# Writing Conditional Statements in the ArcGIS Field Calculator

The Python code-block in the ArcGIS Field Calculator can be used to run mathematical and logical statements using Python script. This is useful when you need to enter values into a field depending on the values in another field. In this example I use a Conditional Statements in the ArcGIS Field Calculator to populate a field with Miles Per Hour (MPH) values depending on the road type.

Advantages:

* Only write the code once
* Can update the numbers later and re-run if needed
* can save the script in ArcMap

Disadvantages:

* No error checking help
* The syntax is slightly different
* Issues with overwriting data that is not covered by the code

This is the cross walk from the published research and the road types in my attribute table.

Table 1

|  |  |  |
| --- | --- | --- |
| **Categories From Han** | **MPH** | **My Categories (StatusSurface)** |
| Spur | 10.9 | Spur dirt, Spur Rock |
| 1 ½ Lane | 13 | Secondary Dirt, Secondary Rock, Mainline dirt, Public Dirt |
| Gravel | 20 | Mainline Rock |
| Highway | 34.7 | Public Paved |
| Interstate | 50.5 | Highway Paved |

I need the MPH field to populate based on the road type in the StatusSurface field.



First, right click on the field header of the field you want the calculation to take place on – in my case this is the MPH field. Click on ***Field Calculator***. Make sure the ***Python*** radio button is selected (beware, it will randomly unselect itself), and check the box for ***Show Codeblock***.



Write your def function in the codeblock and call the function in the box below. When referencing a field name in the codeblock, don’t use exclamation points, but do use them in the lower box when you call the function.

Indentation must be two or four spaces. Single or double quotes are acceptable for strings.

You can press the save button to save your scripts in ArcMap. They can be loaded later to perform the same function on the same fields. This was very useful in my project when I needed to go back and change some things to an earlier version of the road layer. All I had to do was add these fields back in, load the python script and re-run it.

A note on overwriting:

If you already have values in the MPH field, and enter a statement that only references one road type, all the other values will be overwritten with <Null> no matter what road type they are. I addressed this by including all road types in my code. I tried to find a solution but my solution only worked when the conditional statement was simple (only used one road type).

If you do this….

def MPH(StatusSurface):

 if StatusSurface =='SpurDirt':

 return 10.9

….and there are already other values in the MPH field, those values will be overwritten with <Null>

If you don’t want your stuff to be overwritten, add else: return (the name of the field)

For example:

The field must be added the list in the function call in the lower box.

This was my final code to populate all cells with a MPH value based on road type:

def MPH(StatusSurface):

 if (StatusSurface =='SpurDirt') or (StatusSurface =='SpurRock'):

 return 10.9

 elif (StatusSurface =='SecondaryDirt') or (StatusSurface =='SecondaryRock') or (StatusSurface =='PublicDirt') or (StatusSurface =='MainlineDirt') :

 return 13

 elif (StatusSurface =='MainlineRock'):

 return 20

 elif (StatusSurface =='PublicPaved'):

 return 34.7

 elif (StatusSurface =='HighwayPaved'):

 return 50.5

MPH( !StatusSurface! )