

# Algorithmes et Pensée Computationnelle

## Programmation de base - suite

Le but de cette séance est d'aborder des notions de base en programmation et de consolider les connaissances acquises lors de la séance de TP précédente. Au terme de cette séance, l'étudiant sera capable de :

- interagir avec un utilisateur à travers des inputs et outputs,
- définir et manipuler des variables en Java et Python,
- manipuler des chaînes de caractères,
- utiliser des branchements conditionnels.

## 1 Input / Output

### Question 1: (🕒 5 minutes) Output (Java ou Python)

Créez une variable *nom* (str) contenant votre nom, et une autre *prenom* (str) contenant votre prénom puis affichez : "Bonjour, *prenom nom*".

#### 💡 Conseil

Utilisez la fonction `print()` de Python et `System.out.println()` de Java.

#### >\_ Solution

##### Python :

```
1 prenom = "John"
2 nom = "Doe"
3 print("Bonjour, " + prenom + " " + nom)
```

##### Java :

```
1 String prenom = "John";
2 String nom = "Doe";
3 System.out.println("Bonjour, " + prenom + " " + nom);
```

### Question 2: (🕒 5 minutes) Input (Java ou Python)

En vous référant à l'exercice précédent (**Output (Java ou Python)**), demandez à l'utilisateur d'entrer son nom et son prénom via la fonction `input()` au lieu d'initialiser vous-même les variables.

#### 💡 Conseil

Utilisez la fonction `input()` en Python, la classe `Scanner()` en Java (n'oubliez pas d'ajouter `import java.util.Scanner;` tout au début de votre code.

### >\_ Solution

#### Python :

```
1 prenom = input("Quel est votre prénom ?")
2 nom = input("Quel est votre nom ?")
3 print("Bonjour, " + prenom + " " + nom)
```

#### Java :

```
1 Scanner my_scanner = new Scanner(System.in);
2
3 System.out.println("Entrez votre Prénom : ");
4 String prenom = my_scanner.nextLine();
5
6 System.out.println("Entrez votre Nom : ");
7 String nom = my_scanner.nextLine();
8 System.out.println("Bonjour, " + prenom + " " + nom);
```

## 2 Utilisation de variables

### Question 3: (🕒 3 minutes) Type (Python uniquement)

Déclarez deux variables *nom* (String) et *age* (int), puis affichez le type de chacune de ces deux variables.

#### 💡 Conseil

Vous pouvez contrôler le type de vos variables via la fonction `type()`.

### >\_ Solution

```
1 nom = "John"
2 age = 23
3 print(type(nom))
4 print(type(age))
```

### Question 4: (🕒 5 minutes) Conversion des variables (Type casting) (Java ou Python)

Il est possible de convertir une variable d'un certain type vers un autre type. Il est par exemple possible de changer un `int` en `float` ou un `float` en `int`. Cette opération se nomme le *Type Casting*.

Déclarez une variable *nombre\_entier* de type `int`, puis une autre variable *nombre\_decimal* de type `float`. Affichez *nombre\_entier* en le convertissant en `float` et *nombre\_decimal* en le convertissant en `int`.

#### 💡 Conseil

Utilisez la fonction `int(float)` et `float(int)` en Python / Utilisez `(int) float` et `(float) int` en Java.

### >\_ Solution

#### Python :

```
1 nombre_entier = 0
2 nombre_decimal = 3.14
3 print(float(nombre_entier))
4 print(int(nombre_decimal))
```

#### Java :

```
1 int nombre_entier = 0;
2 float nombre_decimal = 3.14f;
3 System.out.println((float) nombre_entier);
4 System.out.println((int) nombre_decimal);
```

### Question 5: (🕒 3 minutes) Calculs (multiplication) (Java ou Python)

Créez 2 variables *facteur\_1* (= 11) et *facteur\_2* (= 3). Multipliez la première variable par la deuxième et stockez le résultat dans une nouvelle variable *produit*. Vous pouvez afficher les différentes variables pour voir leurs valeurs. Vous pouvez répéter l'exercice avec l'addition et la soustraction.

#### 💡 Conseil

L'opérateur de multiplication est le \*, celui d'addition est le + et celui de soustraction est le -.

### >\_ Solution

#### Python :

```
1 facteur_1 = 11
2 facteur_2 = 3
3 produit = facteur_1*facteur_2
4 print(facteur_1)
5 print(facteur_2)
6 print(produit)
```

#### Java :

```
1 int facteur_1 = 11;
2 int facteur_2 = 3;
3 int produit = facteur_1*facteur_2;
4 System.out.println(facteur_1);
5 System.out.println(facteur_2);
6 System.out.println(produit);
```

### Question 6: (🕒 10 minutes) Calculs (division) (Java ou Python)

Créez 2 variables *nb\_bonbons* avec pour valeur 11 et *nb\_personnes* avec pour valeur 3. Divisez la première variable par la deuxième et stockez le résultat dans une nouvelle variable *bonbons\_personnes*. Pour finir, calculez le nombre de bonbons restants via l'opérateur % (modulo) et stockez le résultat dans une nouvelle variable *reste*. Vous pouvez afficher les différentes variables pour voir leurs valeurs.

#### 💡 Conseil

Attention, en Python il existe 2 opérateurs de division, / effectue une division classique, tandis que // effectue une division entière. En Java, si vous travaillez uniquement avec des int, / effectuera une division entière tandis que si vous travaillez avec au moins un float, / effectuera une division classique. Vous pouvez aussi formater le type du résultat lorsque vous créez une variable.

## >\_ Solution

### Python :

```
1 #1
2 nb_bonbons = 11
3 nb_personnes = 3
4 bonbons_personnes = nb_bonbons // nb_personnes
5 reste = nb_bonbons % nb_personnes
6 print(nb_bonbons)
7 print(nb_personnes)
8 print(bonbons_personnes)
9 print(reste)
10
11 #2
12 nb_bonbons = 11
13 nb_personnes = 3
14 bonbons_personnes = nb_bonbons / nb_personnes
15 print(nb_bonbons)
16 print(nb_personnes)
17 print(bonbons_personnes)
```

### Java :

```
1 // 1
2 int nb_bonbons = 11;
3 int nb_personnes = 3;
4 int bonbons_personnes = nb_bonbons / nb_personnes;
5 int reste = nb_bonbons % nb_personnes;
6 System.out.println(nb_bonbons);
7 System.out.println(nb_personnes);
8 System.out.println(bonbons_personnes);
9 System.out.println(reste);
10
11 // 2
12 float nb_bonbons = 11;
13 int nb_personnes = 3;
14 float bonbons_personnes = nb_bonbons / nb_personnes;
15 System.out.println(nb_bonbons);
16 System.out.println(nb_personnes);
17 System.out.println(bonbons_personnes);
```

### Question 7: (🕒 5 minutes) Calculs (incrémentation / décrémentation) (Java ou Python)

Gardez vos variables de l'exercice précédent (**Calculs (division) (Java ou Python)**), augmentez la valeur de *nb\_bonbons* de 1, et diminuez celle de *nb\_personnes* de 1.

### 💡 Conseil

Vous pouvez utiliser les opérateurs += et -= en Python, et les opérateurs ++ et -- en Java.

### >\_ Solution

#### Python :

```
1 nb_bonbons = 11
2 nb_personnes = 3
3 nb_bonbons += 1
4 nb_personnes -= 1
5 bonbons_personnes = nb_bonbons // nb_personnes
6 reste = nb_bonbons % nb_personnes
7 print(nb_bonbons)
8 print(nb_personnes)
9 print(bonbons_personnes)
10 print(reste)
```

#### Java :

```
1 int nb_bonbons = 11;
2 int nb_personnes = 3;
3 nb_bonbons++;
4 nb_personnes--;
5 int bonbons_personnes = nb_bonbons / nb_personnes;
6 int reste = nb_bonbons % nb_personnes;
7 System.out.println(nb_bonbons);
8 System.out.println(nb_personnes);
9 System.out.println(bonbons_personnes);
10 System.out.println(reste);
```

## 3 Manipulation de chaînes de caractères

### Question 8: (🕒 5 minutes) Format d'impression (Python uniquement)

Créez et assignez des valeurs à 2 variables *prenom* (str) et *age* (int), puis affichez : "Je m'appelle *prenom* et j'ai *age* ans". Gérez le format de l'impression via l'opérateur +, puis en utilisant la fonction `format()`.

#### 💡 Conseil

N'hésitez pas à consulter ce lien pour plus de détails concernant l'utilisation de la fonction `format()` : <https://docs.python.org/fr/3/library/stdtypes.html#str.format>

### >\_ Solution

```
1 prenom = input("Quel est votre prénom ?")
2 age = input("Quel est votre age ?")
3 print("Bonjour, je m'appelle " + prenom + " et j'ai " + age + " ans.")
4 print("Bonjour, je m'appelle {0} et j'ai {1} ans.".format(prenom,age))
```

### Question 9: (🕒 5 minutes) Manipulation des chaînes de caractères (indexation) (Java ou Python)

Créez une variable *mon\_mot* de type chaîne de caractères avec pour valeur "Hard But Cool!!". Créez ensuite une variable *premiere* contenant la première lettre de *mon\_mot* en utilisant l'indexation. Créez enfin une variable *derniere* contenant la dernière lettre de *mon\_mot* en utilisant l'indexation. Affichez les résultats. Qu'obtenez-vous ?

#### 💡 Conseil

Pour Python, utilisez `[]`, et pour Java, utilisez la fonction `substring()` ainsi que la fonction `length()` qui permet d'obtenir la taille d'un élément.

### >\_ Solution

#### Python :

```
1 mon_mot = "Hard But Cool !!"
2 premiere = mon_mot[0]
3 derniere = mon_mot[-1]
4 print(premiere)
5 print(derniere)
```

#### Java :

```
1 String mon_mot = "Hard But Cool !!";
2 String premiere = mon_mot.substring(0,1);
3 String derniere = mon_mot.substring(mon_mot.length()-1);
4 System.out.println(premiere);
5 System.out.println(derniere);
```

### Question 10: (🕒 5 minutes) Manipulation des chaînes de caractères (indexation 2) (Java ou Python)

Gardez votre variable, *mon\_mot* et créez une variable *lettre\_5* contenant la cinquième lettre de *mon\_mot* en utilisant l'indexation. Créez ensuite une variable *lettre\_9\_13* contenant les lettres 9, 10, 11, 12, 13 de *mon\_mot*. Afficher les résultats et voyez ce que vous obtenez.

### 💡 Conseil

Attention, ici les espaces comptent comme des lettres !

Pour Python, utilisez `[:]`, et pour Java, utilisez la fonction `substring()`.

### >\_ Solution

#### Python :

```
1 mon_mot = "Hard But Cool !!"
2 lettre_5 = mon_mot[4]
3 lettre_9_13 = mon_mot[8:13]
4 print(lettre_5)
5 print(lettre_9_13)
```

#### Java :

```
1 String mon_mot = "Hard But Cool !!";
2 String lettre_5 = mon_mot.substring(4,5);
3 String lettre_9_13 = mon_mot.substring(8,13);
4 System.out.println(lettre_5);
5 System.out.println(lettre_9_13);
```

### Question 11: (🕒 10 minutes) Manipulation des chaînes de caractères (Java ou Python) Optionnel

Il est possible d'obtenir la longueur d'une chaîne de caractère (ou d'une liste ou d'un dictionnaire) en utilisant la fonction `len()`. Gardez votre variable *mon\_mot* et créez une nouvelle variable nommée *ln\_mon\_mot* contenant le nombre de caractère de la variable *mon\_mot*, puis une nouvelle variable *moitie* contenant la première moitié de la variable *mon\_mot* (utilisez la variable que vous venez de créer). Affichez le résultat.

### 💡 Conseil

La fonction présentée dans l'énoncé de la question n'est valable que pour python. L'équivalent pour Java est la fonction `length()`.

## >\_ Solution

### Python :

```
1 mon_mot = "Hard But Cool !!"
2 ln_mon_mot = len(mon_mot)
3 moitie_mon_mot = mon_mot[:ln_mon_mot//2]
4 print(ln_mon_mot)
5 print(moitie_mon_mot)
```

### Java :

```
1 String mon_mot = "Hard But Cool !!";
2 int ln_mon_mot = mon_mot.length();
3 String moitie = mon_mot.substring(0,ln_mon_mot/2);
4 System.out.println(ln_mon_mot);
5 System.out.println(moitie);
```

## 4 Conditions

### Question 12: (🕒 5 minutes) Conditions (python)

Qu'affiche le programme suivant ?

```
1 teacher = "Garbinato"
2 assistant = "Diallo"
3 lesson = "ACT"
4
5 if ((teacher == "Diallo") or not(teacher == "Garbinato")) and (lesson == "ACT"):
6     print("Alpha Diallo et Benoît Garbinato")
7 elif ((teacher == "Benfares") and (lesson == "ACT")) or (teacher == assistant):
8     print("Anass Benfares")
9 elif (teacher == "Gaelle" or assistant == "Diallo") and (lesson == "ACT" and teacher == "Julien"):
10    print("Gaelle Barillère")
11 elif ((teacher == "Garbinato" or lesson == "INF") and assistant == "Diallo") or (lesson == "ACT" or teacher == "Hafsa"):
12    print(teacher + " : Professeur du cours Algorithmique et pensée computationnelle.")
13 else:
14    print("Benoît")
```

#### 💡 Conseil

En Python, la fonction `not` renvoie l'opposé d'une valeur booléenne. Par exemple, `not(False)` renverra `True`. En Java, on utilise un point d'exclamation avant la valeur.

#### >\_ Solution

Garbinato : Professeur du cours Algorithmique et pensée computationnelle.

### Question 13: (🕒 10 minutes) Branchement conditionnel en Java

Qu'affiche le programme suivant ?

```
1 int numero_mois = 7;
2
3 switch(numero_mois) {
4
5     case 1:
6         System.out.println("Janvier");
7         break;
8     case 2:
9         System.out.println("Février");
10        break;
11    case 3:
12        System.out.println("Mars");
13        break;
14    case 4:
15        System.out.println("Avril");
16        break;
17    case 5:
18        System.out.println("Mai");
19        break;
20    case 6:
21        System.out.println("Juin");
22        break;
23    case 7:
24        System.out.println("Juillet");
25        numero_mois = 9 ;
26    case 8:
27        System.out.println("Aout");
28    case 9:
29        if (numero_mois == 8){
30            System.out.println("Septembre");
31            break;
32        }
33        else{
34            System.out.println("Décembre");
35            numero_mois = 13;
36            break;
```



```

37     }
38 case 10:
39     System.out.println("Octobre");
40     break;
41 case 11:
42     System.out.println("Novembre");
43     break;
44 case 12:
45     System.out.println("Décembre");
46     break;
47 default:
48     System.out.println("Ce n'est pas un mois. ");
49 }

```

#### Conseil

- `break` indique que l'on sort de l'accolade. Les cas suivants ne seront pas traités.
- L'absence de `break` indique que l'on va rentrer dans tous les cas suivants, jusqu'à enfin atteindre un `break`.
- Lorsque l'on pose `case n` où `n` est un nombre cela est équivalent au test `n == numero_mois`. Ce test est aussi valable si on cherche à comparer des chaînes de caractères (par exemple si `numero_mois = "Juin"`, à ce moment là `n` sera aussi une chaîne de caractères).

#### >\_ Solution

Juillet  
Aout  
Décembre

Explications :

- Comme la case 7 ne contient pas de `break` et modifie `numero_mois`, la lecture du code va continuer.
- On rentre dans le case 9, qui contient un `break`. Le `numero_mois` sera aussi modifié mais cela ne sera pas important car on sort de l'accolade et les cas suivants ne seront pas traités.