Essential Apps Script livecoding 10 walkthrough: On edit trigger

This walkthrough guide accompanies the Essential Apps Script 10: On edit trigger video and the Essential Apps Script guide. You can use the video, this document, or both, to help you do this livecoding exercise.

In this exercise, we are going to use an 'on edit' trigger to cause some code to run whenever someone edits a spreadsheet.

Livecoding instructions

1. In your Google Drive folder for the course, create a new Google Sheet and call it 'On edit cell demo' then go to **Extensions** > **Apps Script** to open the Apps Script editor and give the project the same name. Rename the Apps Script file on the left hand side to **onEditCell** and then rename the function to **onEditCell** as well. Put your cursor at the end of the first line, hit Enter twice, and then Save.

```
function onEditCell(){

}
```

2. Create a comment using two forward slashes **//** saying **get spreadsheet** then Enter and we will store the spreadsheet in variable as usual so write **var ss = SpreadsheetApp.getActiveSpreadsheet();**

```
function onEditCell(){

    // get spreadsheet
    var ss = SpreadsheetApp.getActiveSpreadsheet();

}
```

3. We don't need a specific sheet as we are just going to target the active cell, which will be the one that was just edited. So a new comment using **//** saying **get active cell** and then hit Enter. We will store the active cell in a variable so write **var activeCell =** and then we'll use the spreadsheet and the command getActiveCell() so write **ss.getActiveCell()** and end the line with a semicolon **;**

```
function onEditCell(){

    // get spreadsheet
    var ss = SpreadsheetApp.getActiveSpreadsheet();

    // get active cell
    var activeCell = ss.getActiveCell();

}
```

4. Hit Enter twice and now we need a final comment, so write **// set active cell colour** and then hit Enter again. We don't need to store anything in variable, but we are going to work with the active cell, so write **activeCell** and then **.setBackground()** to change the background colour. Inside the brackets it needs a colour as a string, so it needs to be in single or double quotes, so write **'yellow'** (or any other colour you like). End the line with a semicolon and hit Save.

```
function onEditCell(){

    // get spreadsheet
    var ss = SpreadsheetApp.getActiveSpreadsheet();

    // get active cell
    var activeCell = ss.getActiveCell();

    // set active cell colour
    activeCell.setBackground('yellow');

}
```

5. Next, we need to set up a trigger to make this happen when someone edits the sheet. If you hover your cursor over the left hand side of the screen there's some menu options and one is **Triggers**. Click on this and you'll open the Triggers page (if this is the first time you've accessed Triggers there may be a dialogue box welcoming you).

6. Click on the **Add Trigger** button and a window will open. We will need to tweak some of the options here.

    a. Under **Choose which function to run** make sure the function selected is your **onEditCell** function (you shouldn't have any others in the project anyway).

    b. No need to change anything for **Choose which deployment should run**.

    c. No need to change anything for **Select even source** as it will be **From spreadsheet** because it is a spreadsheet cell edit that will cause the trigger.

    d. Under **Select event type** change it to **on edit**.

    e. Under **Failure notification settings** you may want to change it to **Notify me immediately** so you don't get any Apps Script failure emails long after the script has failed or you've been working on it.

7. Click **Save** and it will ask us the authorise the script just like when we run one. It will run on whichever Google account sets up the trigger, regardless of who edits the spreadsheet.

8. Return to the Google Sheet and try editing a cell by writing something in it. It may take a few seconds for the colour to change the first time.

Now you can move on to the next part of section 4.

Reference: full code from exercise:

```
function onEditCell(){

    // get spreadsheet
    var ss = SpreadsheetApp.getActiveSpreadsheet();

    // get active cell
    var activeCell = ss.getActiveCell();

    // set active cell colour
    activeCell.setBackground('yellow');

}
```