So our next big O notation is O of one.

So let's start out with some code.

Here we have a function called add items.

And all we're going to do is we're going to return n plus n.

There's only one operation here and that's the addition.

So it doesn't matter if n is one or n is a million.

The number of operations is going to be one and that is O of one.

Well, what if we add.

Another edition.

Well, now that's two operations, right o of two.

And as you may have guessed, we're going to simplify this and make it O of one.

So O of one a lot of times is referred to as constant time.

So this situation here where we have two editions.

Its two operations.

That is constant no matter what N is.

The number of operations does not change as n changes.

So let's take a look at this on the graph.

Just a flat line across the bottom.

It is the most efficient big O.

Nothing is more efficient than O of one.

Remember you will also hear this referred to as **constant time.**

So if you hear that phrase, that is o of one.