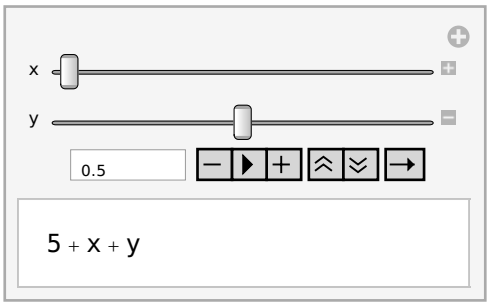
Using Manipulate[] with global variables

In[77]:=
```
equation = x+y+5;
Manipulate[equation,{x,0,1},{{y,0.5},0,1}]
Style["   WTF?   ",FontSize→30,FontWeight→Bold]
```

Out[78]=



Out[79]=

WTF?

This issue often comes up when using Manipulate[] to tune parameters in a large equation.

Explanation

Essentially, Manipulate[] creates temporary variables for each of the things you are manipulating that are only available inside the Manipulate itself. That means the "x" in the first line and the one in the Manipulate aren't actually the same variable. The following three solutions can be useful in different contexts.

Fix 1: Replace external variables with the internal ones

This is the quick and simple solution I have used before when encountering this problem. It is a little bit ugly (particularly if you need to change which variables are being manipulated), but doesn't require many changes to the code.

In[29]:=
```
fix1`equation = x+y+5;
Manipulate[fix1`equation/.{x→x2,y→y2},{x2,0,1},{y2,0,1}]
```

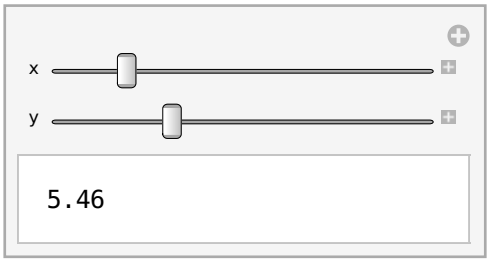
Out[30]=

Fix 2: Move everything inside Manipulate

This solution is bad if the number of steps needed to build up the equation is large, and can result in bad code duplication, but does work. It is also less computationally efficient, making it less responsive.

In[39]:=
```
Manipulate[fix2`equation = x+y+5;
fix2`equation,{x,0,1},{y,0,1}]
```

Out[39]=



Fix 3: Inject the solution into the Manipulate

This is a really cool solution using the ReplaceAll operator (->) to inject the equation inside of the Manipulate[]. If you are manipulating lots of variables, it is the cleanest solution I have seen.

In[42]:=
```
fix3`equation = x+y+5;
Manipulate[temporaryvariable,{x,0,1},{y,0,1}]/.temporaryvariable→fix3`equation
```

Out[43]=

In[87]:=
```
SetDirectory[NotebookDirectory[]];
Export["Manipulate variable scoping.pdf", EvaluationNotebook[]];
```