

Quiz Section

Python – Les fonctions

1. Quel est le type de variable qui peut être passé à une fonction en tant que paramètre par défaut ?
 - a) int
 - b) str
 - c) list
 - d) None
2. Quelle est la différence entre les mots-clés ``def`` et ``lambda`` pour définir une fonction en Python ?
 - a) Le mot-clé ``def`` est utilisé pour les fonctions anonymes, tandis que ``lambda`` est utilisé pour les fonctions nommées.
 - b) Le mot-clé ``def`` est utilisé pour les fonctions nommées, tandis que ``lambda`` est utilisé pour les fonctions anonymes.
 - c) Le mot-clé ``def`` est utilisé pour les fonctions qui prennent des paramètres, tandis que ``lambda`` est utilisé pour les fonctions sans paramètres.
 - d) Le mot-clé ``def`` est utilisé pour les fonctions qui retournent une valeur, tandis que ``lambda`` est utilisé pour les fonctions qui ne retournent pas de valeur.
3. Comment appeler une fonction qui est définie à l'intérieur d'une autre fonction en Python ?
 - a) En utilisant le mot-clé ``global``
 - b) En utilisant le mot-clé ``nonlocal``
 - c) En utilisant le mot-clé ``return``
 - d) En utilisant le mot-clé ``self``
4. Quel est l'avantage de l'utilisation de fonctions décorateurs en Python ?
 - a) Ils permettent de répéter du code
 - b) Ils permettent de simplifier le code
 - c) Ils permettent de modifier le comportement d'une fonction sans la modifier
 - d) Ils permettent de supprimer le code
5. Comment passer un argument par référence à une fonction en Python ?
 - a) En utilisant le mot-clé ``pass``
 - b) En utilisant le mot-clé ``return``
 - c) En utilisant le mot-clé ``global``
 - d) En utilisant le mot-clé ``nonlocal``

Sommaire

1. Quel est le nom du premier président de la République française ?

- a) Louis-Napoléon Bonaparte
 - b) Adolphe Thiers
 - c) Léon Gambetta
 - d) Louis-Napoléon Bonaparte
2. Quel est le nom de la déclaration des droits de l'homme et du citoyen adoptée en 1789 ?
- a) Déclaration des droits de l'homme et du citoyen
 - b) Charte constitutionnelle
 - c) Acte additionnel aux constitutions de l'Empire
 - d) Déclaration des droits de l'homme et du citoyen
3. Quel est le nom du roi qui a abdiqué en 1792 ?
- a) Louis XVI
 - b) Louis XVII
 - c) Louis XVIII
 - d) Louis XVI
4. Quel est le nom de la bataille qui a eu lieu en 1792 ?
- a) Bataille de Valmy
 - b) Bataille de Jemappes
 - c) Bataille de Fleurus
 - d) Bataille de Valmy
5. Quel est le nom du dirigeant qui a été exécuté en 1793 ?
- a) Georges Danton
 - b) Paul Barras
 - c) Louis Antoine de Saint-Just
 - d) Georges Danton

Python – Les fonctions

1. Quel est le type de variable qui peut être passé à une fonction en tant que paramètre par défaut ?
- a) int
 - b) str
 - c) list
 - d) None
2. Quelle est la différence entre les mots-clés `def` et `lambda` pour définir une fonction en Python ?
- a) Le mot-clé `def` est utilisé pour les fonctions anonymes, tandis que `lambda` est utilisé pour les fonctions nommées.
 - b) Le mot-clé `def` est utilisé pour les fonctions nommées, tandis que `lambda` est utilisé pour les fonctions anonymes.

- c) Le mot-clé ``def`` est utilisé pour les fonctions qui prennent des paramètres, tandis que ``lambda`` est utilisé pour les fonctions sans paramètres.
- d) Le mot-clé ``def`` est utilisé pour les fonctions qui retournent une valeur, tandis que ``lambda`` est utilisé pour les fonctions qui ne retournent pas de valeur.
3. Comment appeler une fonction qui est définie à l'intérieur d'une autre fonction en Python ?
- a) En utilisant le mot-clé ``global``
 - b) En utilisant le mot-clé ``nonlocal``
 - c) En utilisant le mot-clé ``return``
 - d) En utilisant le mot-clé ``self``
4. Quel est l'avantage de l'utilisation de fonctions décorateurs en Python ?
- a) Ils permettent de répéter du code
 - b) Ils permettent de simplifier le code
 - c) Ils permettent de modifier le comportement d'une fonction sans la modifier
 - d) Ils permettent de supprimer le code
5. Comment passer un argument par référence à une fonction en Python ?
- a) En utilisant le mot-clé ``pass``
 - b) En utilisant le mot-clé ``return``
 - c) En utilisant le mot-clé ``global``
 - d) En utilisant le mot-clé ``nonlocal``

Fonctions

1. Quelle est la fonction de la fonction ``map()`` en programmation ?
- a) Appliquer une fonction à chaque élément d'un tableau
 - b) Faire une boucle sur un tableau
 - c) Récupérer un élément spécifique d'un tableau
 - d) Supprimer un élément d'un tableau
2. Quelle est la fonction de la fonction ``filter()`` en programmation ?
- a) Appliquer une fonction à chaque élément d'un tableau
 - b) Faire une boucle sur un tableau
 - c) Récupérer les éléments d'un tableau qui répondent à une condition
 - d) Supprimer un élément d'un tableau
3. Quelle est la fonction de la fonction ``reduce()`` en programmation ?
- a) Appliquer une fonction à chaque élément d'un tableau
 - b) Faire une boucle sur un tableau
 - c) Récupérer les éléments d'un tableau qui répondent à une condition
 - d) Combinez tous les éléments d'un tableau en un seul élément
4. Quelle est la fonction de la fonction ``sort()`` en programmation ?

- a) Appliquer une fonction à chaque élément d'un tableau
 - b) Faire une boucle sur un tableau
 - c) Récupérer les éléments d'un tableau qui répondent à une condition
 - d) Trier les éléments d'un tableau dans l'ordre croissant ou décroissant
5. Quelle est la fonction de la fonction `forEach()` en programmation ?
- a) Appliquer une fonction à chaque élément d'un tableau
 - b) Faire une boucle sur un tableau
 - c) Récupérer les éléments d'un tableau qui répondent à une condition
 - d) Supprimer un élément d'un tableau

Fonctions mathématiques (module math)

1. Quelle est la fonction mathématique qui retourne le sinus d'un angle en radians ?
- a) `math.cos(x)`
 - b) `math.sin(x)`
 - c) `math.tan(x)`
 - d) `math.asin(x)`
2. Quelle est la fonction mathématique qui retourne le logarithme népérien d'un nombre ?
- a) `math.log10(x)`
 - b) `math.log(x)`
 - c) `math.exp(x)`
 - d) `math.sqrt(x)`
3. Quelle est la fonction mathématique qui retourne la racine carrée d'un nombre ?
- a) `math.pow(x, 0.5)`
 - b) `math.sqrt(x)`
 - c) `math.ceil(x)`
 - d) `math.floor(x)`
4. Quelle est la fonction mathématique qui retourne le nombre pi ?
- a) `math.e`
 - b) `math.pi`
 - c) `math.tau`
 - d) `math.inf`
5. Quelle est la fonction mathématique qui retourne le nombre e (base des logarithmes népériens) ?
- a) `math.e`
 - b) `math.pi`
 - c) `math.tau`
 - d) `math.inf`

Fonctions trigonométriques

1. Quelle est la valeur de $\sin(\pi/6)$?

- a) $1/2$
- b) $\sqrt{3}/2$
- c) $1/\sqrt{3}$
- d) 1

2. Quelle est la valeur de $\cos(\pi/4)$?

- a) $1/\sqrt{2}$
- b) $\sqrt{2}$
- c) $1/2$
- d) 1

3. Quelle est la valeur de $\tan(\pi/3)$?

- a) $\sqrt{3}$
- b) $1/\sqrt{3}$
- c) 1
- d) $\sqrt{3}/2$

4. Quelle est la valeur de $\sin(\pi/2)$?

- a) 0
- b) 1
- c) $1/2$
- d) $\sqrt{3}/2$

5. Quelle est la valeur de $\cos(0)$?

- a) 0
- b) 1
- c) $1/2$
- d) $1/\sqrt{2}$

1. Fonctions

1. Quelle est la fonction de la fonction ``typeof`` en JavaScript ?

- a) Renvoie le type d'un objet
- b) Renvoie la valeur d'un objet
- c) Renvoie la valeur de retour d'une fonction
- d) Renvoie la valeur de la variable

2. Quelle est la différence entre les fonctions ``let`` et ``const`` en JavaScript ?

- a) La fonction ``let`` est utilisée pour déclarer des variables, tandis que la fonction ``const`` est utilisée pour déclarer des constantes
 - b) La fonction ``const`` est utilisée pour déclarer des variables, tandis que la fonction ``let`` est utilisée pour déclarer des constantes
 - c) La fonction ``let`` est utilisée pour déclarer des constantes, tandis que la fonction ``const`` est utilisée pour déclarer des variables
 - d) Les deux fonctions sont utilisées pour déclarer des variables
3. Quelle est la fonction de la fonction ``map()`` en JavaScript ?
- a) Renvoie un nouveau tableau avec les résultats d'une fonction appliquée à chaque élément d'un tableau
 - b) Renvoie un tableau avec les indices des éléments d'un tableau
 - c) Renvoie un tableau avec les valeurs des éléments d'un tableau
 - d) Renvoie un tableau vide
4. Quelle est la différence entre les fonctions ``forEach()`` et ``map()`` en JavaScript ?
- a) La fonction ``forEach()`` applique une fonction à chaque élément d'un tableau, tandis que la fonction ``map()`` renvoie un nouveau tableau avec les résultats
 - b) La fonction ``map()`` applique une fonction à chaque élément d'un tableau, tandis que la fonction ``forEach()`` renvoie un nouveau tableau avec les résultats
 - c) La fonction ``forEach()`` renvoie un nouveau tableau avec les résultats, tandis que la fonction ``map()`` applique une fonction à chaque élément d'un tableau
 - d) Les deux fonctions sont identiques
5. Quelle est la fonction de la fonction ``reduce()`` en JavaScript ?
- a) Renvoie la somme des éléments d'un tableau
 - b) Renvoie le produit des éléments d'un tableau
 - c) Renvoie un objet avec les clés et les valeurs d'un tableau
 - d) Renvoie un tableau avec les indices des éléments d'un tableau

1.1 Principe et généralités

1. Quel est le terme scientifique pour désigner la force qui attire deux objets l'un vers l'autre ?
- a) Force de gravité
 - b) Force électromagnétique
 - c) Force de friction
 - d) Force de tension
2. Quel est le principe fondamental qui décrit la stabilité d'un système isolé ?
- a) Principe d'entropie
 - b) Principe de conservation de l'énergie
 - c) Principe de conservation de la quantité de mouvement
 - d) Principe de stabilité thermodynamique

3. Quel est le type de mouvement qui se produit lorsque l'objet en mouvement conserve sa vitesse et sa direction ?

- a) Mouvement uniforme
- b) Mouvement uniformément accéléré
- c) Mouvement circulaire
- d) Mouvement rectiligne

4. Quel est le terme scientifique pour désigner la mesure de la force qui s'exerce sur un objet lorsqu'il est soumis à une force extérieure ?

- a) Force de friction
- b) Force de tension
- c) Force de résistance
- d) Force de pression

5. Quel est le principe qui décrit la relation entre la vitesse et l'énergie d'un objet en mouvement ?

- a) Principe de conservation de l'énergie
- b) Principe de conservation de la quantité de mouvement
- c) Principe de conservation de la masse
- d) Principe de conservation de la température

En programmation, les fonctions sont très utiles pour réaliser plusieurs fois la même opération au sein d'un programme.

1. Quel est l'avantage principal des fonctions en programmation ?

- a) Elles permettent de répéter des instructions complexes
- b) Elles permettent de réduire la taille du code
- c) Elles permettent de réaliser plusieurs fois la même opération
- d) Elles permettent de gérer les erreurs de manière efficace

2. Qu'est-ce que l'on appelle une fonction en programmation ?

- a) Un bloc d'instructions qui s'exécute automatiquement
- b) Un ensemble de variables qui sont partagées entre les fonctions
- c) Un ensemble d'instructions qui peuvent être exécutées plusieurs fois
- d) Un nom donné à un bloc d'instructions qui peut être réutilisé

3. Pourquoi les fonctions sont-elles utiles en programmation ?

- a) Elles permettent de réduire la complexité du code
- b) Elles permettent de gagner du temps en réécrivant des instructions
- c) Elles permettent de réduire la taille du code et de gagner du temps
- d) Elles permettent de gérer les erreurs de manière efficace

4. Quel est l'exemple le plus simple d'une fonction en programmation ?

- a) Une fonction qui affiche un message d'erreur

- b) Une fonction qui calcule la somme de deux nombres
 - c) Une fonction qui affiche le nom d'un utilisateur
 - d) Une fonction qui gère les erreurs de connexion à une base de données
5. Qu'est-ce que l'on appelle la réutilisation d'une fonction en programmation ?
- a) La réécriture d'une fonction existante
 - b) La suppression d'une fonction inutile
 - c) L'appel d'une fonction à plusieurs reprises
 - d) La modification d'une fonction existante

1.2 Fonctions mathématiques (module math)

1. Quelle est la fonction mathématique qui retourne le sinus d'un angle en radians ?
- a) `math.cos(x)`
 - b) `math.sin(x)`
 - c) `math.tan(x)`
 - d) `math.asin(x)`
2. Quelle est la fonction mathématique qui retourne le logarithme népérien d'un nombre ?
- a) `math.log10(x)`
 - b) `math.log(x)`
 - c) `math.exp(x)`
 - d) `math.sqrt(x)`
3. Quelle est la fonction mathématique qui retourne la racine carrée d'un nombre ?
- a) `math.pow(x, 0.5)`
 - b) `math.sqrt(x)`
 - c) `math.ceil(x)`
 - d) `math.floor(x)`
4. Quelle est la fonction mathématique qui retourne le nombre pi ?
- a) `math.e`
 - b) `math.pi`
 - c) `math.tau`
 - d) `math.inf`
5. Quelle est la fonction mathématique qui retourne le nombre e (base des logarithmes népériens) ?
- a) `math.e`
 - b) `math.pi`
 - c) `math.tau`
 - d) `math.inf`

faire import math

1. Quel est le résultat de 7×9 ?

- a) 63
- b) 65
- c) 67
- d) 69

2. Quel est le résultat de $24 \div 4$?

- a) 5
- b) 6
- c) 8
- d) 10

3. Quel est le résultat de $11 + 3$?

- a) 12
- b) 13
- c) 14
- d) 15

4. Quel est le résultat de $9 - 2$?

- a) 6
- b) 7
- c) 8
- d) 10

5. Quel est le résultat de 6×8 ?

- a) 45
- b) 48
- c) 50
- d) 52

math.floor(-7.6) : partie entière, donne ici -8.0.

1. Quel est le résultat de la fonction `math.floor` appliquée à -7.6 ?

- a) -7.5
- b) -8.0
- c) -7.0
- d) -9.0

2. La fonction `math.floor` arrondit un nombre à la partie entière la plus proche. Quel est le résultat de `math.floor(-3.8)` ?

- a) -3.7

- b) -4.0
- c) -3.0
- d) -5.0

3. Quel est le résultat de `math.floor(2.9)` ?

- a) 2.0
- b) 3.0
- c) 2.9
- d) 1.0

4. La fonction `math.floor` arrondit un nombre à la partie entière la plus proche. Quel est le résultat de `math.floor(-2.2)` ?

- a) -2.1
- b) -2.0
- c) -3.0
- d) -1.0

5. Quel est le résultat de `math.floor(0.5)` ?

- a) 0.0
- b) 1.0
- c) 0.5
- d) -1.0

`int(math.floor(4.5))` : pour avoir l'entier 4.

1. Quel est le résultat de l'expression `math.floor(4.5)` ?

- a) 3
- b) 4
- c) 5
- d) 6

2. L'expression `math.floor(3.7)` équivaut à quel entier ?

- a) 3
- b) 4
- c) 5
- d) 6

3. Quel est le résultat de l'expression `math.floor(2.9)` ?

- a) 2
- b) 3
- c) 4
- d) 5

4. L'expression `math.floor(1.1)` équivaut à quel entier ?

- a) 1
- b) 2
- c) 3
- d) 4

5. Quel est le résultat de l'expression `math.floor(0.5)` ?

- a) 0
- b) 1
- c) 2
- d) 3

1.3 Définition d'une fonction

1. Quel est le rôle de la fonction `return` dans une fonction en Python ?

- a) Permet d'appeler une autre fonction
- b) Permet de sortir de la fonction et renvoyer une valeur
- c) Permet d'assigner une valeur à une variable
- d) Permet d'effectuer une boucle

2. Comment définir une fonction en Python qui prend un argument ?

- a) Utiliser la syntaxe `def fonction() :`
- b) Utiliser la syntaxe `def fonction(argument) :`
- c) Utiliser la syntaxe `function(argument) :`
- d) Utiliser la syntaxe `function() :`

3. Quel est l'avantage de définir une fonction en Python ?

- a) Améliorer la lisibilité du code
- b) Réduire la taille du code
- c) Permettre de réutiliser du code
- d) Tous les avantages ci-dessus

4. Comment appeler une fonction en Python ?

- a) Utiliser la syntaxe `fonction()`
- b) Utiliser la syntaxe `function()`
- c) Utiliser la syntaxe `def fonction()`
- d) Utiliser la syntaxe `function = fonction()`

5. Quel est le type de variable qui peut être passé en argument à une fonction en Python ?

- a) Seulement des nombres
- b) Seulement des chaînes de caractères
- c) Tous les types de variables
- d) Aucun type de variable

Une fonction est définie par une fonction

1. Quelle est la définition d'une fonction en mathématiques ?
 - a) Une relation entre deux variables
 - b) Une règle qui prend une entrée et produit une sortie
 - c) Un ensemble de nombres
 - d) Une équation qui relie deux variables
2. Quel est l'exemple d'une fonction simple ?
 - a) $f(x) = 2x + 1$
 - b) $f(x) = x^2 + 3$
 - c) $f(x) = 5$
 - d) $f(x) = x + 2$
3. Quelle est la propriété d'une fonction qui garantit qu'elle est bien définie ?
 - a) La fonction est définie pour tous les nombres réels
 - b) La fonction est définie pour tous les nombres entiers
 - c) La fonction est définie pour tous les nombres rationnels
 - d) La fonction est définie pour tous les nombres réels sauf un
4. Quel est l'exemple d'une fonction qui n'est pas définie ?
 - a) $f(x) = 1 / x$
 - b) $f(x) = x^2 + 3$
 - c) $f(x) = 5$
 - d) $f(x) = x + 2$
5. Quelle est la propriété d'une fonction qui garantit qu'elle est injective ?
 - a) La fonction prend deux entrées différentes et produit deux sorties différentes
 - b) La fonction prend deux entrées différentes et produit la même sortie
 - c) La fonction prend deux entrées différentes et produit deux sorties différentes sauf une
 - d) La fonction prend deux entrées différentes et produit la même sortie sauf une

Une fonction est définie par une fonction

1. Quelle est la définition d'une fonction en mathématiques ?
 - a) Une relation entre deux variables
 - b) Une règle qui prend une entrée et produit une sortie
 - c) Un ensemble de nombres
 - d) Une équation qui relie deux variables
2. Quel est l'exemple d'une fonction simple ?

- a) $f(x) = 2x + 1$
 - b) $f(x) = x^2 + 3$
 - c) $f(x) = 5$
 - d) $f(x) = x + 2$
3. Quelle est la propriété d'une fonction qui garantit qu'elle est bien définie ?
- a) La fonction est définie pour tous les nombres réels
 - b) La fonction est définie pour tous les nombres entiers
 - c) La fonction est définie pour tous les nombres rationnels
 - d) La fonction est définie pour tous les nombres réels sauf un
4. Quel est l'exemple d'une fonction qui n'est pas définie ?
- a) $f(x) = 1 / x$
 - b) $f(x) = x^2 + 3$
 - c) $f(x) = 5$
 - d) $f(x) = x + 2$
5. Quelle est la propriété d'une fonction qui garantit qu'elle est injective ?
- a) La fonction prend deux entrées différentes et produit deux sorties différentes
 - b) La fonction prend deux entrées différentes et produit la même sortie
 - c) La fonction prend deux entrées différentes et produit deux sorties différentes sauf une
 - d) La fonction prend deux entrées différentes et produit la même sortie sauf une

1.4 Passage d'arguments

1. Quel est le but d'une introduction dans un discours ou un texte argumentatif ?
- a) Présenter les faits
 - b) Définir le but du texte
 - c) Captiver l'audience
 - d) Résumer le contenu
2. Quelle est la fonction d'un contre-argument dans un texte argumentatif ?
- a) Répondre aux objections
 - b) Présenter les faits
 - c) Définir le but du texte
 - d) Résumer le contenu
3. Quel est le but d'un exemple dans un texte argumentatif ?
- a) Illustrer un point
 - b) Présenter les faits
 - c) Définir le but du texte
 - d) Résumer le contenu

4. Quelle est la fonction d'un appel à l'action dans un texte argumentatif ?

- a) Résumer le contenu
- b) Définir le but du texte
- c) Présenter les faits
- d) Inviter le lecteur à prendre une décision

5. Quel est le but d'un résumé dans un texte argumentatif ?

- a) Résumer le contenu
- b) Définir le but du texte
- c) Présenter les faits
- d) Illustrer un point

Une particularité des fonctions en Python est que vous n'êtes pas obligé de préciser le type des arguments que vous lui passez

1. Quelle est la raison pour laquelle vous n'êtes pas obligé de préciser le type des arguments dans les fonctions en Python ?

- a) Parce que Python est un langage dynamique
- b) Parce que les fonctions en Python sont définies à l'exécution
- c) Parce que Python est un langage statique
- d) Parce que les fonctions en Python ne nécessitent pas de types de données

2. Quel est l'avantage de ne pas préciser le type des arguments dans les fonctions en Python ?

- a) Il permet d'écrire du code plus rapide
- b) Il permet d'écrire du code plus flexible
- c) Il permet d'écrire du code plus sécurisé
- d) Il permet d'écrire du code plus efficace

3. Quel est l'inconvénient de ne pas préciser le type des arguments dans les fonctions en Python ?

- a) Il peut entraîner des erreurs de type de données
- b) Il peut entraîner des erreurs de syntaxe
- c) Il peut entraîner des erreurs de sécurité
- d) Il peut entraîner des erreurs de performance

4. Quelle est la différence entre les fonctions en Python qui précisent le type des arguments et celles qui ne le font pas ?

- a) Les fonctions qui précisent le type des arguments sont plus rapides
- b) Les fonctions qui ne précisent pas le type des arguments sont plus flexibles
- c) Les fonctions qui précisent le type des arguments sont plus sécurisées
- d) Les fonctions qui ne précisent pas le type des arguments sont plus efficaces

5. Pourquoi est-il important de préciser le type des arguments dans les fonctions en Python lorsque cela est nécessaire ?

- a) Pour éviter les erreurs de type de données
- b) Pour améliorer la sécurité du code
- c) Pour accélérer l'exécution du code
- d) Pour réduire la complexité du code

Une particularité des fonctions en Python est que vous n'êtes pas obligé de préciser le type des arguments que vous lui passez

1. Quelle est la raison pour laquelle vous n'êtes pas obligé de préciser le type des arguments dans les fonctions en Python ?

- a) Parce que Python est un langage dynamique
- b) Parce que les fonctions en Python sont définies à l'exécution
- c) Parce que Python est un langage statique
- d) Parce que les fonctions en Python ne nécessitent pas de types de données

2. Quel est l'avantage de ne pas préciser le type des arguments dans les fonctions en Python ?

- a) Il permet d'écrire du code plus rapide
- b) Il permet d'écrire du code plus flexible
- c) Il permet d'écrire du code plus sécurisé
- d) Il permet d'écrire du code plus efficace

3. Quel est l'inconvénient de ne pas préciser le type des arguments dans les fonctions en Python ?

- a) Il peut entraîner des erreurs de type de données
- b) Il peut entraîner des erreurs de syntaxe
- c) Il peut entraîner des erreurs de sécurité
- d) Il peut entraîner des erreurs de performance

4. Quelle est la différence entre les fonctions en Python qui précisent le type des arguments et celles qui ne le font pas ?

- a) Les fonctions qui précisent le type des arguments sont plus rapides
- b) Les fonctions qui ne précisent pas le type des arguments sont plus flexibles
- c) Les fonctions qui précisent le type des arguments sont plus sécurisées
- d) Les fonctions qui ne précisent pas le type des arguments sont plus efficaces

5. Pourquoi est-il important de préciser le type des arguments dans les fonctions en Python lorsque cela est nécessaire ?

- a) Pour éviter les erreurs de type de données
- b) Pour améliorer la sécurité du code
- c) Pour accélérer l'exécution du code
- d) Pour réduire la complexité du code

1.5 Exercices

1. Quel est le nombre de kilogrammes dans 1,5 tonnes ?

- a) 1500 kg
- b) 1800 kg
- c) 2000 kg
- d) 2500 kg

2. 1,5 km équivaut à combien de mètres ?

- a) 1500 m
- b) 15000 m
- c) 150000 m
- d) 1500000 m

3. Quel est le résultat de $2 + 1,5$?

- a) 3,5
- b) 4
- c) 4,5
- d) 5

4. Quel est le nombre de litres dans 1,5 hectolitres ?

- a) 1500 L
- b) 15000 L
- c) 150000 L
- d) 1500000 L

5. Quel est le résultat de $3 \times 0,5$?

- a) 1,5
- b) 1,8
- c) 2
- d) 2,5

Exercice 1 :

1. Quel est le plus grand fleuve d'Europe?

- a) Le Danube
- b) Le Rhin
- c) La Volga
- d) Le Nil

2. Quel est le plus grand lac d'Europe?

- a) Le lac Ladoga
- b) Le lac Onega
- c) Le lac Baïkal
- d) Le lac Saimaa

3. Quel est le plus grand pays d'Europe?

- a) La Russie
- b) L'Allemagne
- c) La France
- d) L'Espagne

4. Quel est le plus grand fleuve d'Asie?

- a) Le fleuve Jaune
- b) Le fleuve Amour
- c) Le fleuve Ob
- d) Le fleuve Yangtsé

5. Quel est le plus grand lac d'Asie?

- a) Le lac Baïkal
- b) Le lac Sihon
- c) Le lac Balkhash
- d) Le lac Issyk-Kul

Ecrire une fonction itérative qui prend comme paramètre un entier naturel n et calcule le terme un de la suite en utilisant une boucle for.

1. Quelle est la fonction itérative qui calcule le terme n de la suite en utilisant une boucle for ?

- a) `def terme_n(n): return n * (n + 1) / 2`
- b) `def terme_n(n): for i in range(n): return i`
- c) `def terme_n(n): somme = 0; for i in range(n): somme += i; return somme`
- d) `def terme_n(n): return n ** 2`

2. Quelle est la variable qui stocke la somme des termes de la suite dans la fonction itérative ?

- a) somme
- b) n
- c) i
- d) resultat

3. Quelle est la boucle utilisée dans la fonction itérative pour calculer le terme n de la suite ?

- a) boucle while
- b) boucle for
- c) boucle while avec une condition
- d) boucle for avec une condition

4. Quelle est la fonction itérative qui calcule le terme n de la suite en utilisant une boucle for et qui est correcte ?

- a) `def terme_n(n): for i in range(n): return i`
- b) `def terme_n(n): somme = 0; for i in range(n): somme += i; return somme`

- c) `def terme_n(n): return n * (n + 1) / 2`
- d) `def terme_n(n): return n ** 2`

5. Quelle est la variable qui représente le terme actuel de la suite dans la fonction itérative ?

- a) somme
- b) n
- c) i
- d) resultat

Ecrire une fonction récursive qui prend comme paramètre un entier naturel n et calcule le terme un de la suite.

1. Quelle est la fonction récursive qui calcule le terme n de la suite 1, 1, 2, 3, 5, 8, ...?

- a) `def fib(n): return n if n <= 1 else fib(n-1) + fib(n-2)`
- b) `def fib(n): return n if n <= 1 else fib(n-1) + 1`
- c) `def fib(n): return n if n <= 1 else fib(n-1) * fib(n-2)`
- d) `def fib(n): return n if n <= 1 else fib(n-1) - fib(n-2)`

2. Quelle est la fonction récursive qui calcule le terme n de la suite 0, 1, 1, 2, 3, 5, 8, ...?

- a) `def fib(n): return n if n <= 1 else fib(n-1) + fib(n-2)`
- b) `def fib(n): return n if n <= 1 else fib(n-1) + 1`
- c) `def fib(n): return n if n <= 1 else fib(n-1) * fib(n-2)`
- d) `def fib(n): return n if n <= 1 else fib(n-1) - fib(n-2)`

3. Quelle est la fonction récursive qui calcule le terme n de la suite 1, 1, 2, 4, 8, 16, ...?

- a) `def fib(n): return n if n <= 1 else fib(n-1) + fib(n-2)`
- b) `def fib(n): return n if n <= 1 else fib(n-1) + 1`
- c) `def fib(n): return n if n <= 1 else fib(n-1) * fib(n-2)`
- d) `def fib(n): return n if n <= 1 else fib(n-1) + 2 * fib(n-2)`

4. Quelle est la fonction récursive qui calcule le terme n de la suite 1, 2, 4, 8, 16, 32, ...?

- a) `def fib(n): return n if n <= 1 else fib(n-1) + fib(n-2)`
- b) `def fib(n): return n if n <= 1 else fib(n-1) + 1`
- c) `def fib(n): return n if n <= 1 else fib(n-1) * fib(n-2)`
- d) `def fib(n): return n if n <= 1 else fib(n-1) + 2 * fib(n-2)`

5. Quelle est la fonction récursive qui calcule le terme n de la suite 0, 1, 1, 2, 3, 5, 8, 13, ...?

- a) `def fib(n): return n if n <= 1 else fib(n-1) + fib(n-2)`
- b) `def fib(n): return n if n <= 1 else fib(n-1) + 1`
- c) `def fib(n): return n if n <= 1 else fib(n-1) * fib(n-2)`
- d) `def fib(n): return n if n <= 1 else fib(n-1) - fib(n-2)`

Exercice 2 :

1. Quel est le résultat de l'opération 7×3 ?

- a) 14
- b) 21
- c) 28
- d) 35

2. Quel est le résultat de l'opération $9 - 2$?

- a) 5
- b) 7
- c) 11
- d) 13

3. Quel est le résultat de l'opération 4×6 ?

- a) 20
- b) 22
- c) 24
- d) 26

4. Quel est le résultat de l'opération $11 + 1$?

- a) 10
- b) 12
- c) 14
- d) 16

5. Quel est le résultat de l'opération $8 - 4$?

- a) 2
- b) 4
- c) 6
- d) 8

Soit t un tableau d'entiers de taille n .

1. Quel est le temps de complexité de l'algorithme de tri par insertion ?

- a) $O(n^2)$
- b) $O(n \log n)$
- c) $O(n)$
- d) $O(\log n)$

2. Quel est le temps de complexité de l'algorithme de recherche dichotomique ?

- a) $O(n)$
- b) $O(n \log n)$

- c) $O(\log n)$
 - d) $O(n^2)$
3. Quel est le temps de complexité de l'algorithme de tri par sélection ?
- a) $O(n \log n)$
 - b) $O(n^2)$
 - c) $O(n)$
 - d) $O(\log n)$
4. Quel est le temps de complexité de l'algorithme de recherche linéaire ?
- a) $O(n \log n)$
 - b) $O(n^2)$
 - c) $O(n)$
 - d) $O(\log n)$
5. Quel est le temps de complexité de l'algorithme de tri par fusion ?
- a) $O(n \log n)$
 - b) $O(n^2)$
 - c) $O(n)$
 - d) $O(\log n)$

Ecrire une fonction sansDoublons(t) qui retourne True si le tableau d'entiers t est sans doublons (c'est à dire sans apparition multiple d'un élément), False sinon.

1. La fonction sansDoublons(t) retourne True si le tableau d'entiers t est sans doublons, False sinon. Quel est le résultat de sansDoublons([1, 2, 3, 4, 5]) ?
- a) True
 - b) False
 - c) Il n'est pas possible de savoir sans exécuter la fonction
 - d) La fonction ne retourne jamais de résultat
2. La fonction sansDoublons(t) retourne True si le tableau d'entiers t est sans doublons, False sinon. Quel est le résultat de sansDoublons([1, 2, 2, 4, 5]) ?
- a) True
 - b) False
 - c) Il n'est pas possible de savoir sans exécuter la fonction
 - d) La fonction ne retourne jamais de résultat
3. La fonction sansDoublons(t) retourne True si le tableau d'entiers t est sans doublons, False sinon. Quel est le résultat de sansDoublons([]) ?
- a) True
 - b) False
 - c) Il n'est pas possible de savoir sans exécuter la fonction

d) La fonction ne retourne jamais de résultat

4. La fonction sansDoublons(t) retourne True si le tableau d'entiers t est sans doublons, False sinon. Quel est le résultat de sansDoublons([1, 1, 1, 1, 1]) ?

a) True

b) False

c) Il n'est pas possible de savoir sans exécuter la fonction

d) La fonction ne retourne jamais de résultat

5. La fonction sansDoublons(t) retourne True si le tableau d'entiers t est sans doublons, False sinon. Quel est le résultat de sansDoublons([1, 2, 3, 4, 5, 6]) ?

a) True

b) False

c) Il n'est pas possible de savoir sans exécuter la fonction

d) La fonction ne retourne jamais de résultat

Answer Key

Python – Les fonctions

- 1: None
- 2: Le mot-clé ``def`` est utilisé pour les fonctions nommées, tandis que ``lambda`` est utilisé pour les fonctions anonymes.
- 3: En utilisant le mot-clé ``return``
- 4: Ils permettent de modifier le comportement d'une fonction sans la modifier
- 5: En utilisant le mot-clé ``nonlocal``

Sommaire

- 1: Louis-Napoléon Bonaparte
- 2: Déclaration des droits de l'homme et du citoyen
- 3: Louis XVI
- 4: Bataille de Valmy
- 5: Georges Danton

Python – Les fonctions

- 1: None
- 2: Le mot-clé ``def`` est utilisé pour les fonctions nommées, tandis que ``lambda`` est utilisé pour les fonctions anonymes.
- 3: En utilisant le mot-clé ``return``
- 4: Ils permettent de modifier le comportement d'une fonction sans la modifier
- 5: En utilisant le mot-clé ``nonlocal``

Fonctions

- 1: Appliquer une fonction à chaque élément d'un tableau
- 2: Récupérer les éléments d'un tableau qui répondent à une condition
- 3: Combinez tous les éléments d'un tableau en un seul élément
- 4: Trier les éléments d'un tableau dans l'ordre croissant ou décroissant
- 5: Faire une boucle sur un tableau

Fonctions mathématiques (module math)

- 1: `math.sin(x)`
- 2: `math.log(x)`
- 3: `math.sqrt(x)`
- 4: `math.pi`
- 5: `math.e`

Fonctions trigonométriques

- 1: $1/2$
- 2: $1/\sqrt{2}$
- 3: $\sqrt{3}$
- 4: 1
- 5: 1

1. Fonctions

- 1: Renvoie le type d'un objet
- 2: La fonction ``let`` est utilisée pour déclarer des variables, tandis que la fonction ``const`` est utilisée pour déclarer des constantes
- 3: Renvoie un nouveau tableau avec les résultats d'une fonction appliquée à chaque élément d'un tableau
- 4: La fonction ``forEach()`` applique une fonction à chaque élément d'un tableau, tandis que la fonction ``map()`` renvoie un nouveau tableau avec les résultats
- 5: Renvoie la somme des éléments d'un tableau

1.1 Principe et généralités

- 1: Force de gravité
- 2: Principe d'entropie
- 3: Mouvement uniforme
- 4: Force de pression
- 5: Principe de conservation de l'énergie

En programmation, les fonctions sont très utiles pour réaliser plusieurs fois la même opération au sein d'un programme.

- 1: Elles permettent de réaliser plusieurs fois la même opération
- 2: Un nom donné à un bloc d'instructions qui peut être réutilisé
- 3: Elles permettent de réduire la taille du code et de gagner du temps
- 4: Une fonction qui calcule la somme de deux nombres
- 5: L'appel d'une fonction à plusieurs reprises

1.2 Fonctions mathématiques (module math)

- 1: `math.sin(x)`
- 2: `math.log(x)`
- 3: `math.sqrt(x)`
- 4: `math.pi`
- 5: `math.e`

faire import math

- 1: 63
- 2: 6
- 3: 14
- 4: 7
- 5: 48

`math.floor(-7.6)` : partie entière, donne ici -8.0.

- 1: -8.0
- 2: -4.0
- 3: 2.0
- 4: -2.0
- 5: 0.0

`int(math.floor(4.5))` : pour avoir l'entier 4.

- 1: 4
- 2: 3
- 3: 2
- 4: 1
- 5: 0

1.3 Définition d'une fonction

- 1: Permet de sortir de la fonction et renvoyer une valeur
- 2: Utiliser la syntaxe ``def fonction(argument) :``
- 3: Tous les avantages ci-dessus
- 4: Utiliser la syntaxe ``fonction()``
- 5: Tous les types de variables

Une fonction est définie par une fonction

- 1: Une règle qui prend une entrée et produit une sortie
- 2: $f(x) = 5$
- 3: La fonction est définie pour tous les nombres réels
- 4: $f(x) = 1 / x$
- 5: La fonction prend deux entrées différentes et produit deux sorties différentes
- 1: Une règle qui prend une entrée et produit une sortie
- 2: $f(x) = 5$
- 3: La fonction est définie pour tous les nombres réels
- 4: $f(x) = 1 / x$
- 5: La fonction prend deux entrées différentes et produit deux sorties différentes

1.4 Passage d'arguments

- 1: Définir le but du texte
- 2: Répondre aux objections
- 3: Illustrer un point
- 4: Inviter le lecteur à prendre une décision
- 5: Résumer le contenu

Une particularité des fonctions en Python est que vous n'êtes pas obligé de préciser le type des arguments que vous lui passez

- 1: Parce que Python est un langage dynamique
- 2: Il permet d'écrire du code plus flexible
- 3: Il peut entraîner des erreurs de type de données
- 4: Les fonctions qui ne précisent pas le type des arguments sont plus flexibles
- 5: Pour éviter les erreurs de type de données
- 1: Parce que Python est un langage dynamique
- 2: Il permet d'écrire du code plus flexible
- 3: Il peut entraîner des erreurs de type de données
- 4: Les fonctions qui ne précisent pas le type des arguments sont plus flexibles
- 5: Pour éviter les erreurs de type de données

1.5 Exercices

- 1: 1500 kg
- 2: 1500 m
- 3: 3,5

4: 1500 L
5: 1,5

Exercice 1 :

1: Le Danube
2: Le lac Ladoga
3: La Russie
4: Le fleuve Amour
5: Le lac Baïkal

Ecrire une fonction itérative qui prend comme paramètre un entier naturel n et calcule le terme un de la suite en utilisant une boucle for.

```
1: def terme_n(n): somme = 0; for i in range(n): somme += i; return somme
2: somme
3: boucle for
4: def terme_n(n): somme = 0; for i in range(n): somme += i; return somme
5: i
```

Ecrire une fonction récursive qui prend comme paramètre un entier naturel n et calcule le terme un de la suite.

```
1: def fib(n): return n if n <= 1 else fib(n-1) + fib(n-2)
2: def fib(n): return n if n <= 1 else fib(n-1) + fib(n-2)
3: def fib(n): return n if n <= 1 else fib(n-1) + 2 * fib(n-2)
4: def fib(n): return n if n <= 1 else fib(n-1) + 2 * fib(n-2)
5: def fib(n): return n if n <= 1 else fib(n-1) + fib(n-2)
```

Exercice 2 :

1: 21
2: 7
3: 24
4: 12
5: 4

Soit t un tableau d'entiers de taille n .

1: $O(n)$
2: $O(\log n)$
3: $O(n^2)$
4: $O(n)$
5: $O(n \log n)$

Ecrire une fonction sansDoublons(t) qui retourne True si le tableau d'entiers t est sans doublons (c'est à dire sans apparition multiple d'un élément), False sinon.

1: True

2: False
3: True
4: False
5: True