

Programmer avec Python

Exercices

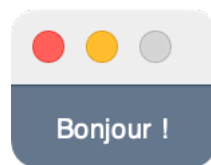
Séance n° 6

Les interfaces graphiques

Objectif : Réalisez chacun des exercices en utilisant le logiciel Spyder comme il a été expliqué en cours. Enregistrez bien votre travail au fur et à mesure !

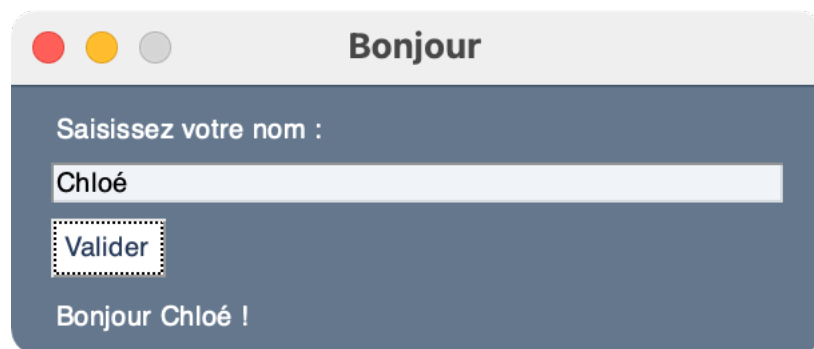
1 Bonjour !

Écrivez un programme affichant une fenêtre qui dit Bonjour ! à l'utilisateur.



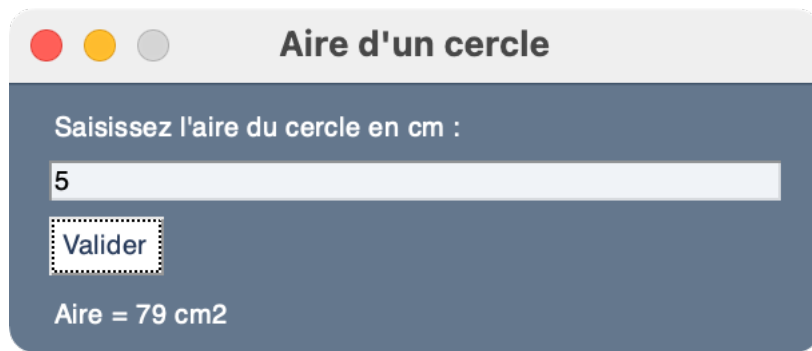
2 Bonjour vous !

Écrivez un programme affichant une fenêtre à l'utilisateur avec un champ lui permettant de saisir son prénom. Un bouton Valider permet de valider la saisie et le programme affiche alors "Bonjour XXX" (XXX étant le prénom) dans la même fenêtre.



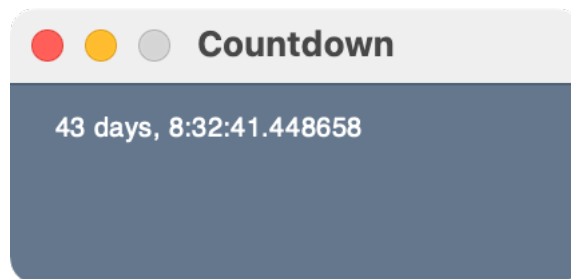
3 Aire d'un cercle

Lors de la séance 5, vous aviez écrit une fonction permettant de calculer l'aire d'un cercle grâce au module math. Reprenez cette fonction (vous avez la correction sur l'ENT si besoin). Programmez une interface graphique dans laquelle on saisira le rayon du cercle et qui affichera l'aire du cercle après validation. Vous utiliserez votre fonction pour faire le calcul.



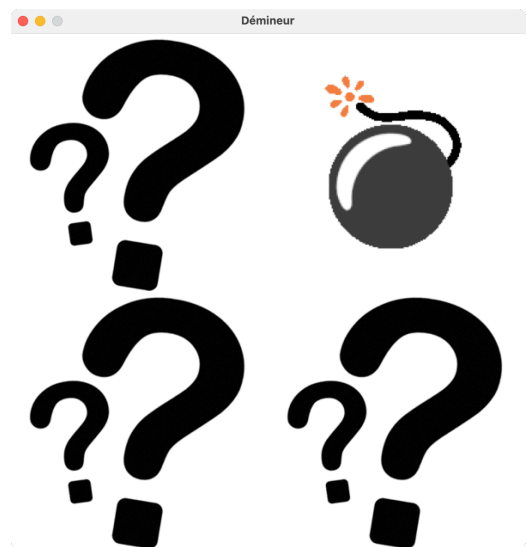
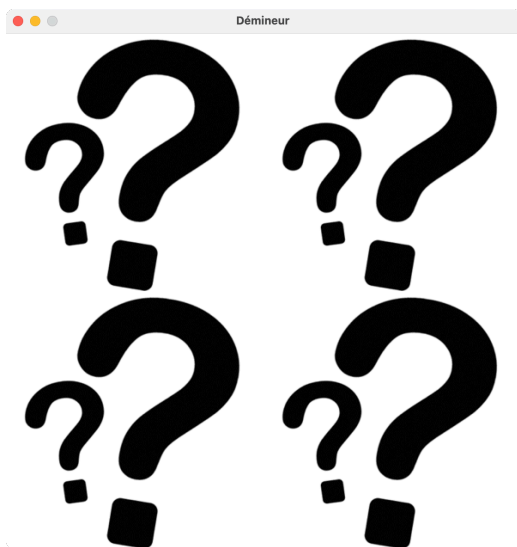
4 Le retour du compte à rebours

Reprenez l'exercice de la séance 5 où vous affichiez le temps restant jusqu'aux vacances de Noël. Ajoutez lui une interface graphique : le compte à rebours apparaît dans la fenêtre et le timer sera arrêté (avec l'instruction `t.cancel()`) lorsqu'on ferme la fenêtre. Il vous faudra modifier la fonction précédemment écrite pour que l'affichage se fasse dans la fenêtre et non plus dans la console.



5 Déminage

Développez un programme avec une interface graphique affichant quatre images avec des points d'interrogations. L'utilisateur peut cliquer sur une seule image. Lorsqu'il clique, l'image est dévoilée, si elle est gagnante c'est l'image gagnante qui apparaît à sa place, si l'image est perdante, on affiche alors une bombe.



Vous trouverez les images sur l'ENT dans les ressources du cours !

6 Tama*

Nous allons réaliser un jeu de tamagotchi. Le jeu affichera le petit chien Tama. Au début du jeu Tama possède un niveau de bonheur et un niveau de nourriture de 100%. Chaque seconde, son bonheur et sa nourriture baisse de 1. Le joueur peut nourrir et distraire Tama pour le maintenir en vie en appuyant sur des boutons. Si le bonheur de Tama ou sa nourriture atteint 30, Tama devient malheureux. Si un seul niveau, nourriture ou bonheur, atteint 0, Tama meurt.



Pour réaliser le jeu, vous allez développer une classe Tama qui représentera le petit chien et ses caractéristiques et un programme principal (dans un autre fichier) qui se chargera de construire et gérer l'interface graphique.

La classe Tama

1. Créez une classe Tama dans un fichier nommé `tama.py`
2. Faîtes en sorte qu'un Tama possède un niveau de bonheur et un niveau de nourriture, toujours initialisés à 100 à sa création
3. Programmez les méthodes de Tama :
 - Une méthode `afficher` qui affiche dans la console le niveau de nourriture et de bonheur de Tama
 - Une méthode `est_malheureux` qui renvoie vrai si les niveaux de nourriture ou bonheur sont inférieurs à 30, faux sinon
 - Une méthode `est_mort` qui renvoie vrai si les niveaux de nourriture ou bonheur sont nuls, faux sinon
 - Une méthode `nourrir` qui ajoute 5 points au niveau de nourriture. Attention le niveau de nourriture ne doit pas dépasser 100 et on ne peut plus nourrir Tama s'il est mort.
 - Une méthode `distraindre` qui ajoute 3 points au niveau de bonheur. Attention le niveau de bonheur ne doit pas dépasser 100 et on ne peut plus jouer avec Tama s'il est mort.
 - Une méthode `vieillir` qui diminue chaque niveau de 1. Attention, Tama ne peut pas non plus vieillir s'il est mort...

L'interface graphique

1. Travaillez dans un nouveau fichier nommé `play_tama.py` et importez la classe `Tama`
2. Développez la fenêtre du jeu en suivant l'exemple de la capture. Les images sont disponibles dans les ressources du cours sur l'ENT
3. Créez un nouvel objet `Tama` et une fonction `cycle`. La fonction `cycle` fait appel à la méthode `vieillir` de `Tama` puis modifie les niveaux de nourriture et bonheur dans l'interface. Si `Tama` devient malheureux ou meurt, la fonction met à jour l'image de `Tama`.
4. Mettez en place un objet `Repeat` pour répéter la fonction `cycle` toutes les 1 secondes. Testez bien.
5. Ajoutez les événements adéquats sur les boutons : un clic sur `Nourrir` fait appel à la méthode `nourrir` de `Tama`, même chose pour le bouton `jouer` avec la méthode `distraindre`.