Being explicit

Mark Bools

February 28, 2021

Last Modified: February 28. 2021

Abstract

There are many opportunities for developers to rely on implicit defaults. In this article I make the general proposition that it is better to be explicit rather than rely on defaults.

I am as guilty as anyone of relying on defaults. That said, it's a bad habit.

The main problem is that defaults change from time to time. As the creator of a service (API, library, etc.) it makes sense to provide default values. They make the service easier to consume and simplifies writing the service. As the consumer of a service defaults can certainly appear to make using the service simpler but at the cost of potential breakages when the defaults change.

Let's look at a trivial (and horribly contrived) example to make the point. Suppose I have created a library to draw a circle.

```circles.py
1  def drawCircle(radius=10):
2      pass
```

My interface provides a default radius (perhaps the most common size for some graphic I want to create). This is convenient and users of my library use this default, calling the routine as `drawCircle()`.

The issue is that `drawCircle` is now 'locked-in' to this default. If I decide to change the default (perhaps my application need cicles of diameter 20 now and the quickest way to make the change in *my* application is the change the default). The problem is doing so will wreak havoc in applications that used the default and continue to need circles of radius 10.

As a real-world example of the changing defaults issue, consider the cookie `SameSite` attribute. Originally cookies were broadly exchanged but problems emerged with cookies being misappropriated so the `SameSite` setting was added. Initially the default was `None` (to match behviour before `SameSite` became available) but in Chrome 80 this was to change to `Lax`. The problem with this change being that it would break many existing resource links (e.g. `<img>`) while letting links (e.g. `<a>`) continue to work. This change was pushed out (due to the disruption the change caused) to Chrome 85).