Quiz 1

Assignment

The starter code contains a Food class (in `food.py`) and a short `main.py` to illustrate the use of Food. You will add some methods to the Food class and create a new class named FoodDiary.

Problem 1: Food class

The starter code for Food contains these methods & properties:

__init__(name, weight, calories, carbs)	Create a new food with the given name, weight (grams), calories (kcal), and carbs (grams).
name {property}	Return the food name.
weight {property}	Return the food weight.
calories {property}	Return the food's energy value (calories).
carbs {property}	Return the number of grams of carbohydrates.

Write code in the Food class for the following behavior

percent_carbs()	Return the fraction of the calories that are from carbohydrates, as a number between 0 and 1. 1 gram of carbohydrate provides 4 calories, so multiply by 4 to convert carbohydrates to calories.
__str__	The string value of a food is the name, then a space, and the weight in parenthesis, such as: `'Brown Rice (100g)'`
food1 == food2	Write a method so that two foods are equal only if they have the same name and same weight. Use the standard template for equals methods. Do not use str(food) to test equality.
food1 + food2	Add two foods. The rules are: if food1 and food2 have the same name, then return a new food with name name but add the weight, calories, and carbs of the two foods. if food1 and food2 have different names, raise a ValueError with a descriptive message.
food * number	Write a method so we can multiply a food by a number to create a different quantity of the same food. Return a new Food with the calories, weight, and carbs scaled by the number.

Problem 2: FoodDiary

Write a FoodDiary class in the file food.py that stores a record of foods we consume.

__init__()	Initialize a new, empty FoodDiary
add_food(food)	Add a food to the diary. If the name of the food matches a food that is already in the diary, then add to the quantity of the existing food object, so each food occurs only once in the diary.
calories {property}	Return the total calories of all foods in the diary.
carbs {property}	Return the total carbs of all foods in the diary.
__str__	Return a string in this format (where N is a digit or number) "N foods with N,NNN calories"
percent_carbs()	Returns the fraction of the total calories from carbohydrates. 1 gram of carbohydrate provides 4 calories.

Problem 3: Unit Tests

Write 3 unit tests of Food in a file name test_food.py

1. Test that == works correctly. Compare foods with the same name and different weights, or different names and the same weight.
2. Test that food1 + food2 works when both foods have the same name (should succeed).
3. Test that food1 + food2 raises an exception when the foods have different names.

Example

```
>>> tofu = Food("Tofu", 100, 110, 2.4)
>>> egg = Food("Fried Egg", 60, 114, 0.6)
>>> str(egg)
"Fried Egg (60g)"
>>> sum = tofu + tofu
>>> type(sum)
food.Food
>>> str(sum)
"Tofu (200g)"
>>> sum.calories
220
>>> sum = Food("Rice", 150, 180, 36) + egg
ValueError: Cannot add different types of food
>>> tofu == egg
False
```