

Les dictionnaires pour l'escape game

Ex 1

Exercice 1 : Créer des objets pour votre escape game

Créez trois dictionnaires représentant les objets suivants :

1.0.1 a) Un livre

- Nom : "livre"
- Position : x=150, y=180
- Dimensions : largeur=80, hauteur=60
- Propriété "lu" : False
- Description : "Un vieux grimoire poussiéreux"

```
1 livre = {  
2  
3  
4  
5  
6  
7  
8 }
```

1.0.2 b) Une lampe

- Nom : "lampe"
- Position : x=500, y=300
- Dimensions : largeur=40, hauteur=80
- Propriété "allumee" : False
- Description : "Une lampe à huile"

```
1 lampe = {  
2  
3  
4  
5  
6  
7  
8 }
```

```
9  
10 }
```

1.0.3 c) Un tiroir

- Nom : “tiroir”
- Position : x=200, y=400
- Dimensions : largeur=100, hauteur=50
- Propriété “ouvert” : False
- Description : “Un tiroir du bureau”

```
1 tiroir = {  
2  
3  
4  
5  
6  
7  
8  
9  
10 }
```

Ex 2

Exercice 2 : Modifier les propriétés

Écrivez le code Python qui effectue les actions suivantes :

a) Allumer la lampe

b) Ouvrir le tiroir

c) Marquer le livre comme lu

d) Eteindre la lampe

Ex 3

Exercice 3 : Interaction simple - Exemple

Voici un exemple d'interaction : cliquer sur un coffre pour l'ouvrir et révéler une clé.

```
1  # Définition des objets
2  coffre = {
3      "nom": "coffre",
4      "x": 300,
5      "y": 350,
6      "visible": True,
7      "ouvert": False
8  }
9
10 cle = {
11     "nom": "clé",
12     "x": 320,
13     "y": 370,
14     "visible": False, # Cachée au début
15     "ramasse": False
16 }
17
18 # Dans la gestion des clics
19 if event.type == pygame.MOUSEBUTTONDOWN:
20     # Clic sur le coffre
21     if rect_coffre.collidepoint(event.pos) and coffre["visible"]:
22         if not coffre["ouvert"]:
23             coffre["ouvert"] = True
24             cle["visible"] = True # La clé apparaît !
25             message = "Le coffre contenait une clé !"
```

Question : Que se passe-t-il quand on clique sur le coffre ?

Ex 4

Exercice 4 : Interaction simple - À compléter

Complétez le code suivant pour créer cette interaction :

- Cliquer sur le tableau le fait basculer et révèle un coffre-fort caché derrière

- Le tableau a une propriété "basculer" qui passe à True
- Le coffre-fort a une propriété "visible" qui passe à True

```

1  # Définition des objets
2  tableau = {
3      "nom": "tableau",
4      "x": 200,
5      "y": 150,
6      "rect": pygame.Rect(200, 150, 150, 100),
7      "visible": True,
8      "basculer": False
9  }
10
11 coffre_fort = {
12     "nom": "coffre-fort",
13     "x": 210,
14     "y": 160,
15     "rect": pygame.Rect(210, 160, 130, 80),
16     "visible": False,
17     "ouvert": False
18 }
19
20 # Dans la gestion des clics - À COMPLÉTER
21 if event.type == pygame.MOUSEBUTTONDOWN:
22     pos = event.pos
23
24     # Clic sur le tableau
25     if tableau["rect"].collidepoint(pos):
26         if not tableau["basculer"]:
27             # Complétez ici
28             tableau[_____] = _____
29             coffre_fort[_____] = _____
30             message = "Le tableau bascule ! Un coffre-fort est
caché derrière !"

```

Ex 5

Exercice 5 : Proposez un scénario

En vous inspirant des exercices précédents : Décrivez les objets à l'aide de dictionnaires. Proposez un (extrait de) programme python montrant l'évolution des valeurs stockées dans les dictionnaires décrivant ces objets, en imaginant le mécanisme du jeu.