **if**?
5. Que signifie l'instruction **else**?

DÉF 2. SYNTAXE D'UN « SI... SINON... » EN PYTHON**Syntaxe :**

if *condition* :
 instructions 1

else :
 instructions 2

Exemple :

.... : indentation de 4 espaces

```
if moyenne < 10 :
    .... print("Ta moyenne est insuffisante.")
    .... print("Tu peux mieux faire.")
else :
    .... print("Ta moyenne est satisfaisante.")
```

Ex 3. EXERCICE ÉLÉMENTAIRE

Compléter ci-dessous un programme en Python qui demande de saisir un entier n et affiche suivant le cas : « l'entier dépasse 33 » ou « l'entier ne dépasse pas 33 »

```
n = int(input("entrez un entier n:"))  
  
if n > ... :  
    print(".....")  
  
else :  
    .....
```

Ex 4. EXERCICE : PRIX D'UNE PLACE DE CINÉMA

Les tarifs d'entrée au cinéma sont :

• avant 10 ans : 5 €

• de 10 à 17 ans : 8 €

• adultes : 10 €

1. Compléter le programme en Python suivant afin qu'il demande de saisir un entier age et affiche, suivant le cas, le prix à payer

```
age = int(input("entrez votre âge : "))  
  
if age < .... :  
    prix = .....  
  
else :  
    if ..... :  
        prix = .....  
    else :  
        prix = .....  
  
print("prix à payer =", prix)
```

APPEL

→ Appeler le professeur pour vérification

PARTIE 3: DÉCOUVERTE DE LA BOUCLE « TANT QUE ... »

/6

Ex 1. EXERCICE INTRODUCTIF

On considère l'algorithme ci-dessous ainsi que sa traduction en Python.

Les deux instructions de la boucle sont réalisées indéfiniment, tant que la condition $entier \leq 499$ est vraie.

```
entier ← 5  
tant_que (entier ≤ 499) faire:  
    entier ← entier × 10  
    afficher(entier)  
afficher("fin")
```

↻ boucle

```
entier = 5  
while entier <= 499 :  
    entier = entier * 10  
    print(entier)  
print("fin")
```

↻ boucle

1. Que signifie **while** en Python ?

2. On exécute ce programme. Parmi les propositions suivantes, laquelle s'affiche à l'écran ?

| | | | |
|-----|-----|------|-----|
| 5 | 50 | 50 | 50 |
| 50 | 500 | 500 | fin |
| 500 | fin | 5000 | |
| fin | | fin | |

DÉF 2. À RETENIR : SYNTAXE D'UN « TANT QUE » EN PYTHON

Syntaxe :

```
while condition :  
    instruction 1  
    instruction 2  
    ...
```

Exemple :

```
while x > y :  
    a = x  
    b = x + 1  
    c = y + 8
```

_____ : indentation de 4 espaces

ne pas oublier les :

Ex 3. EXERCICE DE TRADUCTION

1. Compléter les pointillés afin de traduire l'algorithme suivant en Python

| Algorithme | Code Python associé |
|-------------------------------|--------------------------|
| afficher (« bonjour ») | print ("bonjour") |
| $a \leftarrow 10$ | $a = 10$ |
| tant que $a < 1000$: | ... |
| $a \leftarrow a + 2$ | ... |
| $a \leftarrow a \times 10$ | ... |
| afficher ("a=", a) | ... |
| afficher (« fin ») | ... |

2. Un élève a écrit le programme précédent puis l'exécute.

Parmi les propositions suivantes, laquelle s'affiche à l'écran ?

| | | | | | |
|--|---|---|--|--|---|
| bonjour $a=12$ fin | bonjour $a=120$ fin | bonjour $a=122$ fin | bonjour $a=1200$ fin | bonjour $a=1220$ fin | bonjour $a=12200$ fin |
|--|---|---|--|--|---|

Ex 4. EXERCICE DE MODIFICATION

Le programme suivant permet de déterminer le plus petit entier n dont le triple dépasse 10000.

```
1 n = 0  
2 while n*3 <= 10000 :  
3     n = n + 1  
4 print("n =", n)
```

1. Modifier ci-dessous la ligne 2 du programme précédent afin de déterminer le plus petit entier n dont le double dépasse 50000.

```
2 while ..... :
```

1. Allumer l'ordinateur et se connecter à son compte avec les identifiants donnés par son professeur principal.
2. Lancer un navigateur internet (Chrome) et se rendre sur le site <https://console.basthon.fr>
3. Saisir le programme suivant dans la fenêtre de gauche en faisant **très attention** au respect des majuscules, minuscules, les deux points et des espaces (indentations).

```
entier = 5
while entier <= 499 :
    entier = entier * 10
    print(entier)
print("fin")
```

4. Exécuter ce programme.
5. Vérifier que la réponse correspond à celle donnée au début de la **partie 3**.

APPEL

→ Appeler le professeur pour vérification

PARTIE 5: BONUS : UN PROBLÈME DE RECHERCHE

Une feuille de papier mesure 0,1 mm d'épaisseur.

On plie cette feuille en deux, puis encore en deux, puis encore en deux, etc...

1. Compléter le tableau suivant afin de comprendre le principe de cet algorithme.

| nombre de plis | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|--------------------------|-----|-----|-----|-----|-----|---|---|
| épaisseur totale (en mm) | 0,1 | 0,2 | 0,4 | 0,8 | 1,6 | | |

2. Écrire, ci dessous, un algorithme qui calcule et affiche au bout de combien de plis, l'épaisseur obtenue dépassera 1 km (soit 1000000 mm.)

```
n = 1 # n est le nombre de plis
epaisseur = 0.1 # epaisseur totale en mm

while ..... :
    n = n + 1
    epaisseur = epaisseur * ...

print("nombre de plis", n)
```

3. Saisir ce programme sur ordinateur et le tester.

Résumé de la séance du jour

- En python, **if** signifie ...
- En python, **else** signifie ...
- En python, **while** signifie ...

En python :

... *moyenne* < 10 :

```
.....("moyenne est insuffisante")
```

else :

```
..... print("moyenne est satisfaisante")
```

while *moyenne* < 15 :

```
..... moyenne = ...
```

En langage naturel :

si *moyenne* < 10 :

afficher ("moyenne est insuffisante")

..... :

.....("moyenne est satisfaisante")

..... *moyenne* < 15 :

moyenne = *moyenne* × 2