

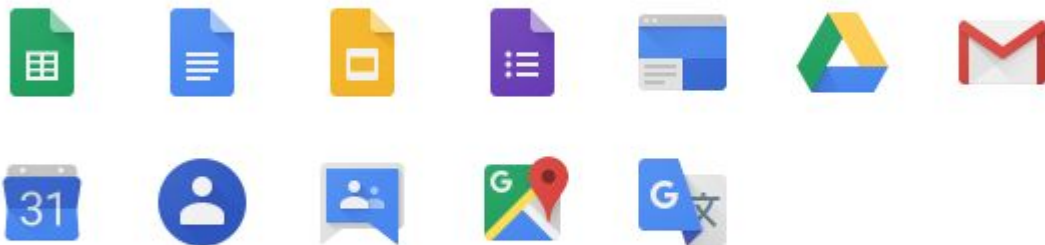
Google Apps Script



Amazing functionality with Google Apps

What is Apps Script???

Many Google apps, one platform in the cloud



Increase the power of your favorite Google apps — like [Calendar](#), [Docs](#), [Drive](#), [Gmail](#), [Sheets](#), and [Slides](#).

Apps Script lets you [do more with Google](#). All on a JavaScript platform in the cloud.

Apps Script

- Javascript based scripting language
- Connecting functionality of Google Apps - like documents, spreadsheets, email, drive and more
- Create and customize how they function
- Comes with its own online IDE (integrated development environment)
- Can be used as a standalone or bound within Google applications

Overview of Google Apps Script : Google Apps Script is a scripting language based on JavaScript that lets you do new and cool things with G Suite products like Docs, Sheets, Slides, and Forms. There's nothing to install

<https://developers.google.com/apps-script/>

What can Apps Script do?

Add menus to Google Docs, Sheets and forms

Add custom functions to Google Sheets

Create and publish web apps

Save time on repetitive task

Use data contained in google suite and do stuff with it.

Complex forms - use sheets as a database

Interact with a wide variety of Google Services - Docs, Sheets, Drive, Gmail, Forms, Maps, Calendar, Sites

What can Apps Script do?

You can do a lot - only limited by your imagination.

Build full featured lightweight web applications

Code in a specialized version of JavaScript customized to access G Suite, and other Google or external services (URLfetch, JDBC, etc.)

Do not have to host your app—it lives and runs on Google servers in the cloud

Getting started with Apps Script

All you need is a Google account (gmail account)

Fundamental JavaScript knowledge is a prerequisite

<https://script.google.com/home> - sign into your Google Account

Two Types of scripts

Bound—meaning it's forever (and only) tied to one Google document (Doc, Sheet, Slide, Site, or Form)

Standalone—an independent script not tied to any G Suite documents

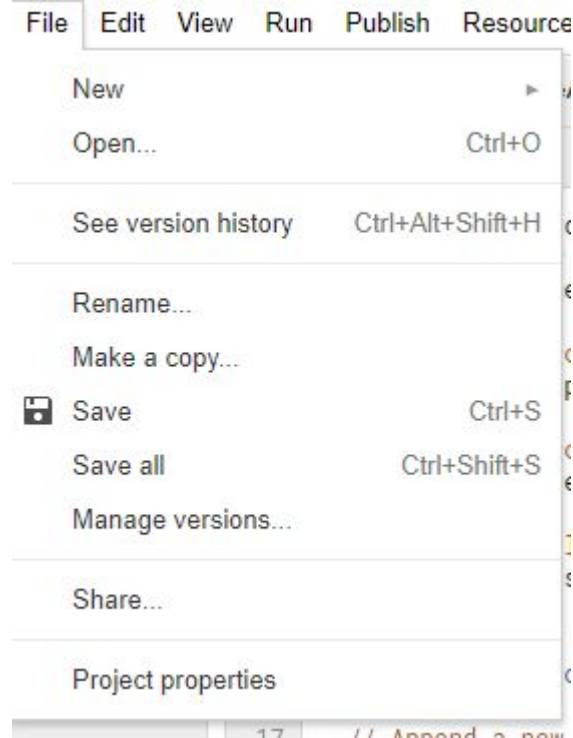
Publishing Scripts

Bound and standalone apps can also be published to expose more broadly:

- Not published—remains private, accessible only to project owners
- Published as an add-on—your app can be installed from the add-on store
- Published as web app—your app handles HTTP requests and has web UI components
- Embedded in Google Sites—published web apps can be embedded in either the new Sites or classic Sites pages
- Published as an API executable—your app can be accessed through the Execution API
- Some valid combination of the above

IDE Introduction

File : create new script and file options.

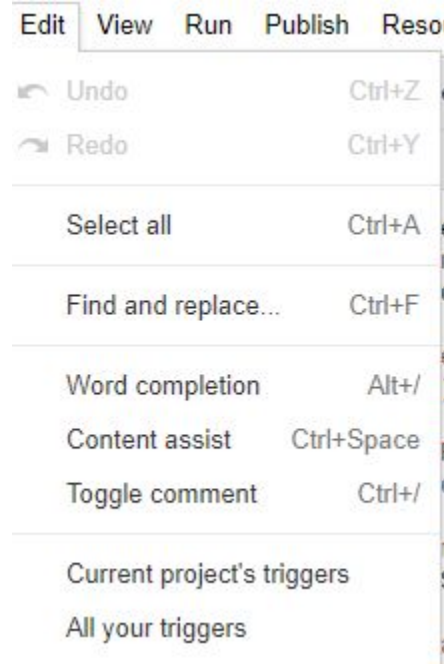


IDE Introduction

Edit: Select and Find options

Several Keyboard shortcuts

Triggers : Set your scripts to run



IDE Introduction

View :

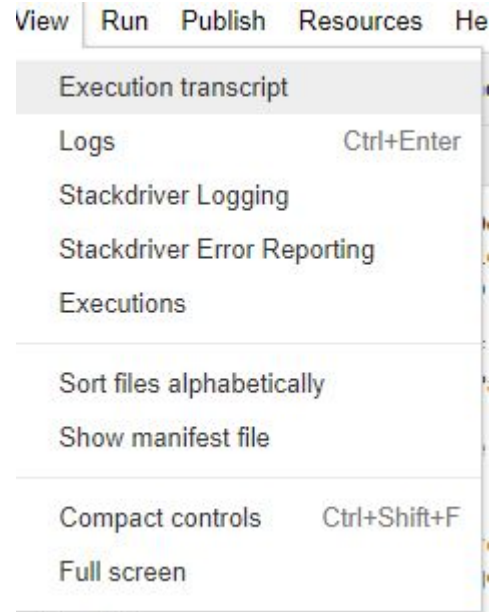
Show what happens when the script runs

Logs : like console log send data;

StackDriver - more detail later in the course

Executions : shows scripts run

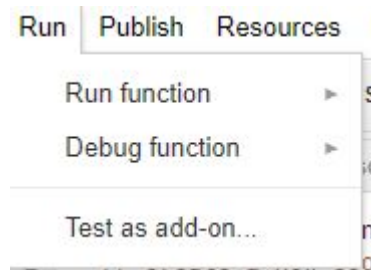
View controls



IDE Introduction

Run : Provides execution of script

Run as add-on to a google application



Authorization required

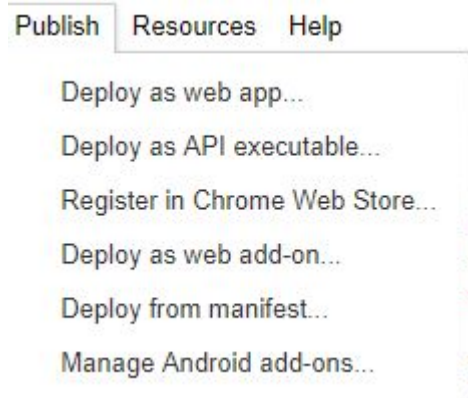
Untitled project needs your permission to access your data on Google.

Review Permissions

Cancel

IDE Introduction

Publish and resources : deploying app will be covered later in the course as well as other options provided here.



IDE Introduction

Help : more help with apps script

Help

Getting started

Documentation

API reference

Tutorials

Support

Report abuse/copyright

Release notes

Welcome screen

Quotas

IDE Introduction

Quick Menu



First Script

Debugging and running script

Project properties under File



The screenshot shows the Google Apps Script editor interface. At the top, there are two tabs: 'Code.gs' and 'appsscript.json'. The 'Code.gs' tab is active, displaying a JavaScript function named 'firstApp()'. The function contains a variable 'welcomeMessage' set to 'Hello World', followed by a 'for' loop that iterates from 0 to 9. Inside the loop, there is a line of code 'Logger.log(welcomeMessage + ' ' + x);' which is highlighted in yellow. A red dot is positioned to the left of this line, indicating a breakpoint. The line numbers 1 through 9 are visible on the left side of the code editor.

```
1 function firstApp(){
2   var welcomeMessage = "Hello World";
3   for(var x=0;x<10;x++){
4     Logger.log(welcomeMessage + ' ' + x);
5   }
6
7 }
8 |
9
```

Classes - Apps Script

<https://developers.google.com/apps-script/reference/>

Listing of services starting with the Calendar Service

Each service has classes, attributes and methods associated with them

Document App

<https://developers.google.com/apps-script/reference/document/>

Properties and methods within the DocumentApp class

Create Document within scripts.

Review options for documents

See how it works

Open you editor
and create some
functions

```
function firstApp(){
  var welcomeMessage = "Hello World";
  for(var x=0;x<10;x++){
    Logger.log(welcomeMessage + ' ' + x);
  }
}

function createDoc(){
  var doc = DocumentApp.create('Hello, world!');
  var body = doc.getBody();
  body.appendParagraph('Some Content in your document');
}

function updateDoc(){
  //how to get ID
  var doc = DocumentApp.openById('***');
  var body = doc.getBody();
  body.appendParagraph('Some New Content in your document : added ' + Date());
}
```