

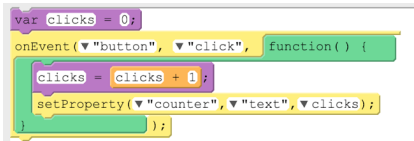
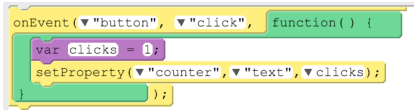
Create Variables Once, At the Top, Outside onEvent()

When you create variables you should:

- **Use var only once.** You don't need to create variables twice and this can cause errors.
- **Create your variables at the top of your program.** This keeps your code organized and easier to read for you and others.
- **Create your variables outside any onEvent blocks.** Read on to understand why.

Global vs. Local Variables

There's two types of variables, global and local, and so far we've only used global variables. Here's the main difference between global and local variables.

Type of Variable	How It Works	How Created	Picture
Global	Permanent. Can be used anywhere in your code	var used outside an onEvent	 <pre>var clicks = 0; onEvent(▼"button", ▼"click", function() { clicks = clicks + 1; setProperty(▼"counter", ▼"text", ▼clicks); });</pre>
Local	Temporary. Can be used only in the part of the code where it was created, like inside an onEvent. Deleted once the onEvent is done running.	var used inside an onEvent	 <pre>onEvent(▼"button", ▼"click", function() { var clicks = 0; setProperty(▼"counter", ▼"text", ▼clicks); });</pre>

Avoiding Local Variables and Debugging

Local variables will eventually be useful but for now they're most likely to just be confusing. The biggest issue you'll run into right now with local variables is accidentally using `var` inside of an `onEvent`. Here's what the code usually looks like:

```
var count = 0;

onEvent (▼ "button1", ▼ "click", function() {
  var count = count + 1;
})
```

This code is pretty confusing. While it looks like there's only one variable being used, it actually has two variables, one local, and one global, and they're both named `count`! Changing the value of one will have no impact on the other. This can cause unexpected behavior in your code and it can get tricky to debug.

The best way to avoid these issues is to **make sure for now that you're not using `var` inside of an `onEvent`**. If you run into a tricky debugging problem, check if you're accidentally creating a local variable.