Google Scripting Class

Log in to your Clemson Google Account

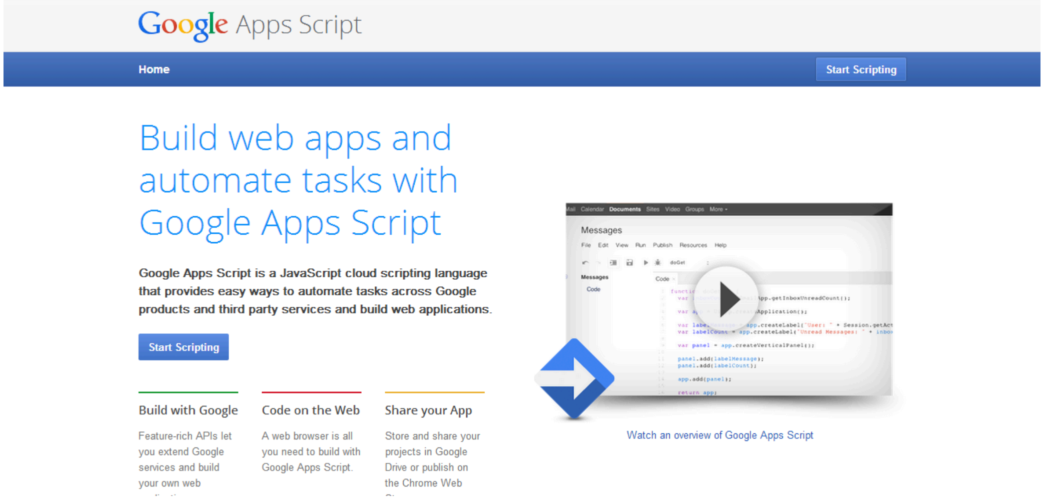
Go to Google Drive and open a new Spreadsheet. Name it Hangman.
Go to Tools, Script Gallery, Fun and Games, and search for **Hangman**. Install the one that says "Google" beside it, Continue, Accept. Tools, Script Manger, Run. Hangman, Start New Game, and play.

Go back to Google Drive and start a second new Spreadsheet. Name it UpsideDown. Type in some words in the first cell.
Go to Tools, Script Gallery, Fun and Games and search for **Flip Letters**. Install. Go to Script Manager and run.

Go to developers.google.com to read more about scripts and requirements. This uses JavaScript.

Go to script.google.com. (Or go to Google Drive, Drive, Create, More, Script.)

You may see the screen below. Click on Start Scripting. This is a one time Welcome Page. It won't recur when you return.



Build web apps and automate tasks with Google Apps Script

Google Apps Script is a JavaScript cloud scripting language that provides easy ways to automate tasks across Google products and third party services and build web applications.

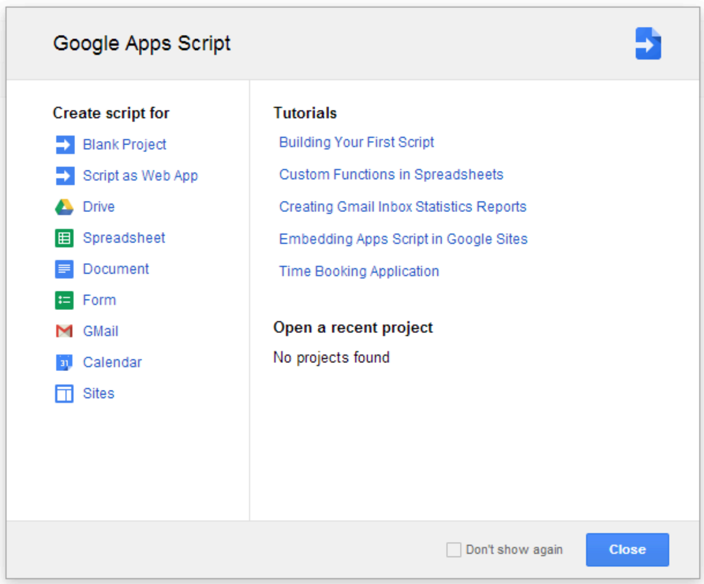
Feature-rich APIs let you extend Google services and build your own web

A web browser is all you need to build with Google Apps Script.

Store and share your projects in Google Drive or publish on the Chrome Web

Messages
File Edit View Run Publish Resources Help
Code
Watch an overview of Google Apps Script

You should see a dialog box that asks you if you want to start writing a script for a specific Google application, like Google Sheets (Spreadsheets app), Google Forms app etc. Click on Blank Project.

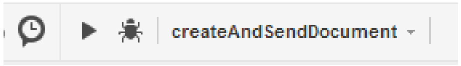


First Example

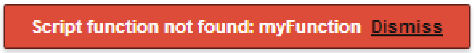
```
Copy and paste this script replacing the default one:

function createAndSendDocument() {
  // Create a new Google Doc named 'Hello, world!'
  var doc = DocumentApp.create('Hello, world!');

  // Access the body of the document, then add a paragraph.
  doc.getBody().appendParagraph('This document was created by Google Apps Script. You may add whatever message you want here.');

  // Get the URL of the document.
  var url = doc.getUrl();

  // Get the email address of the active user - that's you.
  var email = Session.getActiveUser().getEmail();

  // Get the name of the document to use as an email subject line.
  var subject = doc.getName();

  // Append a new string to the "url" variable to use as an email body.
  var body = 'Link to your doc: ' + url;

  // Send yourself an email with a link to the document.
  GmailApp.sendEmail(email, subject, body);}
```

Click on Untitled project and change the name to SendDocument

Click triangle to run:



You may have to accept some security questions.
Run a second time.
If you see this, click Dismiss:



Script function not found: myFunction Dismiss

Look at your Gmail and your Drive to see what it did!

Second Example

Open a new spreadsheet naming it HelloSpreadsheet
Click on Tools, Script Editor, Blank Project
Create new script and replace default with this:

```
function onOpen() {
  var ss = SpreadsheetApp.getActiveSpreadsheet();
  var menuEntries = [ {name: "Say Hi", functionName: "sayHi"},
                      {name: "Say Hello", functionName: "sayHello"} ];
  ss.addMenu("Tutorial", menuEntries);
}

function sayHi() {
  Browser.msgBox("Hi");
}

function sayHello() {
  Browser.msgBox("Hello");
}
```

Rename it to HelloScript
Click Run, Continue, Accept, Close
Back in the spreadsheet, Click Tools, Script Manager, Run, Close
Click Tutorial, Say Hello

Research:

FormEmailer : https://sites.google.com/site/formemailer/

AutoCrat

http://www.steegle.com/websites/google-sites-howtos/apps-script-contact-us-form-email

http://www.blogxpertise.com/2012/04/tip-making-google-doc-submission-form.html

http://www.labnol.org/internet/auto-confirmation-emails/28386/

For

instance, we may do general overview of creating good data sets and then tackling Autocrat together, but then splitting off and doing FormMule for different purposes, as well as formRanger, but Teachers would only get Flubaroo, gClassFolders, Goobric, and Doctopus, whereas admin would dive really deep into FormMule and sheetSpider. Hope that makes sense. You can also see my current Super Scripts material at edutechserve.com/scripts.