

# Afficher un personnage à l'écran en PyGame

L'affichage d'un personnage est une étape essentielle dans un jeu vidéo. Voici comment le charger et l'afficher dans une fenêtre PyGame.

Pour pouvoir afficher un personnage à l'écran, il faut connaître sa position et l'image utilisée pour le représenter.

Comme on l'a aperçu précédemment, la position du joueur est gérée via des coordonnées x et y. Ces coordonnées seront stockées dans des variables.

Pygame permet de charger facilement une image.

🔥 Dans la plupart des cas, tu devras utiliser des images avec un **fond transparent**. Tu utiliseras donc généralement des images au format **PNG**.

## Charger et afficher une image

### 1. Charger l'image du personnage :

```
1 | player = pygame.image.load('images/player.png')
```

Tu peux télécharger des images de personnages depuis la section téléchargement. Remplace le nom de fichier par celui dont tu as besoin et déplace l'image dans le dossier `images` de ton de projet (dossier que tu créeras au besoin).

### 2. Définir sa position initiale :

```
1 | player_x = 100  
2 | player_y = 100
```

*Player* en anglais signifie joueur en français. On a choisi d'appeler la variable comme ça parce qu'elle représente la position `x` et la position `y` d'un player. On aurait pu l'appeler `joueur_x`, `perso_x`, `personnage_x`... Au choix, tant que ça a du sens pour s'y retrouver plus tard.

### 3. L'afficher à l'écran :

```
1 | screen.blit(player, (player_x, player_y))
```

## Exemple complet

```
1 | import pygame  
2 | import sys  
3 |  
4 | # Initialisation  
5 | pygame.init()
```

```

6
7 # Création de la fenêtre de jeu
8 screen = pygame.display.set_mode((800, 600))
9 player = pygame.image.load('player.png')
10
11 clock = pygame.time.Clock()
12
13 # position initiale du joueur
14 player_x = 100
15 player_y = 100
16
17 # Boucle de jeu
18 running = True
19 while running:
20     for event in pygame.event.get():
21         if event.type == pygame.QUIT:
22             running = False
23
24     # Mise à jour de l'écran
25     screen.blit(player, (player_x, player_y))
26     pygame.display.flip()
27
28     # Limiter la vitesse d'exécution (60 FPS)
29     clock.tick(60)
30
31 pygame.quit()
32 sys.exit()

```

Ce code est identique à celui réalisé dans l'introduction. Dans les articles suivants, nous allons animer le personnage et ajouter un fond d'écran.

## Astuces

---

- Assure-toi que l'image du personnage est au format **PNG** avec un **fond transparent**.

Avec ces bases, ton personnage apparaît à l'écran, prêt pour être animé et déplacé !