

1 Exercices de connaissance du cours

1.1 Utilisation de fonctions prédéfinies

1. Étant données deux variables `chaine1` et `chaine2` de type `str` et une variable `x` de type entier¹, donner des expressions qui représentent :
 - a. le nombre de caractères contenus dans `chaine1`
 - b. le nombre de caractères contenus dans la chaîne la plus longue des deux
 - c. la valeur absolue de `x`
 - d. un caractère pris au hasard dans la chaîne `chaine1`, en supposant que `chaine1` est non vide

1.2 Vocabulaire, instructions et expressions

2. Compléter avec les suites de mots : instruction, jour, valeur, affectation, expression, variables, mémoire, affecte, gauche, droite, identificateur, évalué, l'association (variable, valeur), variable.
 - a. pour exécuter une affectation, on l'..... en partie du = puis on la valeur obtenue à la en partie La mémoire du programme est mise à avec
 - b. une s'évalue à une, les valeurs des qu'elle contient sont prises dans la du programme.
 - c. une s'exécute d. le nom d'une variable est appelé e. une est une instruction.

1.3 Erreurs, quotidien du programmeur...

3. Pour chaque suite d'instructions, décrivez dans un tableau l'état de la mémoire et donner la valeur de chaque variable à la fin de l'exécution. En cas d'erreur, dire si c'est une erreur de syntaxe, une erreur de type, une autre erreur (laquelle ?). Pour les erreurs de syntaxe : indiquez quelle règle syntaxique Python n'est pas respectée.

- | | | | | |
|------------------------|--|------------------------|-----------------------------|-----------------------|
| a. | c. | d. | e. | g. |
| <code>i = 0</code> | <code>total = 4 * '4' + '2' * 2</code> | <code>x = 1</code> | <code>x = 1</code> | <code>x = ---1</code> |
| <code>i + 1 = i</code> | <code>total = total - '1'</code> | <code>y = x - 1</code> | <code>y := x+-1</code> | <code>x ** 2</code> |
| b. | <code>total = total + '10'</code> | <code>z = 1 / y</code> | f. | |
| <code>x = 2</code> | | | <code>un = 'un'</code> | |
| <code>y = y + x</code> | | | <code>'deux' = 1 + 1</code> | |

2 Variables et affectations

4. Décrivez dans un tableau l'état de la mémoire du programme durant l'exécution des instructions Python suivantes, puis indiquer pour chaque variable quelle est sa valeur à la fin de l'exécution.

- | | | | |
|------------------------|------------------------|-----------------------------|---|
| a. | b. | c. | d. |
| <code>a = 10</code> | <code>a = 6</code> | <code>x = 2</code> | <code>operation = "3+6"</code> |
| <code>b = a</code> | <code>b = 7</code> | <code>carre_x = x**2</code> | <code>question = 'Combien font :'</code> |
| <code>a = 12</code> | <code>c = a</code> | <code>x = 3</code> | <code>espace = ' '</code> |
| <code>c = b + 1</code> | <code>c = c + b</code> | | <code>invite = question + espace + operation</code> |
| | | | <code>invite = invite + espace + '?' * 2</code> |

3 Échange de valeurs

La variable `un` est initialisée avec la valeur 1. La variable `deux` est initialisée avec la valeur 2. On veut échanger la valeur des variables `un` et `deux`, c'est à dire qu'à la fin de l'exécution la valeur de `un` est 2 et la valeur de `deux` est 1.

5. Ces instructions réalisent-elles l'échange ? Justifier.

```
un = deux
deux = un
```

6. Modifier les instructions pour réaliser l'échange (la méthode doit fonctionner quelles que soient les valeurs initiales).

¹abus de langage : la bonne formulation est "variable associée à des valeurs de type `str`".

4 Appels de fonctions

Dans cet exercice la docstring complète n'est pas donnée car non pertinente pour la compétence travaillée.

7. Décrivez le résultat des expressions suivantes ainsi que l'état de la mémoire durant leur évaluation :

```
def f(x:int, y:int)->int:          def g(x:int, y:int)->int:          DELTA = 1
    return x + y + 1                res = x                            def h(x:int, y:int)->int:
                                    res = res + y                          res = x + y
                                    return res + 1                          return res + DELTA
```

a. `f(2, 3+4)`

b. `g(2, 3+4)`

c. `h(DELTA+1, 3+4)`

8. Que pensez-vous du code suivant ?

```
x = 3
def f(x : int, y : int) -> int:
    return x + y + 1
def g(z : int) -> int:
    x = z**2
    return x + 12
```

5 Écriture de fonctions

Le but de ces exercices est d'apprendre à décomposer un problème en sous-problèmes et d'identifier quelles fonctions intermédiaires définir.

5.1 Racines

Nous désirons définir des fonctions qui renvoient les racines d'un polynôme du second degré $ax^2 + bx + c$.

$$r_1 = \frac{-b + \sqrt{\Delta}}{2a}$$

$$r_2 = \frac{-b - \sqrt{\Delta}}{2a}$$

avec: $\Delta = b^2 - 4ac$, pourvu que $\Delta \geq 0$

9. Quelle est la signature d'une fonction `racine1` qui calcule la valeur de r_1 ?

10. Quelle fonction intermédiaire utiliser au vu de l'énoncé ?

11. Écrire la fonction intermédiaire et les fonctions `racine1` et `racine2`.

5.2 Masque

Le but de l'exercice est de traiter un problème complexe en le décomposant et en le codant avec des fonctions. Le problème est : on dispose d'un entier positif à 4 chiffres (compris entre 1000 et 9999 inclus), par ex 1234. On veut calculer une chaîne de caractères de taille 4 qui masque les chiffres des centaines et des dizaines de cet entier. Par exemple le masquage de 1234 produira la chaîne de caractères '1**4'².

12. Suggérer une décomposition de ce problème en sous-problème pouvant être codés par des fonctions.

13. Écrire une fonction `masque` qui prend en paramètre un entier comme décrit dans l'énoncé et renvoie une chaîne de caractères correspondant à son masquage, ainsi que les fonctions intermédiaires prévues précédemment. Par exemple `masque(1234)` vaut '1**4'.

²On trouve cette fonctionnalité, en plus élaborée, pour masquer partiellement les numéros de carte bancaire ou de téléphone.