

Questions sur "Python Algo" ?

Si vous avez des questions/remarques sur les questions, merci d'ajouter un commentaire (Insertion / commentaire ou Ctrl+Alt+M).

Question : À la page 114, dans l'exercice, pourquoi dans le premier programme `x2 += 1` ne modifie pas `x1` ? `x2` n'est-il pas un alias du même objet que `x1` ? ?

Réponse : C'est car les entiers ne sont pas modifiables. Ainsi `x2 += 1` est compris comme `x2 = X2 + 1`.

On calcule donc `x2 + 1` qui donne un objet 422. On n'a donc pas touché à 421 qui est associé à `x1`. On associe cet objet 422 à `x2`. On a donc `x2` qui vaut 422 et `x1` qui vaut 421. C'est ce à quoi on s'attend naturellement.

Question : À la page 120, qu'est-ce qu'un code ?

Réponse : chaque caractère est représenté par un entier. Le code est cet entier. `ord` permet de passer du caractère à l'entier et `chr` de l'entier au code. L'entier dépend de l'encodage choisi : ASCII, latin1, utf-8, etc.

Question : Pour `chiffre = ord(c) - ord('0')` ainsi que pour `c = chr(ord('0') + chiffre)`, pourquoi y a-t-il besoin de passer par une opération par 0 ? ?

Réponse : On s'appuie sur le fait que les chiffres (de '0' à '9') sont consécutifs. Leurs codes se suivent donc. Si `c` est un caractère entre '0' et '9', on `ord(c) - ord('0')` qui sera donc l'entier qui correspond à ce chiffre. Par exemple, si `c == '2'`, `ord(c) = 50`, et `ord(c) - ord('0')` vaut 2. En Python, on utilise plutôt `int(c)` qui est plus général et marchera même si `c` est une chaîne qui est une suite de chiffres.

Question : Concernant les raffinages pour le jeu du devin : quand on a créé différentes fonctions faut-il imbriquer leurs raffinages entre elles ou peut-on procéder par fonction pour plus de lisibilité ?

Réponse : Il s'agit de faire un raffinement par fonction : chaque fonction est considérée comme un programme différent. Pour les exercices 3 et 4, on peut faire référence aux actions complexes des exercices 1 et 2 qu'il n'y a pas lieu de décomposer à nouveau puisque ceci a déjà été fait.

Question : pour l'exercice 1.1 du jeu du devin, faut-il prévoir une vérification des propositions de l'utilisateur (entier compris entre 1 et 999) ?

Réponse : On peut répondre trop grand ou trop petit...

Question : Où sont les programmes auxquels fait référence le cours sur les séquences (diapo 107, 110 s'il vous plaît ?

Réponse : tp-seq-cours (j'avais oublié de les mettre en ligne).

Question : Pour l'exercice du jeu du devin, l'énoncé du 2.4 (même s'il n'est pas à faire pour le 5/11) dit qu'il ne devrait y avoir qu'une seule ligne à changer si on souhaite modifier les valeurs limites du jeu. Cela laisse entendre que ces valeurs doivent être passées comme arguments aux programmes correspondants aux 1.1 et 1.2. Comment décrire les arguments reçus par une fonction dans ses raffinages ?

Réponse : Pas besoin de paramètre ici. Utiliser une variable locale (en fait une constante) est suffisant ici.

Question : Pour l'exercice du jeu du devin, pour les parties 1.3 et 1.4, faut-il utiliser les programmes des parties 1.1 et 1.2 comme des modules à importer dans le programme du jeu complet ?

Réponse : Pour la partie raffinement, il suffit d'utiliser l'action complexe définie pour le R0 des questions 1 & 2. Dans le code, une fonction est définie pour chaque R0. Si elles sont toutes dans le même fichier, on peut les utiliser directement. Si elles sont dans des fichiers différents, il faut effectivement utiliser import.

Question : ... ?

Réponse : ...

Question : ... ?

Réponse : ...

Question : ... ?

Réponse : ...

Question : ... ?

Réponse : ...