

Algorithmes et Pensée Computationnelle

Fonctions, gestion de la mémoire et des exceptions - Exercices de base

Le but de cette séance est d'approfondir vos connaissances en programmation. Au terme de cette séance, l'étudiant sera capable de :

- Utiliser des bibliothèques contenant des fonctions prédéfinies,
- définir une fonction et l'utiliser dans un programme,
- connaître quelle est la portée d'une variable,
- comprendre comment fonctionne la gestion de la mémoire,
- gérer des exceptions.

1 Variables et Fonctions

Question 1: (🕒 5 minutes) Les fonctions (fonctions basiques) (Java ou Python)

Définissez une fonction nommée `ping()` qui, lorsqu'elle est appelée, affiche "pong". Appelez la plusieurs fois et observez le résultat.

💡 Conseil

- Référez vous aux diapositives du cours pour la création et l'appel des fonctions.
- Vous pourriez utiliser une boucle `for` pour effectuer plusieurs appels à la fonction `ping()`.

>_ Solution

Python :

Java :

Question 2: (🕒 5 minutes) Les Fonctions (Fonction multiplication) (Java ou Python)

Définissez une fonction nommée `multiplicateur()` qui prend deux arguments `multiple_1` et `multiple_2`, les multiplie et retourne le résultat. Stockez le résultat de `multiplicateur(2,3)` dans une variable `resultat` et affichez la.

💡 Conseil

- Référez vous au cours pour la création et l'appel des fonctions.
- Pour retourner une valeur au lieu de l'imprimer, utilisez le mot clé `return` (pour Python et Java).

>_ Solution

Python :

Java :

Question 3: (🕒 5 minutes) Les Fonctions (Fonctions Aire et Périmètre) (Java ou Python)

Définissez deux fonctions nommées `aire()` et `perimetre()` qui prennent un argument (`rayon`) et renvoient respectivement l'aire et le périmètre d'un cercle. Stockez les résultats dans des variables `aire` et `perimetre` et affichez le contenu de ces variables.

Conseil

- Référez vous au cours pour la création et l'appel des fonctions.
- Pour retourner une valeur au lieu de l'imprimer, utilisez le mot clé `return` (pour Python et Java).
- Pour rappel, le périmètre d'un cercle s'obtient en utilisant la formule $P = 2 * \pi * r$ et l'aire s'obtient en utilisant la formule $A = r^2 * \pi$.

>_ Solution

Python :

Java :

Question 4: (🕒 5 minutes) Portée des variables (à compléter)

Exercice 1 - structure

Conseil

>_ Solution

Python :

Java :

2 Gestion de la mémoire

Question 5: (🕒 5 minutes) Template (langage à utiliser)

Exercice 1 - structure

Conseil

>_ Solution

Python :

Java :

3 Gestion des exceptions

Question 6: (🕒 5 minutes) Qu'affiche le programme suivant ? Types d'erreurs (Python)

```
1 value1 = "Algorithms"
2 value2 = 4
3
4 try:
5     size = len(value1)
6     result = size/value2
```

```

7  print(f"Le résultat de la division est: {result}")
8
9  except Exception as error:
10     print("On ne peut pas effectuer l'opération")
11
12  try:
13     result = value1/2
14     print(f"Le résultat de la division est: {result}")
15
16  except TypeError as error:
17     print("On ne peut pas diviser une chaîne de caractère")

```



Conseil

— La fonction len() compte le nombre d'élément dans la variable.



Solution

Le résultat de la division est : 2.5
On ne peut pas diviser une chaîne de caractère

Question 7: (🕒 5 minutes) Qu'affiche le programme suivant ? **Types d'erreurs (Python)**

```

1  value1 = 4
2  value2 = ""
3
4  try:
5     count = len(value2)
6     result = value1/count
7  except ZeroDivisionError as error:
8     print("Nous ne pouvons pas diviser un nombre par 0")

```



Conseil

La chaîne de caractère vide est représentée par ""



Solution

Nous ne pouvons pas diviser un nombre par 0.

Question 8: (🕒 5 minutes) Qu'affiche le programme suivant ? **Types d'erreurs (Python)**

```

1  value1 = "Algorithms"
2  value2 = 4
3
4
5  try:
6     decimal = float(value2)
7  except ValueError as error:
8     print("Nous ne pouvons pas convertir un entier en decimal")
9
10 finally:
11     try:
12         value2 = int(value1)
13     except ValueError as error:
14         print("Nous ne pouvons pas convertir une chaîne de caractère en nombre")

```



Conseil

— La fonction float() permet de convertir une variable en nombre à décimal.
— La fonction int() permet de convertir une variable en nombre entier.

>_ Solution

Nous ne pouvons pas convertir une chaîne de caractère.

Question 9: (🕒 5 minutes) Est-ce que le programme suivant s'exécute correctement ? Si la réponse est non, expliquez pourquoi et comment vous corrigerez le programme ? **Gestion d'erreurs (Java)**

```
1 public class Question9 {
2
3     public static void division(int a, int b) throws ArithmeticException{
4         if(b==0){
5             throw new ArithmeticException();
6         }else{
7             float result = a/b;
8             System.out.println("Le résultat de la division de " + a + "/" + b + " = " + result);
9         }
10
11     }
12 }
13 public static void main(String args[]){
14     int value1 = 2;
15     int value2 = 4;
16     try {
17         division(value1,value2);
18         value2 = 0;
19         division(value1,value2);
20     }catch(IndexOutOfBoundsException err){
21         System.out.println("Nous ne pouvons pas effectuer une division par 0");
22     }
23 }
24
25
26 }
```

💡 Conseil

Référez-vous aux cours pour la gestion d'erreur en Java (slide 26)

>_ Solution

Le programme suivant est faux, le type d'erreur que la fonction `division()` retourne est différent de celui qu'on intercepte dans le `catch`.
Pour corriger cela, il suffit de mettre au lieu de `IndexOutOfBoundsException` le type de l'erreur retourné par la fonction `division()` qui est `ArithmeticException`.