Hi, my name is Ben, my pronouns are he/him and I'm here to tell you How to Get Nothing Done. By following this advice you will minimise your productivity and spend the most time doing it possible.

And the good news is you're already following some of these simple steps.

The first step to getting nothing done is to always be working, which may sound a little counter-intuitive, so let's examine what happens when you decide you're going to be working 100% of the time.

First you need to pick something to do.

Let's not forget, you need to always be working, so you need that task as soon as possible. This means we've got to pick up a task before we can be sure it's the most important thing. If you're lucky, it won't be useful at all. Maybe it will be detrimental!

In 2009, Microsoft found only a third of their features improved the metrics they were meant to. Because we're always working, we won't have time to validate assumptions or run experiments. Just straight into our task.

Now, any time that you can't work on your task, we have to pick up a new one. Otherwise we wouldn't be working. We would much rather start a new task than stop working for even a second.

Before long, we'll have loads of tasks, and that's one of our main tactics. The more tasks we have on the go the harder it is to finish any single task. After all, we've only been productive if we complete a task.

Another huge benefit of having a few tasks on the go, is we don't have to work very hard to lose track of a task. Your less important tasks will

naturally slip through the cracks because you can only find short periods to work on them and most of *that time* is spent context switching.

These lost tasks often either find themselves completely abandoned or handed over to someone else. Either the work you've already done is *wasted*, or you can *both* waste your time passing over context.

But, even better, you can prioritise all tasks equally. This means you can rotate through them with the illusion of being more productive, while focusing on neither task and context switching as often as you want. Here we see the "Always working" approach takes a lot longer to get either task finished than the "One at a time" approach.

This is great, because it lets us delay finishing any particular task. Delaying finishing anything is where the magic happens.

By increasing the amount of time that passes before we finish a task we have less time benefitting from whatever the task was meant to achieve and Increase the chance that an emergency is going to come up. And we *can* rely on emergencies to come up.

This emergency will now become your new top priority. That task that was precariously close to being finished has now become an afterthought. Your second priority will be completely forgotten.

If you weren't always working, this emergency might be no big deal. But always working has meant that the emergency has caused the maximum disruption and caused the most delay possible. Which is exactly what we're trying to achieve.

I want to stress here that while I'm talking about emergencies, you will achieve this same disruption with any new priority or any task that comes with a deadline.

Now, while we *could* wait for the tasks to come up organically, why not skip this step.

Start with more than one task and actively volunteer for more. Be more than 100% busy. Now you get all the problems from before without the benefit of any increased efficiency.

And, while small tasks are fine, they are easier to finish. Consider working in really big chunks. Spend a month working on a task and wait until you think it's done. Now we take your big load of work and show it to someone to get feedback

They are definitely going to have suggestions. So you've now got a lot more work to do, and you've still not *finished* anything. As a bonus, while you're waiting for that feedback, you can start a new task!

Finally, we need to avoid collaboration at all costs. Collaboration will shortcut our long feedback loops. If we pair program we might not need code reviews. If you work closely with your customer, you might reduce their changes. You have to work alone as long as possible.

If you follow the "always be working" principle, I can promise you'll find yourself forever working, but you get nothing finished. When you do finish something, it's months down the line and had much less impact than it should have.