

20.11. Réflexion sur les classes et les instances

Vous pouvez maintenant imaginer quelques raisons pour lesquelles vous voudrez peut-être définir une classe. Vous avez vu des exemples de création de types plus compliqués ou spécifiques que ceux intégrés à Python (comme des listes ou des chaînes). `Turtle`, avec toutes les variables et méthodes d'instance que vous avez apprises à utiliser plus tôt au cours du semestre, est une classe définie par les programmeurs et qui est désormais incluse dans le langage Python. Dans ce chapitre, nous avons défini `Point` certaines fonctionnalités qui peuvent faciliter l'écriture de programmes impliquant `x, y` des `Point` instances de coordonnées. Et brièvement, vous verrez comment vous pouvez définir des classes pour représenter des objets dans un jeu.

Vous pouvez également utiliser des classes auto-définies pour contenir des données - par exemple, les données que vous obtenez en effectuant une demande à une API REST.

Avant de décider de définir une nouvelle classe, il y a quelques points à garder à l'esprit et des questions que vous devriez vous poser:

- **Quelles sont les données que vous souhaitez traiter?** (Des données sur un tas de chansons d'iTunes? Des données sur un tas de tweets de Twitter? Des données sur un tas de recherches de hashtag sur Twitter? Deux nombres qui représentent les coordonnées d'un point sur un plan en 2 dimensions?)
- **Que représentera une instance de votre classe?** En d'autres termes, ce qui sorte de nouvelle chose dans votre programme devrait avoir des fonctionnalités de fantaisie? Une chanson? Un hashtag? Un tweet? Un point? La réponse à cette question devrait vous aider à décider comment appeler la classe que vous définissez.
- **Quelles informations chaque instance doit-elle avoir comme variables d'instance?** Ceci est lié à ce qu'une instance représente. Voyez si vous pouvez en faire une phrase. *"Chaque instance représente une <chanson> et chaque <chanson> a un <artiste> et un <titre> comme variables d'instance."* Ou, *"Chaque instance représente un <Tweet> et chaque <Tweet> a un <utilisateur (qui l'a publié)> et <une chaîne de contenu de message> comme variables d'instance."*
- **Quelles méthodes d'instance chaque instance doit-elle avoir?** Que devrait être capable de faire chaque instance? Pour continuer à utiliser les mêmes exemples: Peut-être que chaque chanson a une méthode qui utilise une API de paroles pour obtenir une longue chaîne de ses paroles. Peut-être que chaque chanson a une méthode qui renvoie une chaîne du nom de son artiste. Ou pour un tweet, peut-être que chaque tweet a une méthode qui renvoie la longueur du message du tweet. (Devenir fou!)
- **À quoi doit ressembler la version imprimée d'une instance?** (Cette question vous aidera à déterminer comment écrire la `__str__` méthode.) Peut-être, "Chaque chanson imprimée affichera le titre de la chanson et le nom de l'artiste." ou "Chaque Tweet imprimé affichera le nom d'utilisateur de la personne qui l'a publié et le contenu du message du tweet."

Après avoir examiné ces questions et pris des décisions sur la façon dont vous allez commencer avec une définition de classe, vous pouvez commencer à définir votre classe.

'variables de classe et variables d'instance' > (ClassVariablesInstanceVariables.html)

20.12. Tester les class

N'oubliez pas qu'une définition de classe, comme une définition de fonction, est une description générale de ce que *chaque instance de la classe devrait avoir*. (Chaque point a un `x` et un `y`.) Les instances de classe sont spécifiques: par exemple le point avec *un `x` et `y` spécifiques*. Vous pourriez avoir un point avec

une `x` valeur de 3 et une `y` valeur de 2, donc pour cette *instance* particulière de la *classe* `Point` , vous passeriez dans `3` et `2` au constructeur, la `__init__` méthode, comme ceci:, comme vous l'avez vu dans les dernières sections. `new_point = Point(3,2)`

Vous avez essayé 1 de 1 activités sur cette page

20.10. Variables de classe et variables d'instance ">



(ClassVariablesInstanceVariables.html)

20.12. Cours de test ">



(TestingClasses.html)

Marquer comme terminé