Let's say that we are aspiring chefs, and we're following a recipe to cook chicken parmesan with a side-salad.

1) Set out a bowl of flour, a bowl of breadcrumbs, and some beaten eggs.
2) Flour the chicken.
3) Dip the chicken in your batter.
4) Hit that chicken with some breadcrumbs.
5) Throw the chicken in a preheated oven, and set a timer to 20 minutes.
6) Take the chicken out of the oven.
7) Pour marinara sauce over the chicken.
8) Sprinkle mozzarella on top.

The first five steps are fairly simple. They take us through the process of making chicken parmesan, one step at a time, in a linear fashion. Step number five, however, has us setting a timer for twenty minutes, and then, if we were following the recipe word-for-word, removing the chicken from the oven immediately afterwards, before it ever had the chance to finish cooking. That's certainly not what we want to happen since no one wants to eat raw chicken. As such, we should specify in our recipe that steps 6, 7, and 8 should only occur when the timer has gone off.

1) Set out a bowl of flour, a bowl of breadcrumbs, and some beaten eggs.
2) Flour the chicken.
3) Dip the chicken in your batter.
4) Hit that chicken with some breadcrumbs.
5) Throw the chicken in a preheated oven and set a timer to 20 minutes.*

*WHEN the timer has gone off...
5.1) Take the chicken out of the oven.
5.2) Pour marinara sauce over the chicken.
5.3) Sprinkle mozzarella on top.

By doing this, step 5 must reach its completion before any future steps can be taken with our chicken. This is a great example of having a callback within code since one function (the finishing touches on our meal) requires a previous function (the preparation of the chicken) to reach completion beforehand.

Since we don't want to just wait around for 20 minutes doing nothing, however, we can jump tracks to our second recipe and work on that side-salad whilst waiting for our original five steps to reach completion. As such, we are no longer working synchronously on one recipe, but instead, we're working asynchronously on two recipes at once (working on one during the downtime of another).