





Putting Together the Pieces

Building Apps with Google Apps Script

Saurabh Gupta
Product Manager, Google

DriveEye

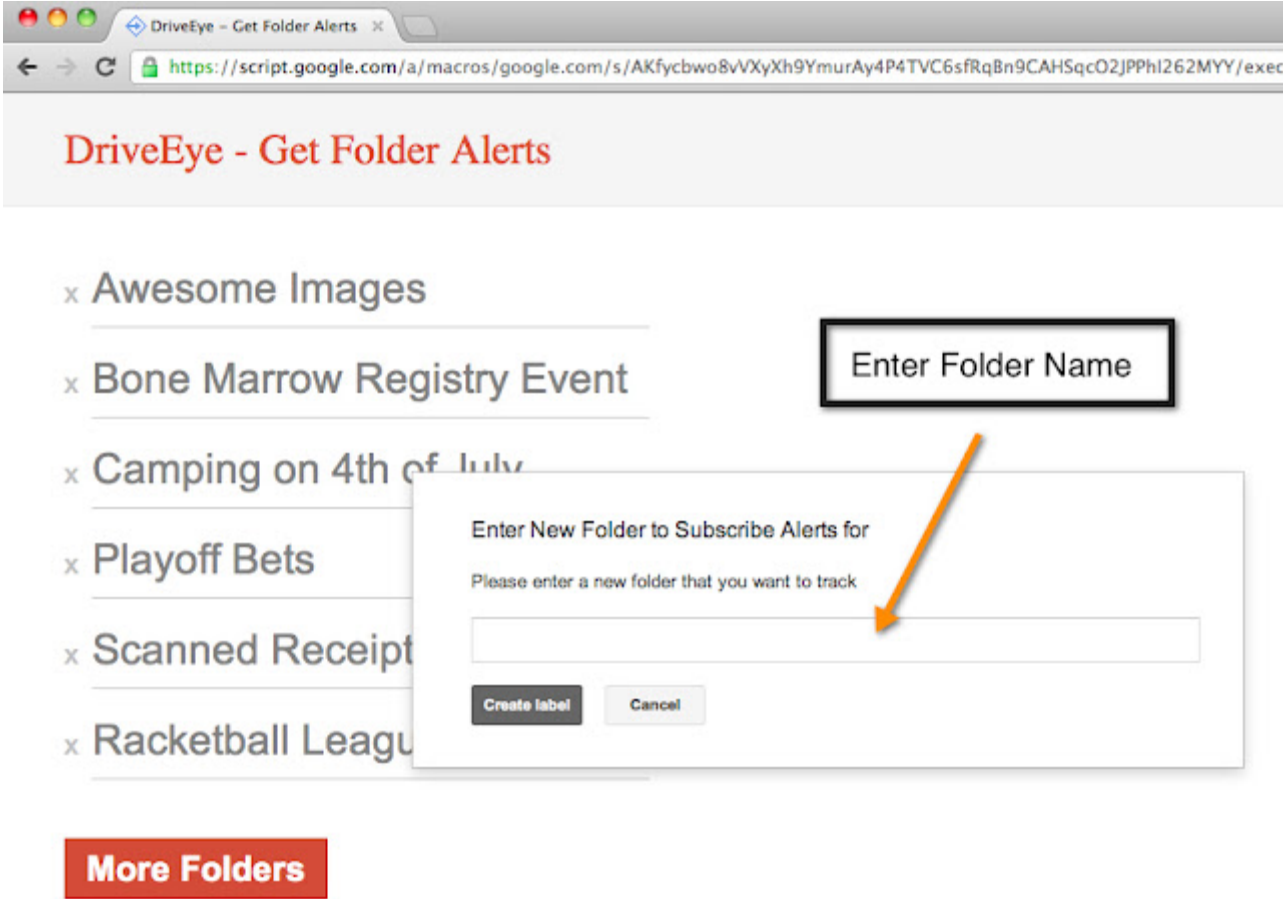
Shared Folder Notifications

Built using Apps Script

<http://goo.gl/90W4e>



DriveEye - Add Folders



The screenshot shows a web browser window with the title "DriveEye - Get Folder Alerts" and a URL starting with "https://script.google.com/a/macros/google.com/s/AKfycbwo8vVxyXh9YmurAy4P4TVC6sfRqBn9CAHSqcO2JPPHi262MYY/exec". The page content includes a header "DriveEye - Get Folder Alerts" and a list of folders with an 'x' icon to the left of each name:


- x Awesome Images
- x Bone Marrow Registry Event
- x Camping on 4th of July
- x Playoff Bets
- x Scanned Receipts
- x Racketball League

A modal dialog box is open over the list, titled "Enter New Folder to Subscribe Alerts for". It contains the text "Please enter a new folder that you want to track" and a text input field. An orange arrow points from a box labeled "Enter Folder Name" to the input field. Below the input field are two buttons: "Create label" and "Cancel".





At the bottom of the page, there is a red button labeled "More Folders".




DriveEye - Add files to subscribed folders


CREATE 

My Drive **Playoff Bets**

<input type="checkbox"/>	TITLE
<input type="checkbox"/> ☆	 2011 Superbowl Bets <small>Shared</small>
<input type="checkbox"/> ☆	 2012 Superbowl Bets Chart <small>Shared</small>
<input type="checkbox"/> ☆	 2012 NBA Playoff Bets <small>Shared</small>
<input type="checkbox"/> ☆	 2012 MLB Bets <small>Shared</small>

My Drive
Shared with me
Starred
Recent
More ▾

DriveEye FolderNotification  Inbox x

 saurabh@gluemesh.com
to me ▾

5:45 PM (6 minutes ago) ☆

DriveEye Alert

(You are getting this email because you have installed DriveEye notifications. DriveEye sends you notification whenever there are any changes to your subscribed folders)

Folder Name Changed	Files Added / Removed
Playoff Bets	<i>(Added file)</i> 2011 Superbowl Bets <i>(Added file)</i> 2012 Superbowl Bets Chart

How to Uninstall DriveEye?

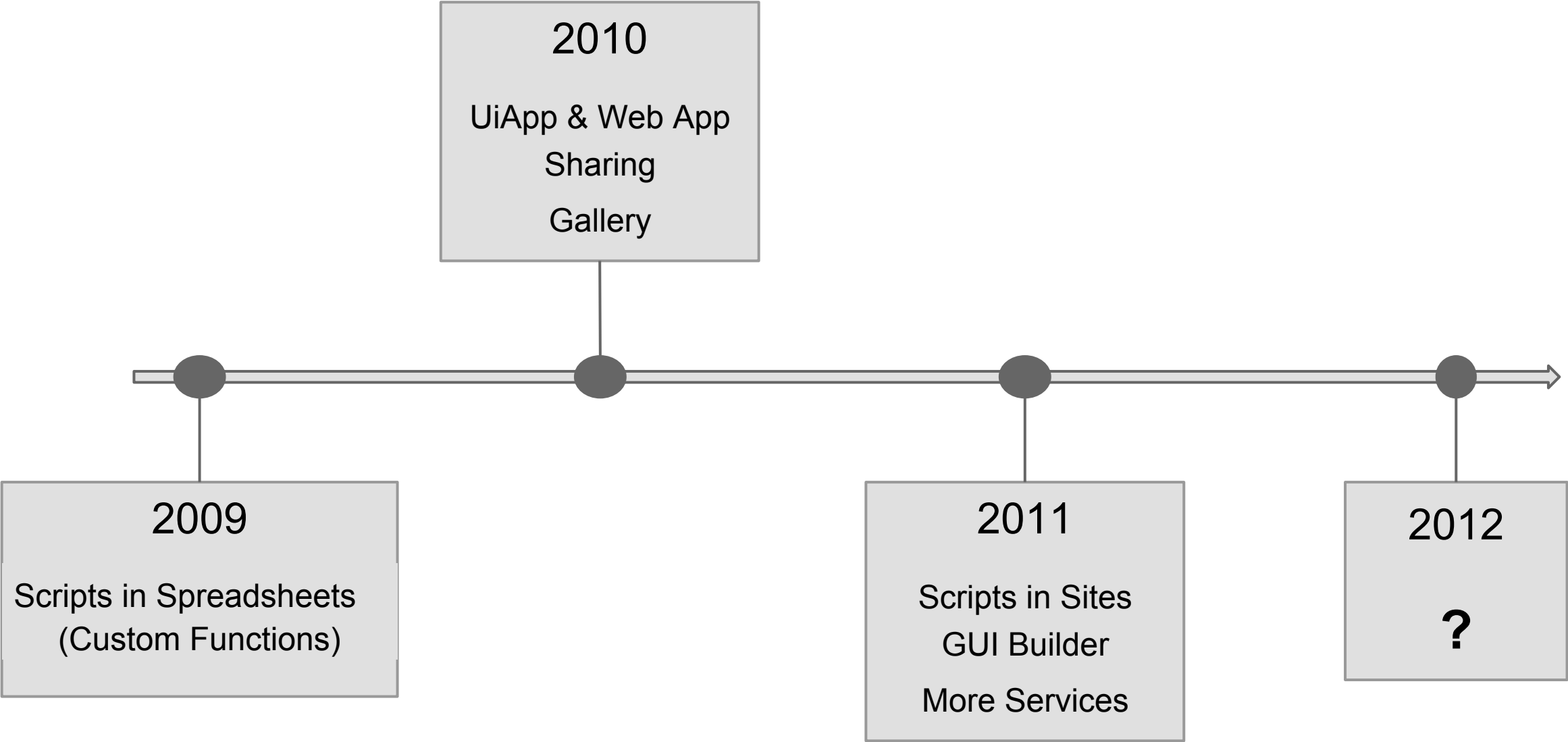
You can click the button below to uninstall DriveEye.

Uninstall DriveEye

 Click here to [Reply](#) or [Forward](#)



Apps Script over the years





Introducing

[script.google.com](https://www.googleapis.com)



Create scripts in
Google Drive

OR

script.google.com

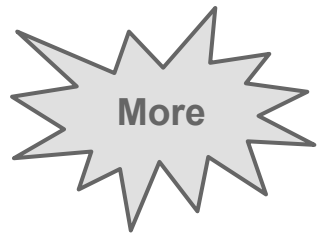


Did anything change?

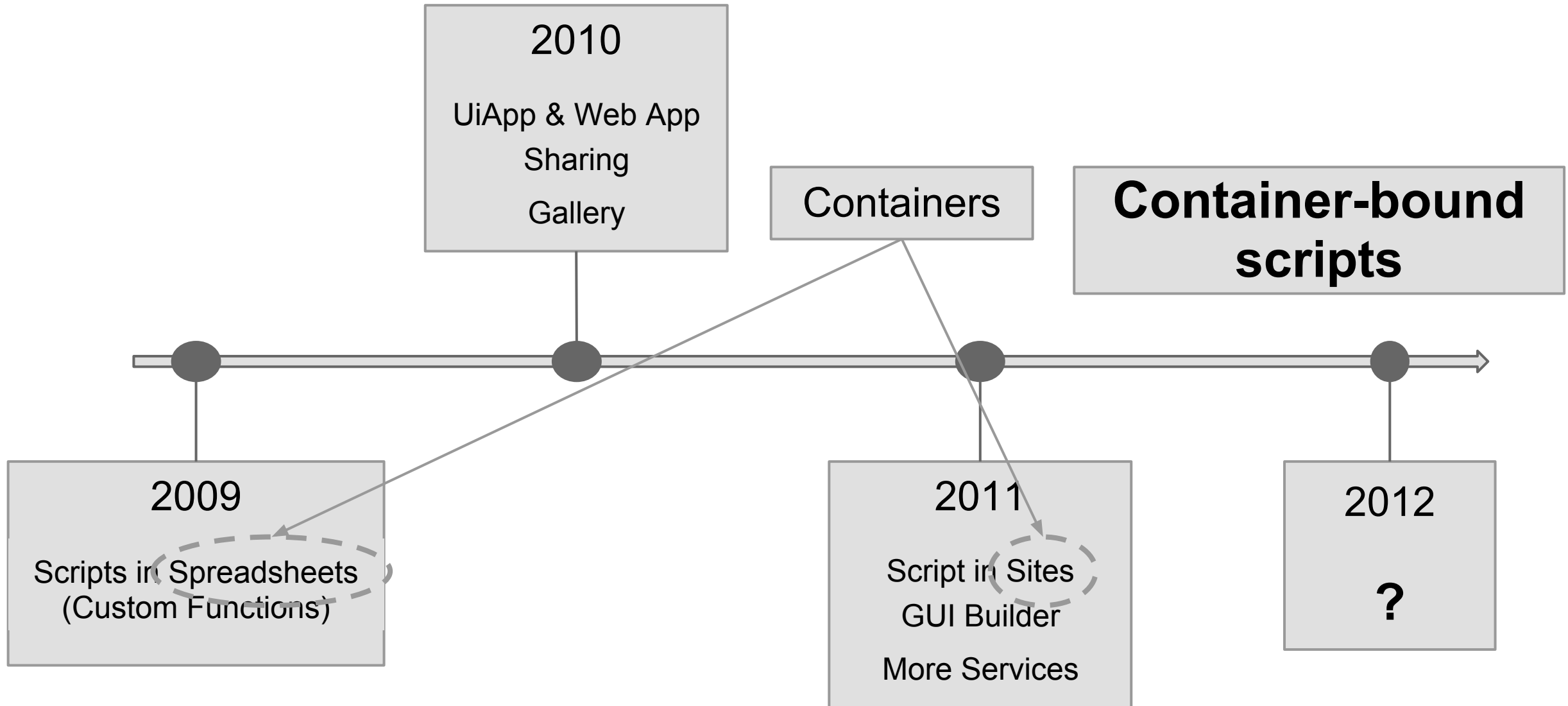
All the features of Apps Script are still available

No changes to Services

No changes to Events and Triggers



Apps Script over the years





Create

New Features of Google Apps Script

1

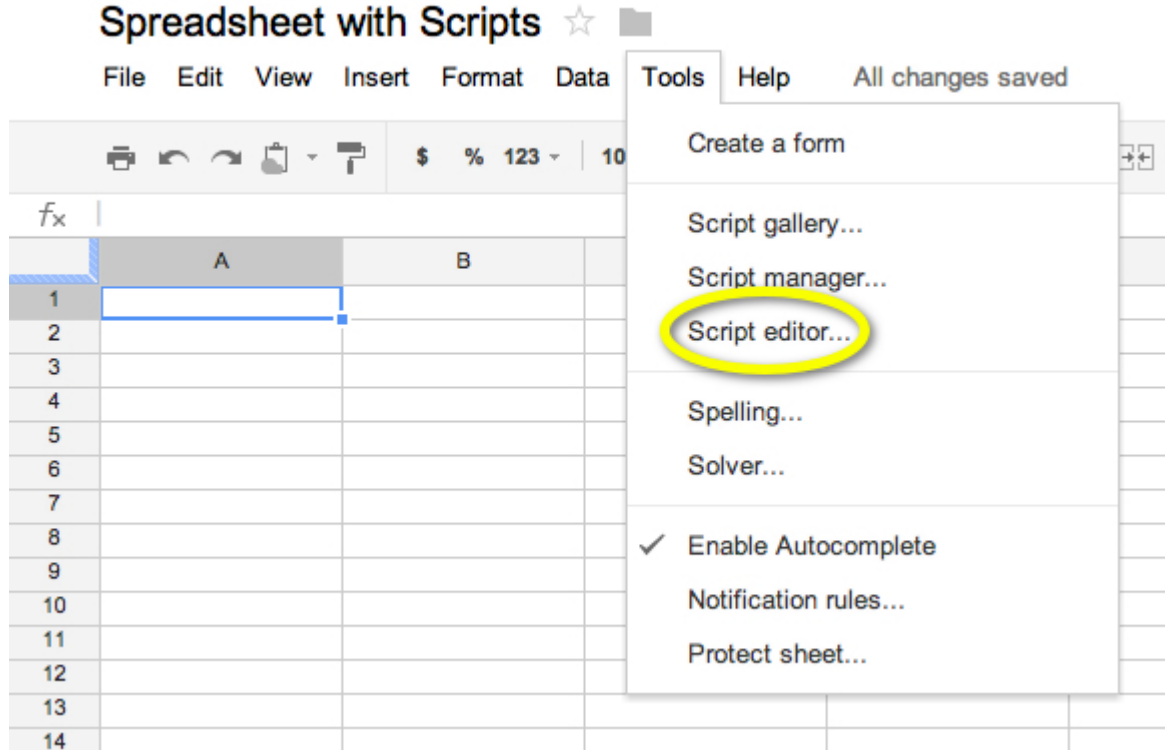
Create Standalone Scripts



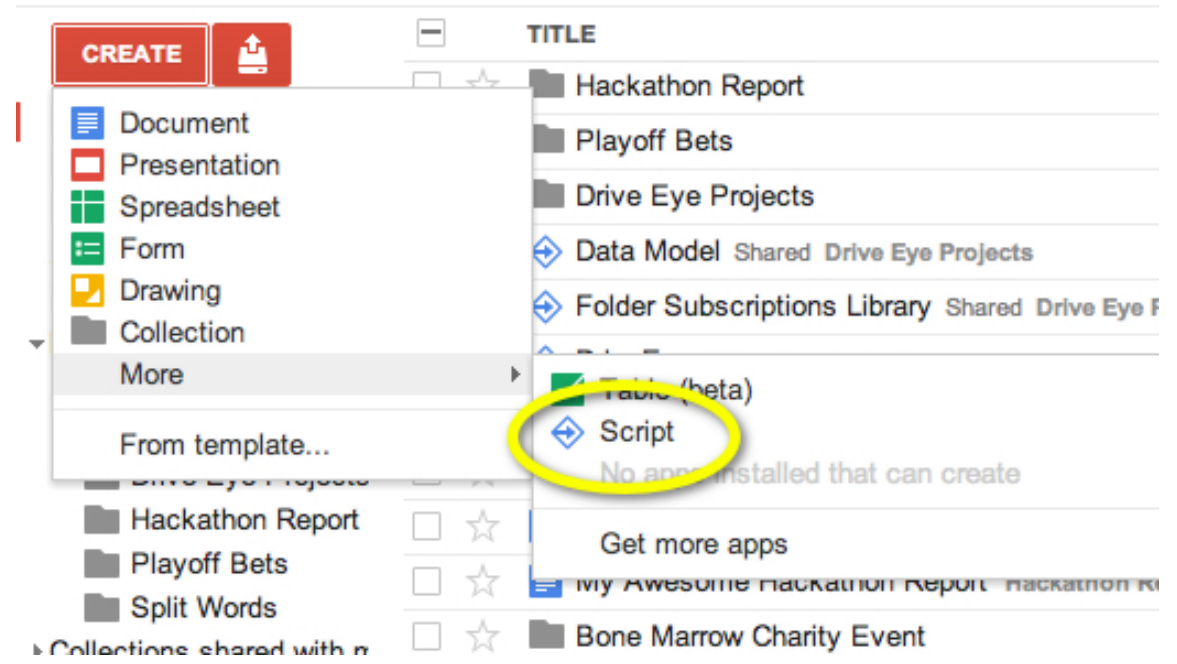
Create

Container-bound vs. Standalone Scripts

Container-bound



Standalone



Create

Standalone vs. Container-bound - Create scripts from script.google.com

The screenshot shows the Google Apps Script editor interface. The browser address bar displays the URL: <https://script.google.com/macros/d/MCtDMDwZxQ2WbMcAJARIYiz09B0xsIRCT/edit?splash=yes>. The main editor area shows an 'Untitled project' with a menu bar (File, Edit, View, Run, Publish, Resources, Help) and a toolbar. The code editor displays a simple JavaScript function:

```
1 function myFunction() {  
2  
3 }  
4
```

. A 'Welcome screen' dialog box is overlaid on the editor, titled 'Google Apps Script'. It contains two columns: 'Create script for' and 'Tutorials'. The 'Create script for' column lists options: Blank Project, Script as Web App, Drive, Spreadsheet, Document, GMail, Calendar, and Sites. The 'Tutorials' column lists: Building Your First Script, Custom Functions in Spreadsheets, Creating Gmail Inbox Statistics Reports, Embedding Apps Script in Google Sites, and Time Booking Application. Below these columns is the section 'Open a recent project' with the text 'No projects found'. At the bottom of the dialog, there is a checkbox for 'Don't show again' and a 'Close' button. An orange arrow points from a text box labeled 'Welcome screen' to the dialog box.

Untitled project

File Edit View Run Publish Resources Help

Untitled project

```
1 function myFunction() {  
2  
3 }  
4
```

Code.gs

Code.gs

Google Apps Script

Create script for

- Blank Project
- Script as Web App
- Drive
- Spreadsheet
- Document
- GMail
- Calendar
- Sites

Tutorials

- Building Your First Script
- Custom Functions in Spreadsheets
- Creating Gmail Inbox Statistics Reports
- Embedding Apps Script in Google Sites
- Time Booking Application

Open a recent project

No projects found

Don't show again **Close**

Welcome screen





Create

Standalone vs. Container-bound Scripts

Container-bound

Follow lifecycle of the
container

Script gets deleted when parent
spreadsheet is trashed

Standalone

Independent lifecycle

Scripts get created, renamed,
trashed from Drive



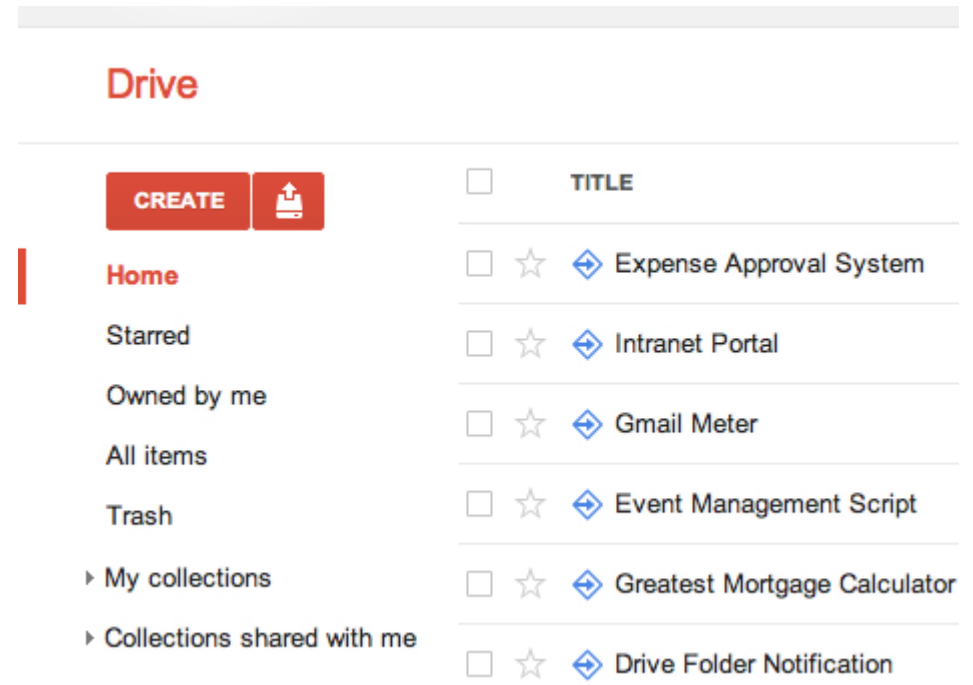
Create

Standalone vs. Container-bound - Finding Scripts

Container-bound

Locate the parent spreadsheet and then go to Script editor.

Standalone



The screenshot shows the Google Drive interface. At the top, the word "Drive" is displayed in red. Below it, there is a "CREATE" button in red and a "New" icon. A sidebar on the left contains navigation options: Home (selected), Starred, Owned by me, All items, Trash, My collections, and Collections shared with me. The main area displays a list of items with checkboxes, star icons, and blue diamond icons. The items listed are:

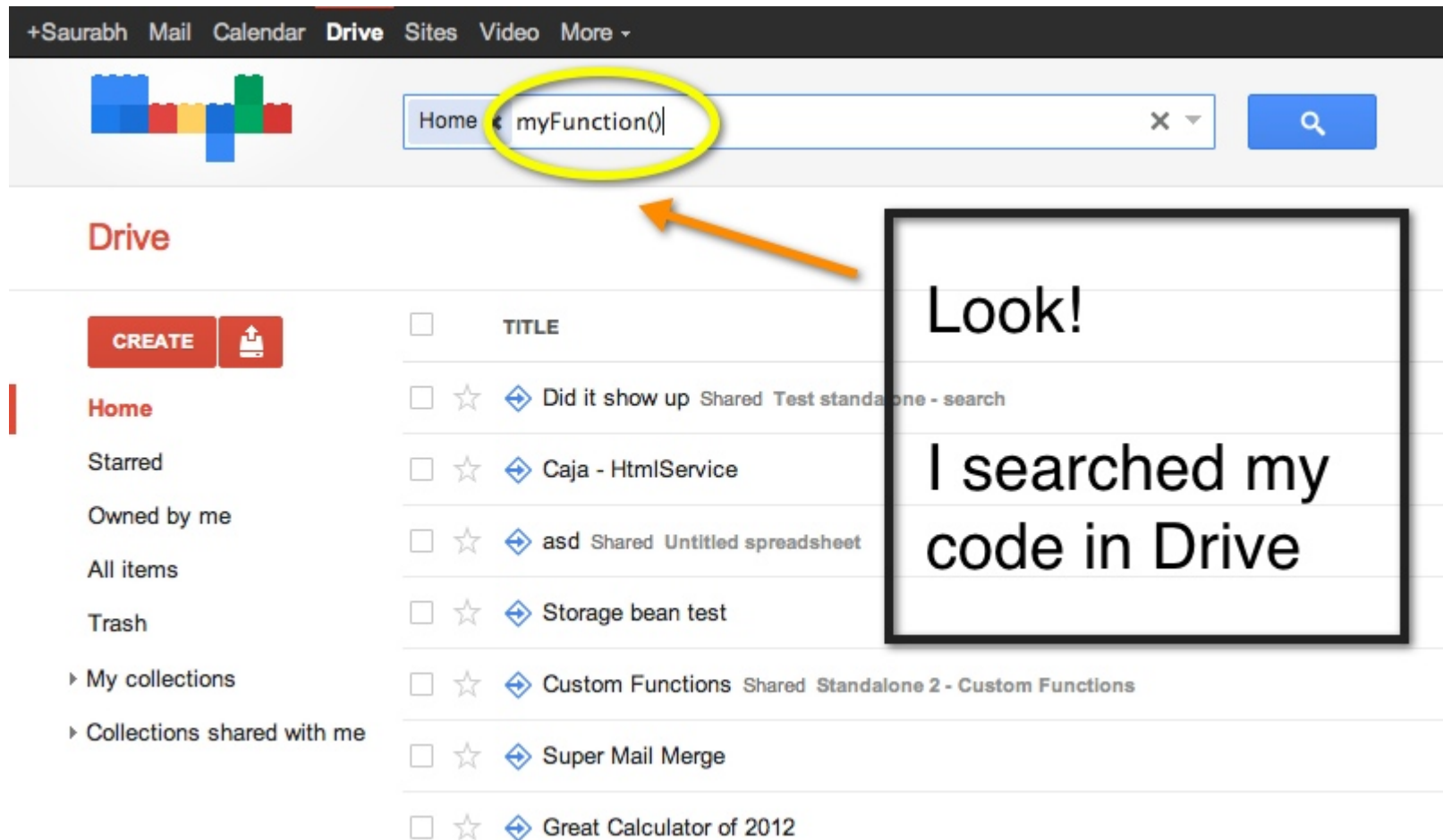
<input type="checkbox"/>	TITLE
<input type="checkbox"/> ☆	Expense Approval System
<input type="checkbox"/> ☆	Intranet Portal
<input type="checkbox"/> ☆	Gmail Meter
<input type="checkbox"/> ☆	Event Management Script
<input type="checkbox"/> ☆	Greatest Mortgage Calculator
<input type="checkbox"/> ☆	Drive Folder Notification



Create

Standalone vs. Container-bound - Search using Code Snippets

Standalone



The screenshot shows the Google Drive interface with a search bar containing the text "myFunction()". The search results list several files, including "Did it show up Shared Test standalone - search", "Caja - HtmlService", "asd Shared Untitled spreadsheet", "Storage bean test", "Custom Functions Shared Standalone 2 - Custom Functions", "Super Mail Merge", and "Great Calculator of 2012". A yellow circle highlights the search bar, and an orange arrow points from a text box to it. The text box contains the text "Look! I searched my code in Drive".

Home myFunction()

Drive

CREATE

Home

Starred

Owned by me

All items

Trash

My collections

Collections shared with me

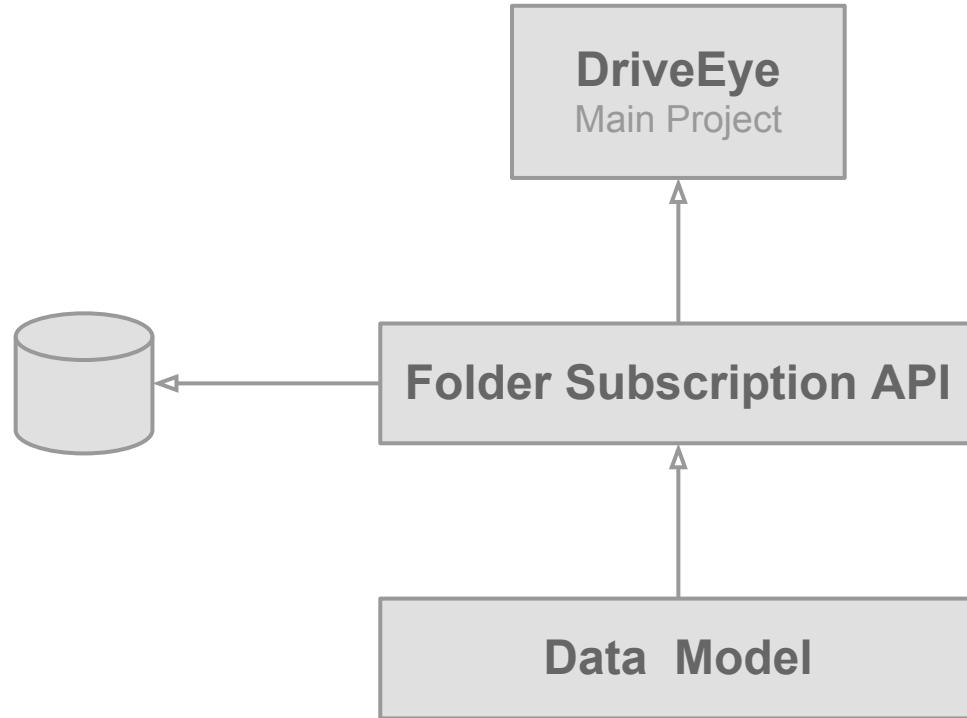
TITLE
Did it show up Shared Test standalone - search
Caja - HtmlService
asd Shared Untitled spreadsheet
Storage bean test
Custom Functions Shared Standalone 2 - Custom Functions
Super Mail Merge
Great Calculator of 2012

Look!
I searched my code in Drive



Create

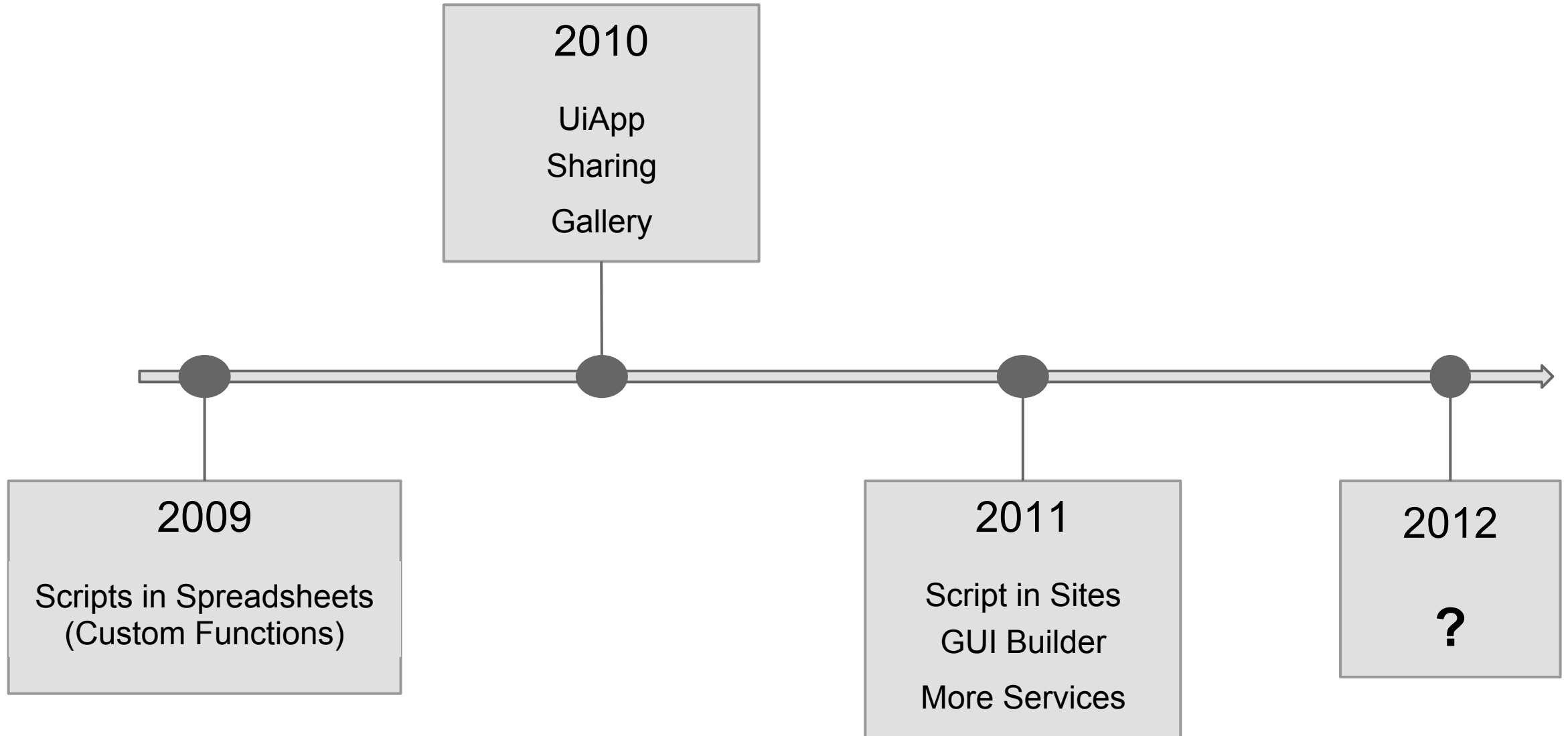
Designing DriveEye - 3 Standalone Projects





What's the best way to build UI and store data?

Apps Script over the years







Create

New Features of Google Apps Script

1

Create Standalone Scripts

2

ScriptDb

3

HtmlService





Create

To Learn more about ScriptDb

"Storing Data using Google Apps Script"

When: **June 28th, 2012**

Time: **17:15pm**

Where: **Room 7**

Who: **Drew Csillag**





Create

To Learn more about HtmlService and using client side JS in Apps Script

"Use what you know: Html and JavaScript in Apps Script"

When: June 28th, 2012

Time: 2:45pm

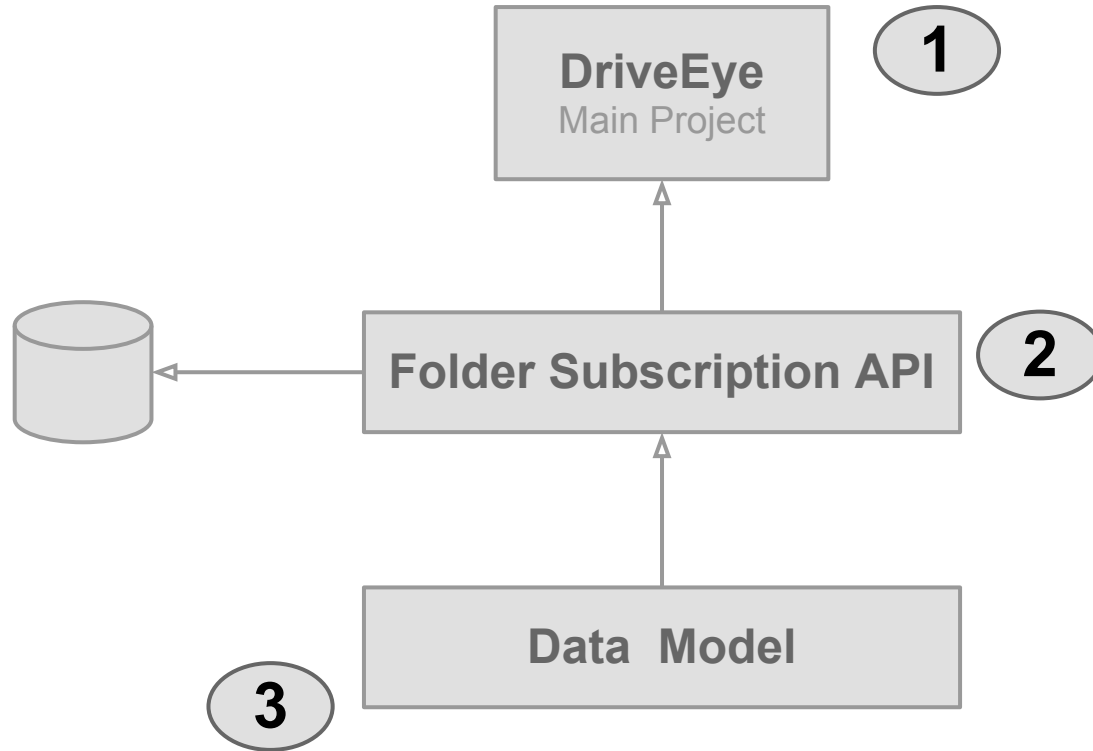
Where: Room 11

Who: Corey Goldfeder



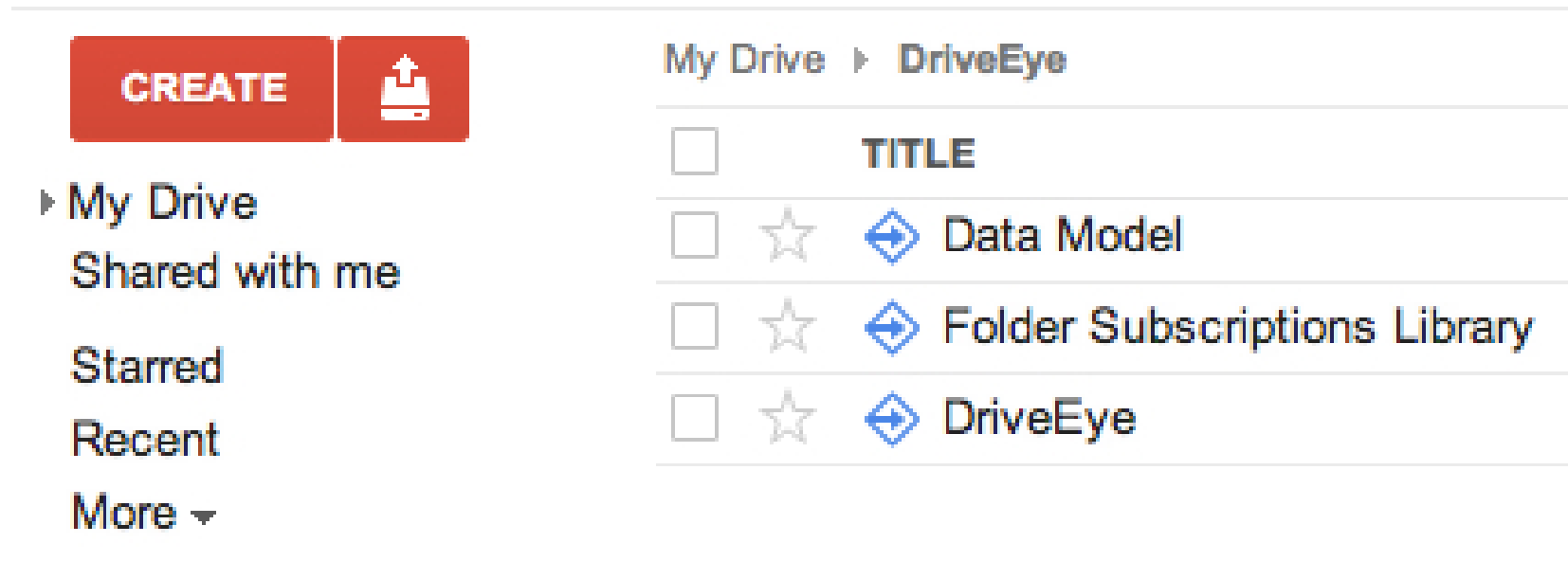
Create

Designing DriveEye - 3 Standalone Projects



Create

DriveEye - Create three standalone script projects



The screenshot shows the Google Drive interface. On the left, there is a sidebar with a red 'CREATE' button and a folder icon. Below it are navigation options: 'My Drive', 'Shared with me', 'Starred', 'Recent', and 'More'. The main area shows the 'My Drive' folder expanded to 'DriveEye'. A table lists the contents of the 'DriveEye' folder:

<input type="checkbox"/>		TITLE
<input type="checkbox"/>	☆	➡ Data Model
<input type="checkbox"/>	☆	➡ Folder Subscriptions Library
<input type="checkbox"/>	☆	➡ DriveEye



Create

Project 1 - Data model to manage folder subscriptions

```
function User(userId, driveFolders, triggerId) {
```

APPS SCRIPT

```
  this.userId = userId;
```

```
  this.driveFolders = driveFolders;
```

```
  this.triggerId = triggerId;
```

```
}
```

```
function DriveFolder(folderId, folderName, files) {
```

```
  this.folderId = folderId;
```

```
  this.folderName = folderName;
```

```
  // string array of files id
```

```
  this.files = files;
```

```
}
```



Create

Project 2 - API to manage folder subscriptions

APPS SCRIPT

```
function addFolderSubscription(folderName)
function getSubscribedFolders()
function removeFolderSubscription(folderName)
function removeUser()
function getUser()
```





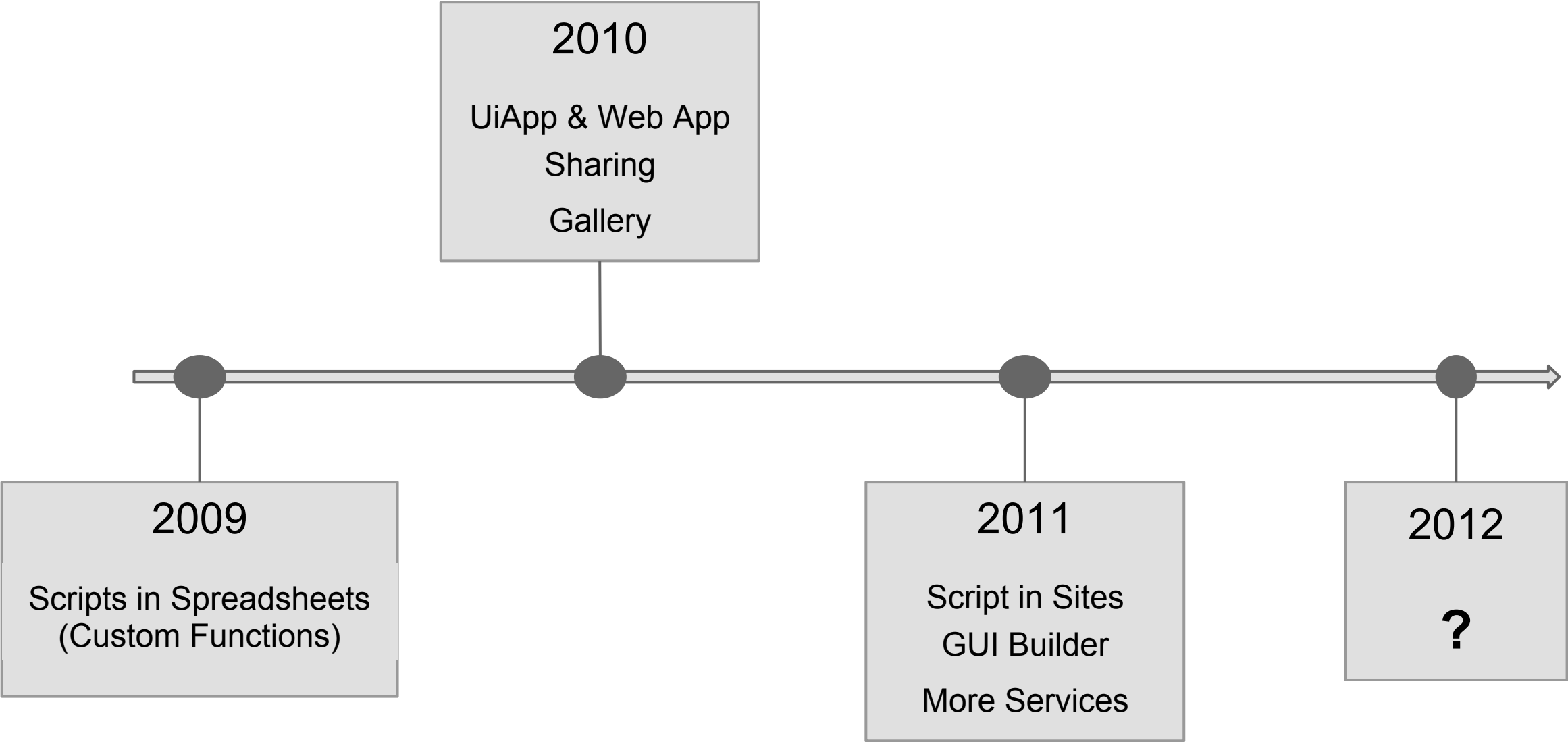
How to glue standalone projects?



Share scripts to
collaborate and organize
your code



Apps Script over the years





Create

Share

New Features of Google Apps Script

1

Create Standalone Scripts

2

ScriptDb

3

HtmlService

4

Share scripts in a simple way





Simple Sharing

- 1 Google Docs way to share scripts
- 2 Container-bound scripts inherit permissions from the parent



Create

Share

Understanding Permissions

Share Scripts using "Share" button

Two types of permissions
View, Edit

All Viewers get Execute permissions



Drive Folder Notification

File Edit View Run Publish Resources Help

Drive Folder Notification

Code.gs

Sharing settings

Link to share (only accessible by collaborators)

<https://script.sandbox.google.com/d/1hEVD2rXXcXjG9yZb6NEGN0fQ23cq8LoDQYT>

Who has access

	Private - Only the people listed below can access	Change...
	Saurabh Gupta (you) saurabh@gluemesh...	Is owner

Add people:

Editors will be allowed to add people and change the permissions. [\[Change\]](#)

[Done](#)

“How does sharing help me organize my code?”

- Logical User





good ol'

Libraries





3 steps to create script libraries

- 1 Save a Version of your script
- 2 Share script with view permissions
- 3 Share project key





DriveEye

- Convert two projects into library
 - Data model library
 - Folder subscription library
- Include the two libraries into the main UI script



Create

Share

DriveEye - Create Two Libraries

CREATE



▶ My Drive

Shared with me

Starred

Recent

More ▼

My Drive ▶ DriveEye



TITLE



Data Model



Folder Subscriptions Library



DriveEye





4 steps for using a script library



Insert Library Project Key



Find a Library

Enter the library's project key (found under File > Project Properties).
Make sure you have permissions to access the library or its containing spreadsheet.

Included Libraries

Title	Version	Identifier	Development Mode	Remove
Folder Alert Library	...	FolderAlertLibrary	off	<input type="button" value="x"/>





4 steps for using a script library



Select a version



Included Libraries

Title	Version	Identifier	Development Mode	Remove
Folder Alert Library	...		off <input type="checkbox"/>	<input type="checkbox"/>

Static Versions

- 4
- 3
- 2
- 1

Find a Library

Enter the library's project key (found under File > Project Properties).
Make sure you have permissions to access the library or its containing spreadsheet.





4 steps for using a script library



Select an identifier

Identifier becomes the top level class name for that library

Included Libraries

Title	Version	Identifier	Development Mode	Remove
Folder Alert Library	3	FolderAlertLib	off	✕

Find a Library

Select

Enter the library's project key (found under File > Project Properties).
Make sure you have permissions to access the library or its containing spreadsheet.

Save

Cancel





4 steps for using a script library



Use Library with identifier

Autocomplete shows all the functions that are available in the library

```
1 function doGet() {
2   var app = UiApp.createApplication();
3
4
5   // using library
6   FolderAlertLib.
7   return app;
8 }
9
10 function myClick
11   var app = UiAp
12
13   var label = ap
14   label.setVisib
15
16   app.close();
17   return app;
--
```

addFolderSubscription(Object folderName) : void
getSubscribedFolders() : void
getUser() : void
hasNewFiles() : void
removeFolderSubscription(Object folderName) : void
removeUser() : void



Create

Share

Using data model library

```
function getNewFolderFromId_(folderId) {  
  var folder = DocsList.getFolderById(folderId);  
  var folderName = folder.getName();  
  // ...  
  var driveFolder =  
    new DataModel.DriveFolder(folderId, folderName, fileIds);  
  return driveFolder;  
}
```

APPS SCRIPT



Create

Share

Using folder subscription api

```
function getFolders() {  
    return FolderAlertLib.getSubscribedFolders();  
}  
  
function createFolder(folderName) {  
    return FolderAlertLib.addFolderSubscription(folderName);  
}  
  
function removeFolder(folderName) {  
    return FolderAlertLib.removeFolderSubscription(folderName);  
}
```

APPS SCRIPT





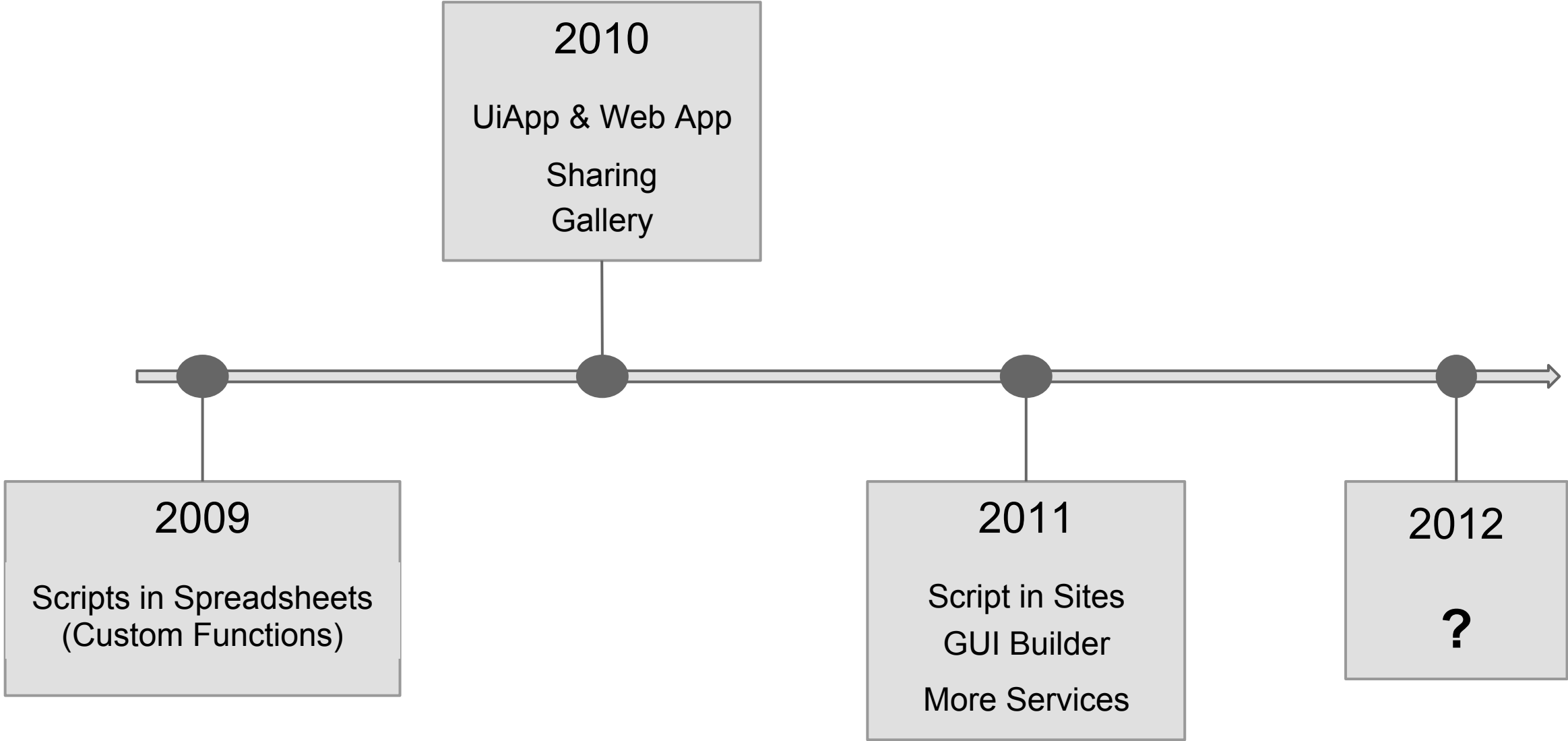
What can I do with scripts?



Deploy scripts as web apps



Apps Script over the years





Understanding Web Apps



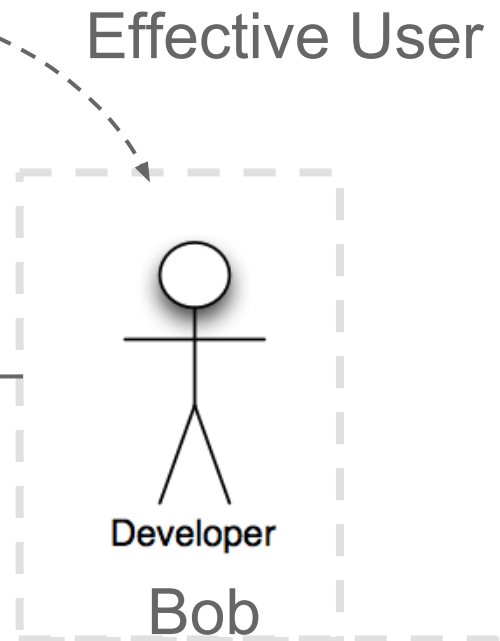
Uses

Web App

```
function doGet() {  
  var app = UiApp.crea  
  // ... more code  
  // ... more ui code  
  return app;  
}
```

APPS SCRIPT

Writes



Access



Active User
(User on the keyboard)







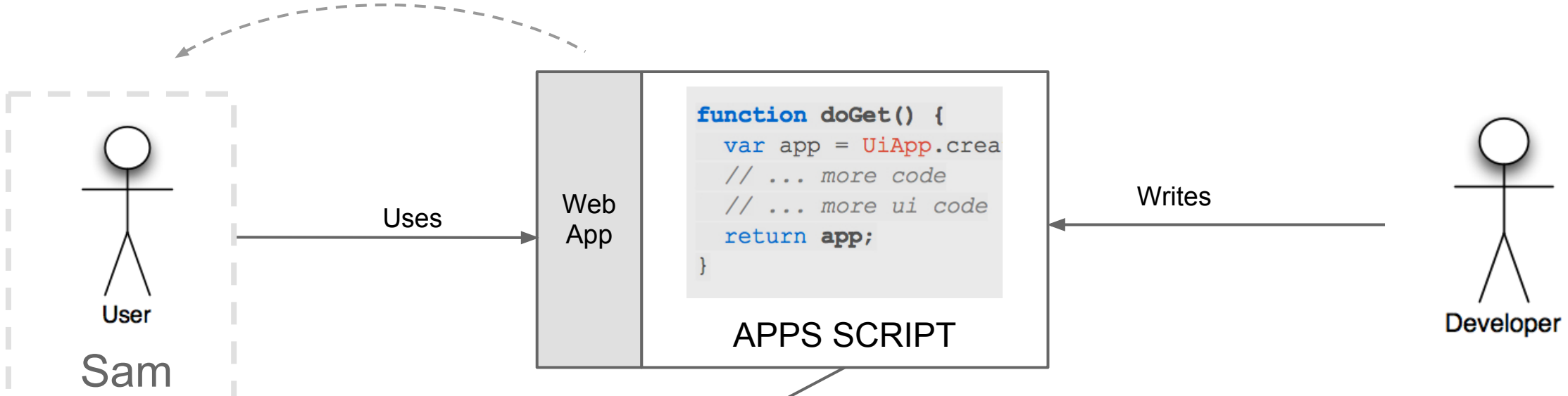
New Features of Google Apps Script

- 1 Create Standalone Scripts
- 2 ScriptDb
- 3 HtmlService
- 4 Share scripts in a simple way
- 5 Deploy new type of web apps**





Scripts run as the end user



Active User == Effective User



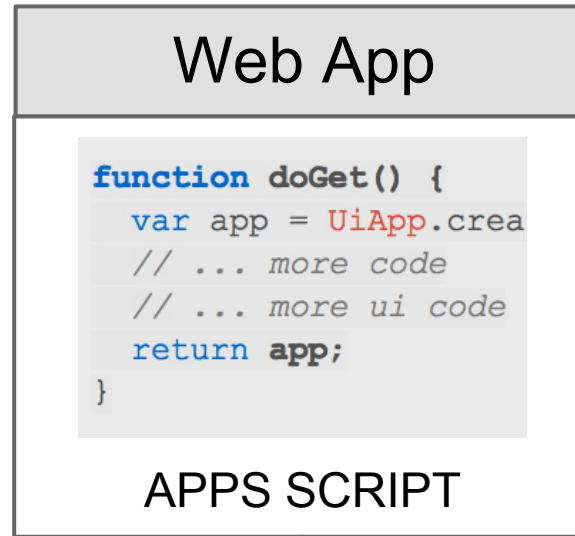
Advantages of scripts running as active user

- Scripts asks user to grant access to resources
- Scripts identifies the end-user
- Script uses end-user's quota
 - Scripts have execution quota. Quota is tied to the account which executes the script.





Two types of Web Apps



Active User is the end user
Effective User is the developer

Active User and Effective User is
the same end-user





How to Deploy a web app

Select versions

Select type of web app

Access Control

Drive Folder Notification
File Edit View Run Publish Resources Help

Drive Folder Notification
Code.gs

Deploy as web app...

Project version:
1

Execute the app as:
User accessing the web app
Users will be asked to authorize the script.

Who has access to the app:
Only myself

Deploy Cancel Help





Versioning for Web Apps

- Immutable Url for web apps
- Change versions without affecting users
- Development Url for testing



Create

Share

Deploy

DriveEye - Writing a simple web app

APPS SCRIPT

```
function doGet(e) {  
  var html = HtmlService.createFromFile(...);  
  // ... more code  
  // ... more ui code  
  return htmlOutput;  
}
```



Create

Share

Deploy

DriveEye - Creating Triggers for each user

APPS SCRIPT

```
function addFolderSubscription(folderName) {  
    // ...  
    // create a new trigger  
    var triggerId =  
        ScriptApp.newTrigger("hasNewFiles")  
            .timeBased().everyMinutes(5)  
            .create().getUniqueId();  
    // ...  
}
```





How do users find your web app?

Create

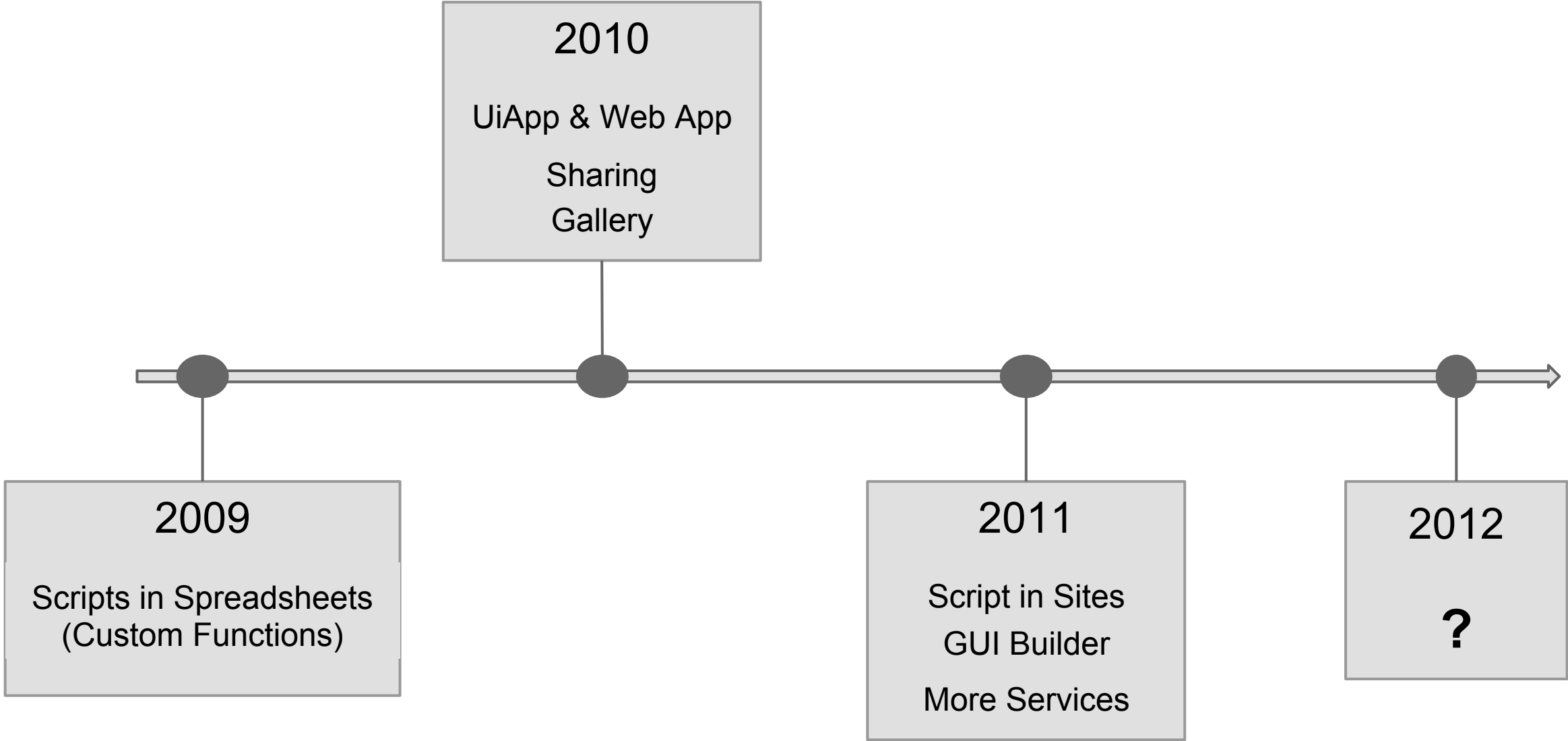
Share

Deploy

Publish



Apps Script over the years







New Features of Google Apps Script

- 1 Create Standalone Scripts
- 2 ScriptDb
- 3 HtmlService
- 4 Share scripts in a simple way
- 5 Deploy new type of web apps
- 6 Integration with Chrome Web Store**



Create

Share

Deploy

Publish



available in the
chrome web store

New!





What's so special about Chrome Web Store integration?

- One click uploads in Chrome Web Store
- Easy steps to verify ownership





Verify the ownership using Google Webmaster Central

- Copy the meta-tags from Webmaster Central
- Paste the meta-tags in "Register with Chrome Web Store dialog"



Create

Share

Deploy

Publish

1

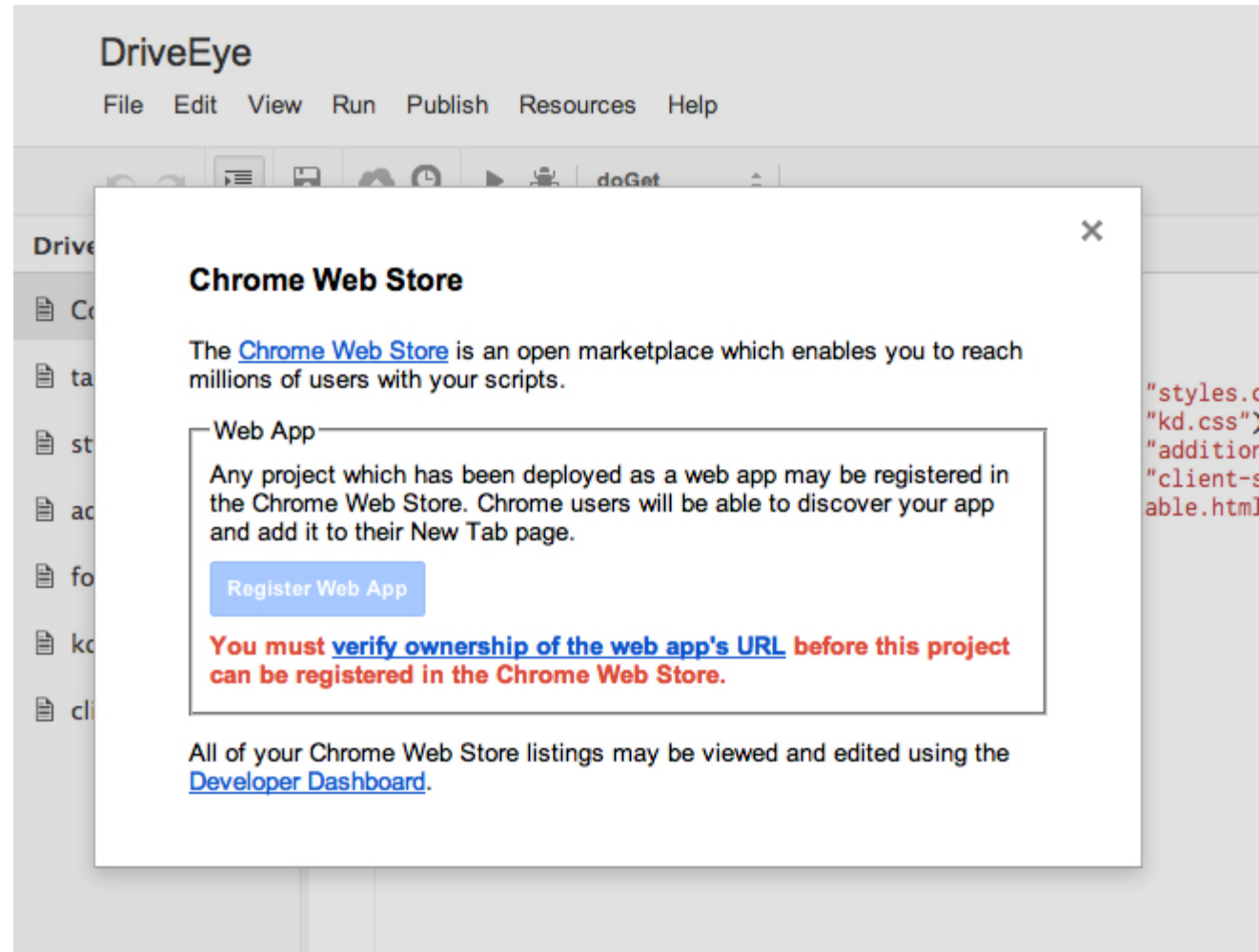
2

3

4

Register the Web App in Chrome Web Store

- Required to register as a developer on Chrome Web Store
- From Script editor, click **Publish > Register with Chrome Web Store**
- Every application in Chrome Web Store needs a manifest
- Visit Developer Dashboard in Chrome Web Store to view the uploaded item



The screenshot shows the DriveEye application interface. At the top, there is a menu bar with options: File, Edit, View, Run, Publish, Resources, and Help. Below the menu bar is a toolbar with various icons. A dialog box titled "Chrome Web Store" is open in the foreground. The dialog box contains the following text:

Chrome Web Store

The [Chrome Web Store](#) is an open marketplace which enables you to reach millions of users with your scripts.

Web App

Any project which has been deployed as a web app may be registered in the Chrome Web Store. Chrome users will be able to discover your app and add it to their New Tab page.

[Register Web App](#)

You must [verify ownership of the web app's URL](#) before this project can be registered in the Chrome Web Store.

All of your Chrome Web Store listings may be viewed and edited using the [Developer Dashboard](#).





Customize Listing Chrome Web Store

- Standard Chrome Web Store requirements
- Icons, Screenshots and Display Images
- Good quality assets == Better rank in Chrome Web Store





Publish and go live !



Completing the story

DriveEye

Shared Folder Notifications





Demo of DriveEye

Create

Share

Deploy

Publish

New Features of Google Apps Script

- 1 Create Standalone Scripts
- 2 ScriptDb
- 3 HtmlService
- 4 Share scripts in a simple way
- 5 Deploy new type of web apps
- 6 Integration with Chrome Web Store



script.google.com



#io12

+[Saurabh Gupta](#)

script.google.com



#io12



Google
Developers