



CONSEIL ET EXPERTISE TECHNIQUE
Infrastructure • Cybersecurity • Cloud • Data

Python 03

Matthieu DESTOMBES
Mail : matthieu.destombes@ynov.com

08 Mars 2021
Toulouse

Travaux

Pratiques

- Création d'un programme complet en 2 fichiers
 - Gestion des variables internes au programme
 - Gestion de l'interface utilisateur
-
- Création d'un jeu de devinette
-
- Utiliser les commentaires
 - Utiliser les doc-chaînes

Gestion des variables internes

Objectifs

- Gérer 3 variables internes au programme
 - Un nombre généré de manière aléatoire entre deux bornes définies
 - Valeur par défaut à 'None'
 - La borne supérieur
 - Valeur par défaut à '1000'
 - La borne inférieur
 - Valeur par défaut à '0'
- Utilisation de 6 fonctions permettant
 - La génération, puis la récupération du nombre aléatoire en mémoire
 - L'affectation, puis la récupération de la borne supérieur en mémoire
 - L'affectation, puis la récupération de la borne inférieur en mémoire
- Un programme de test dans le fichier qui permet de vérifier que les fonctions fonctionnent correctement
 - Affichage des valeurs par défaut en mémoire (Limites et Nombre généré)
 - Génération et affichage du nombre avec limites par défaut
 - Modification des limites supérieur (150) et inférieur (100)
 - Affichage des nouvelles valeurs (Limites uniquement)
 - Génération et affichage du nombre avec limites modifiées

Attendu

- Lors de l'exécution du fichier, vous devez obtenir le rendu suivant

```
/=====
| Original internal number is: 'None'
| Original internal upper limit is: '1000'
| Original internal lower limit is: '0'
|=====
| Generated internal number with default limit is: '264'
\=====

/=====
| Modified internal upper limit is: '150'
| Modified internal lower limit is: '100'
|=====
| Generated internal number with new limit is: '143'
\=====

Process finished with exit code 0
```

Gestion des variables internes

Contraintes (Part I)

- Fichier python nommé « randomize_management.py »
- Les 3 variables globales doivent être nommées
 - « internal_number »
 - « internal_upper_limit »
 - « internal_lower_limit »
- Les 6 fonctions doivent être nommées
 - « set_internal_number »
 - « get_internal_number »
 - « set_internal_upper_limit »
 - « get_internal_upper_limit »
 - « set_internal_lower_limit »
 - « get_internal_lower_limit »
- Aucun « print » dans les fonctions

Contraintes (Part II)

- Fonction « main » avec le programme de test
 - Ne doit pas s'exécuter si le fichier est importé
 - Toutes les affectations, modifications des variables internes doivent utiliser les fonctions définies à cet effet
 - Toutes récupérations des valeurs contenues dans les variables internes doivent utiliser les fonctions définies à cet effet
- Toute la partie mise en forme faite avec des « print » doit-être contenue dans le programme de test

Gestion de l'interface utilisateur

Objectifs (Part I)

- Utiliser le fichier précédemment créé
- Définir 8 variables internes au programme
 - Message de la question principale
 - Message de la question pour limite inférieur
 - Message de la question pour limite supérieur
 - Message indiquant la bonne réponse
 - Message indiquant que la valeur est supérieur
 - Message indiquant que la valeur est inférieur
 - Message indiquant le succès et le nombre de tentative
 - Le nombre de tentative
- Utilisation de 5 fonctions permettant
 - L'initialisation de la partie
 - Le nom de la partie est donné en argument
 - La demande à l'utilisateur d'un nombre
 - La réponse de l'utilisateur doit être retournée par la fonction au format « String » (Chaîne de caractère)
 - La comparaison du nombre donné par l'utilisateur au nombre interne
 - Le nombre donné par l'utilisateur est en argument
 - La modification des limites supérieur et inférieur
 - Les nouvelles limites sont données en argument
 - Valeur par défaut 100 et 200

Félicitation de l'utilisateur et fin de partie

Objectifs (Part II)

- Un programme de test dans le fichier qui permet de vérifier que les fonctions fonctionnent correctement en lançant 5 parties
 - Process commun entre les parties
 - Affichage du nom de la partie
 - Affichage des limites
 - Demande du nombre à l'utilisateur jusqu'à obtention du bon nombre
 - Affichage du résultat de la comparaison
 - Affichage de la réussite
 - Affichage du nombre de tentative
 - Partie 1 avec
 - Valeurs par défaut
 - Partie 2 avec
 - Valeurs limites modifiées (Valeurs par défaut)
 - Partie 3 avec
 - Valeur limite inférieur modifiée en demandant à l'utilisateur
 - Valeur limite supérieur par défaut
 - Partie 4 avec
 - Valeur limite supérieur modifiée en demandant à l'utilisateur
 - Valeur limite inférieur par défaut
 - Partie 5 avec
 - Valeurs limites inférieur et supérieur modifiées en demandant à l'utilisateur

Gestion de l'interface utilisateur

Attendu (Partie 1)

```
/=====
| Game I
|=====
| Aim of this game is to find a randomized number as:
| - Lower or equal to '1000'
| - Upper or equal to '0'
|=====
| Give me a number: 500
| No. It's upper...
|=====
| Give me a number: 750
| No. It's lower...
|=====
| Give me a number: 658
| Yes. It is! You win!
| Success in '3' attempt.
\=====
```

Attendu (Partie 2)

```
/=====
| Game II
|=====
| Aim of this game is to find a randomized number as:
| - Lower or equal to '200'
| - Upper or equal to '100'
|=====
| Give me a number: 150
| No. It's lower...
|=====
| Give me a number: 125
| No. It's upper...
|=====
| Give me a number: 148
| Yes. It is! You win!
| Success in '3' attempt.
\=====
```

Gestion de l'interface utilisateur

Attendu (Partie 3)

```
/=====
| Give me the new lower limit: 150
\=====

/=====
| Game III
|=====
| Aim of this game is to find a randomized number as:
| - Lower or equal to '200'
| - Upper or equal to '150'
|=====
| Give me a number: 160
| No. It's lower...
|=====
| Give me a number: 158
| No. It's upper...
|=====
| Give me a number: 159
| Yes. It is! You win!
| Success in '3' attempt.
\=====
```

Attendu (Partie 4)

```
/=====
| Give me the new upper limit: 150
\=====

/=====
| Game IV
|=====
| Aim of this game is to find a randomized number as:
| - Lower or equal to '150'
| - Upper or equal to '100'
|=====
| Give me a number: 145
| No. It's upper...
|=====
| Give me a number: 148
| No. It's lower...
|=====
| Give me a number: 147
| Yes. It is! You win!
| Success in '3' attempt.
\=====
```

Gestion de l'interface utilisateur

Attendu (Partie 5)

```
/=====
| Give me the new lower limit: 10
| Give me the new upper limit: 20
\=====

/=====
| Game V
|=====
| Aim of this game is to find a randomized number as:
| - Lower or equal to '20'
| - Upper or equal to '10'
|=====
| Give me a number: 15
| No. It's lower...
|=====
| Give me a number: 12
| No. It's upper...
|=====
| Give me a number: 14
| Yes. It is! You win!
| Success in '3' attempt.
\=====
```


Gestion de l'interface utilisateur

Contraintes (Part I)

- Fichier python nommé « finder_management.py »
- Les 8 variables globales doivent être nommées
 - « main_question »
 - « lower_question »
 - « upper_question »
 - « good_answer »
 - « count_display »
 - « bad_lower_answer »
 - « bad_upper_answer »
 - « try_number »
- Les 5 fonctions doivent être nommées
 - « start_the_finder »
 - « ask_the_number_to_user »
 - « check_the_number »
 - « limits_modify »
 - « congrats_user »
- La quasi-totalité de la mise en forme faite avec des « print » doit-être contenue dans les fonctions

Contraintes (Part II)

- Fonction « main » avec le programme de test
 - Ne doit pas s'exécuter si le fichier est importé
 - Toutes les affectations, modifications des variables internes doivent utiliser les fonctions définies à cet effet
- Seule la partie mise en forme faite avec des « print » pour la demande de modification des limites doit-être contenue dans le programme de test