Asynchronous Programming


asyncio

Asynchronous requires changing how you thing about structuring performed

https://github.com/hanattaw/asyncio_class

Asynchorous

- How best to best perform **multiple tasks** at the **same time**
- Not just any tasks, but **specifically tasks** that involve **waiting periods**
- “While you **wait** for this task to **complete**, work on **other tasks** can be performed”

Threadbot Inc.

- Each of your **three WaitBots** at different tables is waiting for one of the diners at their table to give their order. The **WineBot** has already prepared 17 drinks, which are now **waiting** to be collected (it took only a few seconds), and is **waiting** for a new drink order. One of the **GreetBots** has greeted a new party of guests and told them they need to **wait** a minute to be seated, and is **waiting** for the guests to respond. The other GreetBot, now processing a credit card payment for another guest that is leaving, is **waiting** for confirmation on the payment gateway device. Even the ChefBot, who is currently cooking 35 meals, is not actually doing anything at this moment, but is simply **waiting** for one of the meals to complete cooking so that it can be plated up and handed over to a WaitBot.

Threadbot Inc.

- On Sunday, you add a **data collection** module to your ThreadBots. For each ThreadBot, you're **measuring** how much time is spent **waiting** and how much is spent **actively** doing work. Over the course of the following week, the data is collected.

- Then on Sunday evening, you analyze the **results**. It turns out that even when your restaurant is at full capacity, the most hard-working **ThreadBot is idle about 98%** of the time. The ThreadBots are so enormously efficient that they can perform any task in fractions of a second.