





Storing Data in Apps Script

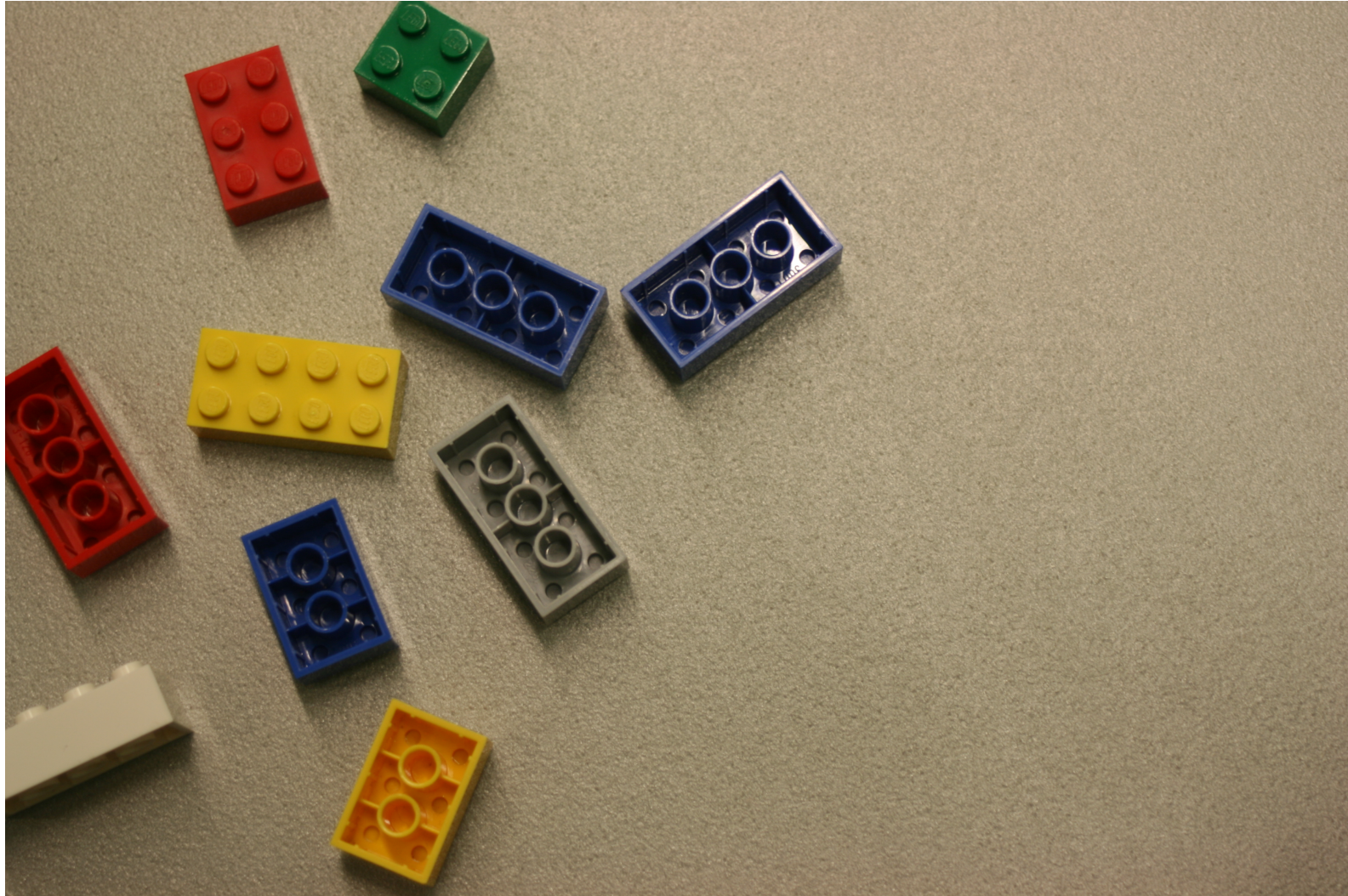
How And Where Do I Put Stuff In Script?

Drew Csillag

Manager and Staff Software Engineer, Google Apps Script

Lego™!







ScriptProperties

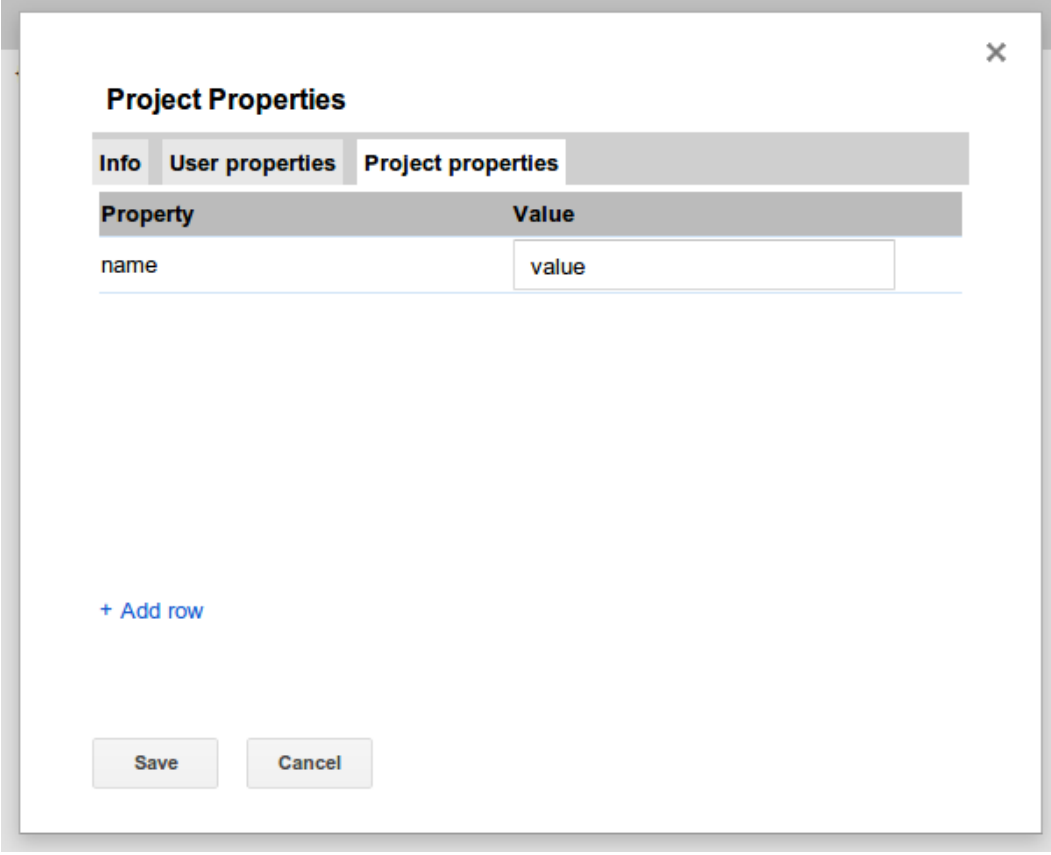
ScriptProperties API

```
var value = ScriptProperties.getProperty(name);
```

ScriptProperties API

```
var value = ScriptProperties.getProperty(name);  
ScriptProperties.setProperty(name, value);
```

Project Properties Editor



The screenshot shows a dialog box titled "Project Properties" with a close button (X) in the top right corner. Below the title, there are three tabs: "Info", "User properties", and "Project properties", with "Project properties" being the active tab. Below the tabs is a table with two columns: "Property" and "Value". The first row contains the text "name" in the "Property" column and "value" in the "Value" column. Below the table is a blue link that says "+ Add row". At the bottom of the dialog, there are two buttons: "Save" and "Cancel".

Property	Value
name	value

[+ Add row](#)

Script Properties

- Keys are `width x height`
- Values are the number of bricks of that type

Key	Value
1x2	56
1x3	88
1x4	92
2x2	109
2x4	62

ScriptProperties API

```
var value = parseFloat(ScriptProperties.getProperty("2x4"));
```

ScriptProperties Advantages

- Dead simple API
- Built in UI
- Zero setup
- Best for settings related to a script

What Not To Use ScriptProperties For

- Data that changes really often (more than once per second)
- Large amounts of data
 - limit is 9k per property
 - total of 512k for all keys and values
- Per-user preferences
- A generalized database
- Structured values like lists and tabular data

ScriptProperties And Structured Data

- Key: 2x4-Blue
- Value: 29

ScriptProperties And Structured Data

- Key: 2x4
- Value: Blue,29,Red,13,Green,23



Spreadsheets

Spreadsheets

Can be used for all the things that script properties can be used for, but spreadsheets are especially good for:

- Semi-structured data like lists and tables
- Has a high functioning built-in UI -- the spreadsheet editor
- Can use Google Forms to do data entry
- It's easy to copy and prepopulate, import and export

Spreadsheet Inventory

fx	Brick							Show all formulas	X
	A	B	C	D	E	F	G	H	
1	Brick	Blue	White	Green	Yellow	Orange	Red	Total By Brick	
2	1x2	100	54	76	89	82	71	472	
3	1x3	56	96	99	56	62	90	459	
4	1x4	86	78	88	69	71	60	452	
5	1x6	99	73	94	84	62	77	489	
6	1x8	80	79	53	85	64	71	432	
7	2x2	84	52	70	66	90	83	445	
8	2x3	66	70	59	74	100	57	426	
9	2x4	54	56	50	52	80	75	367	
10	Total By Color	625	558	589	575	611	584		
11									

Sample Code

APPS SCRIPT

```
function getBrickRow(desc) {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("Bricks");  
  var data = sheet.getRange(2, 1, sheet.getMaxRows(), 2).getValues();  
  for (var i = 0 ; i < data.length; i++) {  
    if (data[i][0] == desc) {  
      return i+2;  
    }  
  }  
  throw "brick type " + desc + " not found!";  
}
```

Sample Code

APPS SCRIPT

```
function getBrickRow(desc) {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("Bricks");  
  var data = sheet.getRange(2, 1, sheet.getMaxRows(), 2).getValues();  
  for (var i = 0 ; i < data.length; i++) {  
    if (data[i][0] == desc) {  
      return i+2;  
    }  
  }  
  throw "brick type " + desc + " not found!";  
}
```

Sample Code

APPS SCRIPT

```
function getBrickRow(desc) {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("Bricks");  
  var data = sheet.getRange(2, 1, sheet.getMaxRows(), 2).getValues();  
  for (var i = 0 ; i < data.length; i++) {  
    if (data[i][0] == desc) {  
      return i+2;  
    }  
  }  
  throw "brick type " + desc + " not found!";  
}
```

Sample Code

APPS SCRIPT

```
function getBrickRow(desc) {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("Bricks");  
  var data = sheet.getRange(2, 1, sheet.getMaxRows(), 2).getValues();  
  for (var i = 0 ; i < data.length; i++) {  
    if (data[i][0] == desc) {  
      return i+2;  
    }  
  }  
  throw "brick type " + desc + " not found!";  
}
```

Sample Code

APPS SCRIPT

```
function getBrickRow(desc) {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("Bricks");  
  var data = sheet.getRange(2, 1, sheet.getMaxRows(), 2).getValues();  
  for (var i = 0 ; i < data.length; i++) {  
    if (data[i][0] == desc) {  
      return i+2;  
    }  
  }  
  throw "brick type " + desc + " not found!";  
}
```

Sample Code

APPS SCRIPT

```
function getBrickRow(desc) {  
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("Bricks");  
  var data = sheet.getRange(2, 1, sheet.getMaxRows(), 2).getValues();  
  for (var i = 0 ; i < data.length; i++) {  
    if (data[i][0] == desc) {  
      return i+2;  
    }  
  }  
  throw "brick type " + desc + " not found!";  
}
```

Sample Code

APPS SCRIPT

```
function getBricksByColor(color, desc)
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("Bricks");
  return sheet.getRange(getBrickRow(desc), getColorColumn(color)).getValue();
}
```


Questions Answered

All Blue Bricks:

```
getBricksByColor("Blue", "Total By Color");
```

APPS SCRIPT

All 2x2 Bricks:

```
getBricksByColor("Total By Brick", "2x2");
```

APPS SCRIPT

All Blue 2x2 Bricks:

```
getBricksByColor("Blue", "2x2");
```

APPS SCRIPT

Spreadsheet Advantages/Disadvantages

- Advantages
 - built in UI
 - can handle tables and lists easily
- Disadvantages
 - slower than script properties
 - more complex API
 - no built in search
- Limits
 - 50k rows per sheet
 - 200 sheets



JDBC

Table Layout

Table bricks

SHORT_SIDE	LONG_SIDE	HEIGHT	COLOR	COUNT
1	2	3	Blue	75
1	2	3	White	89
1	2	3	Green	82

Example JDBC Usage

APPS SCRIPT

```
function getAllBlueBricks() {
  var c = Jdbc.getConnection(
    "jdbc:mysql://hostname:3306/dbname?user=username&password=password");
  var statement = c.createStatement();
  statement.execute("SELECT * FROM bricks WHERE color = 'Blue'");
  var results = statement.getResultSet();
  var result_data = [];
  while (results.next()) {
    var short_side = results.getObject("SHORT_SIDE");//1);
    var long_side = results.getObject("LONG_SIDE"); //2);
    var height = results.getObject("HEIGHT"); //3);
    var count = results.getObject("COUNT"); //5);
    result_data.push({short_side: short_side, long_side: long_side,
                      height: height, count: count});
  }
  return result_data;
}
```

Example JDBC Usage

APPS SCRIPT

```
function getAllBlueBricks() {  
  var c = Jdbc.getConnection(ScriptProperties.getProperty("JDBC_URL"));  
  var statement = c.createStatement();  
  statement.execute("SELECT * FROM bricks WHERE color = 'Blue'");  
  var results = statement.getResultSet();  
  var result_data = [];  
  while (results.next()) {  
    var short_side = results.getObject("SHORT_SIDE");//1);  
    var long_side = results.getObject("LONG_SIDE");//2);  
    var height = results.getObject("HEIGHT");//3);  
    var count = results.getObject("COUNT");//5);  
    result_data.push({short_side: short_side, long_side: long_side,  
                      height: height, count: count});  
  }  
  return result_data;  
}
```

Example JDBC Usage

APPS SCRIPT

```
function getAllBlueBricks() {  
  var c = Jdbc.getConnection(ScriptProperties.getProperty("JDBC_URL"));  
  var statement = c.createStatement();  
  statement.execute("SELECT * FROM bricks WHERE color = 'Blue'");  
  var results = statement.getResultSet();  
  var result_data = [];  
  while (results.next()) {  
    var short_side = results.getObject("SHORT_SIDE");//1);  
    var long_side = results.getObject("LONG_SIDE");//2);  
    var height = results.getObject("HEIGHT");//3);  
    var count = results.getObject("COUNT");//5);  
    result_data.push({short_side: short_side, long_side: long_side,  
                      height: height, count: count});  
  }  
  return result_data;  
}
```

Example JDBC Usage

APPS SCRIPT

```
function getAllBlueBricks() {  
  var c = Jdbc.getConnection(ScriptProperties.getProperty("JDBC_URL"));  
  var statement = c.createStatement();  
  statement.execute("SELECT * FROM bricks WHERE color = 'Blue'");  
  var results = statement.getResultSet();  
  var result_data = [];  
  while (results.next()) {  
    var short_side = results.getObject("SHORT_SIDE");//1);  
    var long_side = results.getObject("LONG_SIDE"); //2);  
    var height = results.getObject("HEIGHT"); //3);  
    var count = results.getObject("COUNT"); //5);  
    result_data.push({short_side: short_side, long_side: long_side,  
                      height: height, count: count});  
  }  
  return result_data;  
}
```


Example JDBC Usage

APPS SCRIPT

```
function getAllBlueBricks() {
  var result_data = [];
  var c = Jdbc.getConnection(ScriptProperties.getProperty("JDBC_URL"));
  var statement = c.createStatement();
  statement.execute("SELECT * FROM bricks WHERE color = 'Blue'");
  var results = statement.getResultSet();
  while (results.next()) {
    var short_side = results.getObject("SHORT_SIDE");//1);
    var long_side = results.getObject("LONG_SIDE");//2);
    var height = results.getObject("HEIGHT");//3);
    var count = results.getObject("COUNT");//5);

    result_data.push({short_side: short_side, long_side: long_side,
                      height: height, count: count});
  }
  return result_data;
}
```

Example JDBC Usage

APPS SCRIPT

```
function getAllBlueBricks() {
  var result_data = [];
  var c = Jdbc.getConnection(ScriptProperties.getProperty("JDBC_URL"));
  var statement = c.createStatement();
  statement.execute("SELECT * FROM bricks WHERE color = 'Blue'");
  var results = statement.getResultSet();
  while (results.next()) {
    var short_side = results.getObject(1);
    var long_side = results.getObject(2);
    var height = results.getObject(3);
    var count = results.getObject(5);

    result_data.push({short_side: short_side, long_side: long_side,
                      height: height, count: count});
  }
  return result_data;
}
```

Example JDBC Usage

APPS SCRIPT

```
function getAllBlueBricks() {
  var result_data = [];
  var c = Jdbc.getConnection(ScriptProperties.getProperty("JDBC_URL"));
  var statement = c.createStatement();
  statement.execute("SELECT short_side, long_side, height, count FROM bricks WHERE color = 'Blu");
  var results = statement.getResultSet();
  while (results.next()) {
    var short_side = results.getObject("SHORT_SIDE");//1);
    var long_side = results.getObject("LONG_SIDE"); //2);
    var height = results.getObject("HEIGHT"); //3);
    var count = results.getObject("COUNT"); //4);
    result_data.push({short_side: short_side, long_side: long_side,
                      height: height, count: count});
  }
  return result_data;
}
```

Example JDBC Usage

APPS SCRIPT

```
function getAllBricksOfColor(color) {  
  var result_data = [];  
  var c = Jdbc.getConnection(ScriptProperties.getProperty("JDBC_URL"));  
  var statement = c.createStatement();  
  statement.execute("SELECT * FROM bricks WHERE color = '" + color + "'");  
  var results = statement.getResultSet();  
  while (results.next()) {  
    var short_side = results.getObject("SHORT_SIDE");//1);  
    var long_side = results.getObject("LONG_SIDE"); //2);  
    var height = results.getObject("HEIGHT"); //3);  
    var count = results.getObject("COUNT"); //4);  
    result_data.push({short_side: short_side, long_side: long_side,  
                      height: height, count: count});  
  }  
  return result_data;  
}
```

Example JDBC Usage

APPS SCRIPT

```
function getAllBricksOfColor(color) {  
  var result_data = [];  
  var c = Jdbc.getConnection(ScriptProperties.getProperty("JDBC_URL"));  
  var statement = c.createStatement();  
  statement.execute("SELECT * FROM bricks WHERE color = '" + color + "'");  
  var results = statement.getResultSet();  
  while (results.next()) {  
    var short_side = results.getObject("SHORT_SIDE");//1);  
    var long_side = results.getObject("LONG_SIDE"); //2);  
    var height = results.getObject("HEIGHT"); //3);  
    var count = results.getObject("COUNT"); //4);  
    result_data.push({short_side: short_side, long_side: long_side,  
                      height: height, count: count});  
  }  
  return result_data;  
}  
getAllBricksOfColor("who-cares"; DROP TABLE bricks; -"); // !!!
```

Evil SQL Injection

```
statement.execute("SELECT * FROM bricks WHERE color = '" + color + "'"); APPS SCRIPT
```

```
getAllBricksOfColor("who-cares"; DROP TABLE bricks; --"); // !!!
```

```
// SELECT * FROM bricks WHERE color = 'who-cares'; DROP TABLE bricks; --'
```

Example JDBC Usage

APPS SCRIPT

```
function getAllBricksOfColor(color) {
  var result_data = [];
  var c = Jdbc.getConnection(ScriptProperties.getProperty("JDBC_URL"));
  var statement = c.createStatement();
  statement.execute("SELECT short_side, long_side, height, count FROM bricks WHERE color = ?");
  statement.setString(1, color);
  var results = statement.getResultSet();
  while (results.next()) {
    var short_side = results.getObject("SHORT_SIDE");//1);
    var long_side = results.getObject("LONG_SIDE"); //2);
    var height = results.getObject("HEIGHT"); //3);
    var count = results.getObject("COUNT"); //4);
    result_data.push({short_side: short_side, long_side: long_side,
                      height: height, count: count});
  }
  return result_data;
}
```

JDBC Advantages

- SQL
- Standard reporting software
- Transactions, consistency, etc.

Limits & Disadvantages

- You have to host it somewhere and make it accessible
- Can be slow as the data has to get shoveled across the interwebz
- Be careful about SQL injection!
- Spend a lot of time marshalling things into and out of SQL -- tend to write DAOs
- Have to plan quite a bit up front [Schema Design]



And Now For Something Different

Introducing ScriptDb

- JavaScript object database
- Query by example

Introducing ScriptDb

Ready to use

- No hosting
- No setup
- No installation
- No configuration

Basic Concepts

```
var db = ScriptDb.getMyDb();
```

APPS SCRIPT

Basic Concepts

APPS SCRIPT

```
var db = ScriptDb.getMyDb();  
var part = {short_side: 2, long_side: 3, height 3, count: 52, color: "blue"};  
var saved = db.save(part);
```

Basic Concepts

```
var db = ScriptDb.getMyDb();  
var part = {short_side: 2, long_side: 3, height 3, count: 52, color: "blue"};  
var saved = db.save(part);  
var part_id = saved.getId();
```

APPS SCRIPT

Basic Concepts

```
var db = ScriptDb.getMyDb();  
var part = {short_side: 2, long_side: 3, height 3, count: 52, color: "blue"};  
var saved = db.save(part);  
saved.is_translucent = true;  
saved.count = 92;  
db.save(saved);
```

APPS SCRIPT

Basic Concepts

```
var db = ScriptDb.getMyDb();  
var part = db.load(part_id);
```

APPS SCRIPT

Basic Concepts

```
var db = ScriptDb.getMyDb();  
var parts = db.load([part_id1, part_id2, part_id3]);
```

APPS SCRIPT

Basic Concepts

```
var db = ScriptDb.getMyDb();  
db.remove(part_id);
```

APPS SCRIPT

Basic Concepts

```
var db = ScriptDb.getMyDb();  
db.remove(part);
```

APPS SCRIPT



Queries

Queries

```
var db = ScriptDb.getMyDb();
var results = db.query({short_side: 2});
while (result.hasNext()) {
  var part = result.next();
  // do something with part
}
```

APPS SCRIPT

Queries

APPS SCRIPT

```
var db = ScriptDb.getMyDb();
var results = db.query({short_side: 2});
while (result.hasNext()) {
  var part = result.next();
  // do something with part
}
```

Queries

```
var db = ScriptDb.getMyDb();
var results = db.query({short_side: 2});
while (result.hasNext()) {
  var part = result.next();
  // do something with part
}
```

APPS SCRIPT

Queries

APPS SCRIPT

```
var db = ScriptDb.getMyDb();
var results = db.query({short_side: 2});
while (result.hasNext()) {
  var part = result.next();
  // do something with part
}
```

Queries

```
var results = db.query({  
  short_side: 2,  
  long_side: 4});
```

APPS SCRIPT

Queries

```
var results = db.query({  
  color: "Green",  
  bin: {bin_number: 23, bag: 19}});
```

APPS SCRIPT

Queries

```
var results = db.query({  
  short_side: 2,  
  color: db.anyOf(['Green', 'Blue'])});
```

APPS SCRIPT

Queries

```
var results = db.query({  
  short_side: 2,  
  color: db.anyOf(['Green', 'Blue']),  
  long_side: db.not(3)});
```

APPS SCRIPT

Queries

```
{  
  minifig_type: "torso",  
  color: "Black",  
  decoration: "badge",  
  clothing: "suit"  
}
```

APPS SCRIPT

```
var results = db.query({  
  color: 'Black',  
  minifig_type: db.anyValue()});
```

APPS SCRIPT

More Query Options

```
var results = db.query(...)  
    .limit(20);
```

APPS SCRIPT

More Query Options

```
var PAGE_SIZE = 20;  
var results = db.query(...)  
    .sortBy("long_side", db.NUMERIC)  
    .limit(PAGE_SIZE);
```

APPS SCRIPT

LEXICAL

1

10

2

3

NUMERIC

1

2

3

10

More Query Options

```
var PAGE_SIZE = 20;
var results = db.query(...)
    .sortBy("long_side", db.NUMERIC)
    .startAt(page_number * PAGE_SIZE)
    .limit(PAGE_SIZE);
```

APPS SCRIPT

More Query Options

```
var PAGE_SIZE = 20;  
var results = db.query(...)  
    .sortBy("long_side", db.NUMERIC)  
    .paginate(page_number, PAGE_SIZE);
```

APPS SCRIPT

Type Equivalences in Queries

- 3.0 will be found on query with values 3.0, "3" or true
- "3" will also be found 3.0, "3" or true
- 0.0 will be found on query with values 0, "0" and false
- "0" will be found with values 0, "0" and true
- "foobar" can be found with values "foobar" and true
- "" can be found with values "" and false
- Arrays and Maps can be found with true

Type Equivalences in Queries

So an item saved like this:

```
db.save({short_side: "2", ...});
```

APPS SCRIPT

Can be queried and be found like this:

```
db.query({short_side: 2}); // will match 2.0 or "2"
```

APPS SCRIPT

```
db.query({short_side: true}); // will match any true value, not just 2.0 or "2"
```

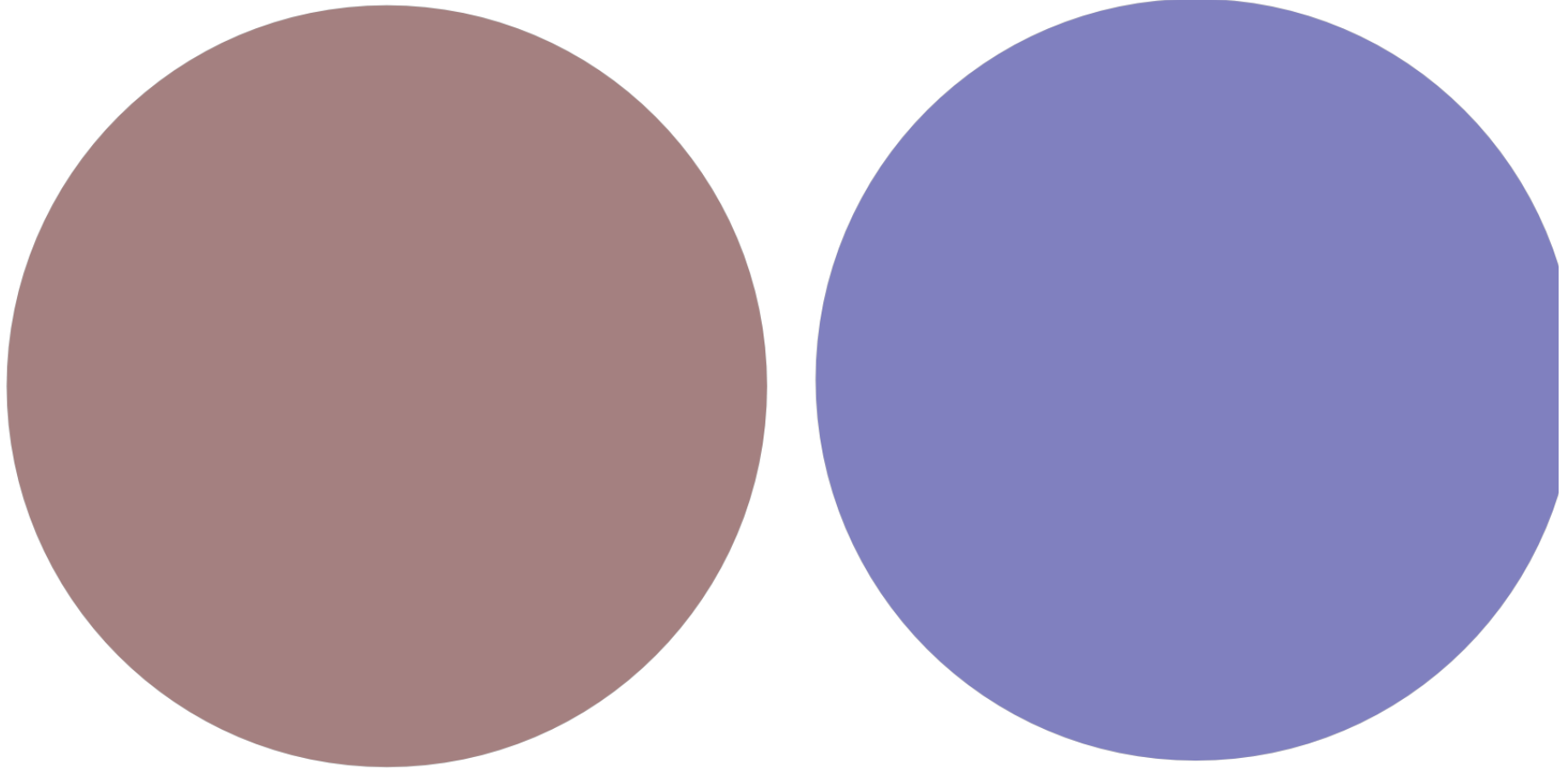
“Drew, this looks really cool, but where are the *!@!@*#!@ tables?”

Name withheld :)



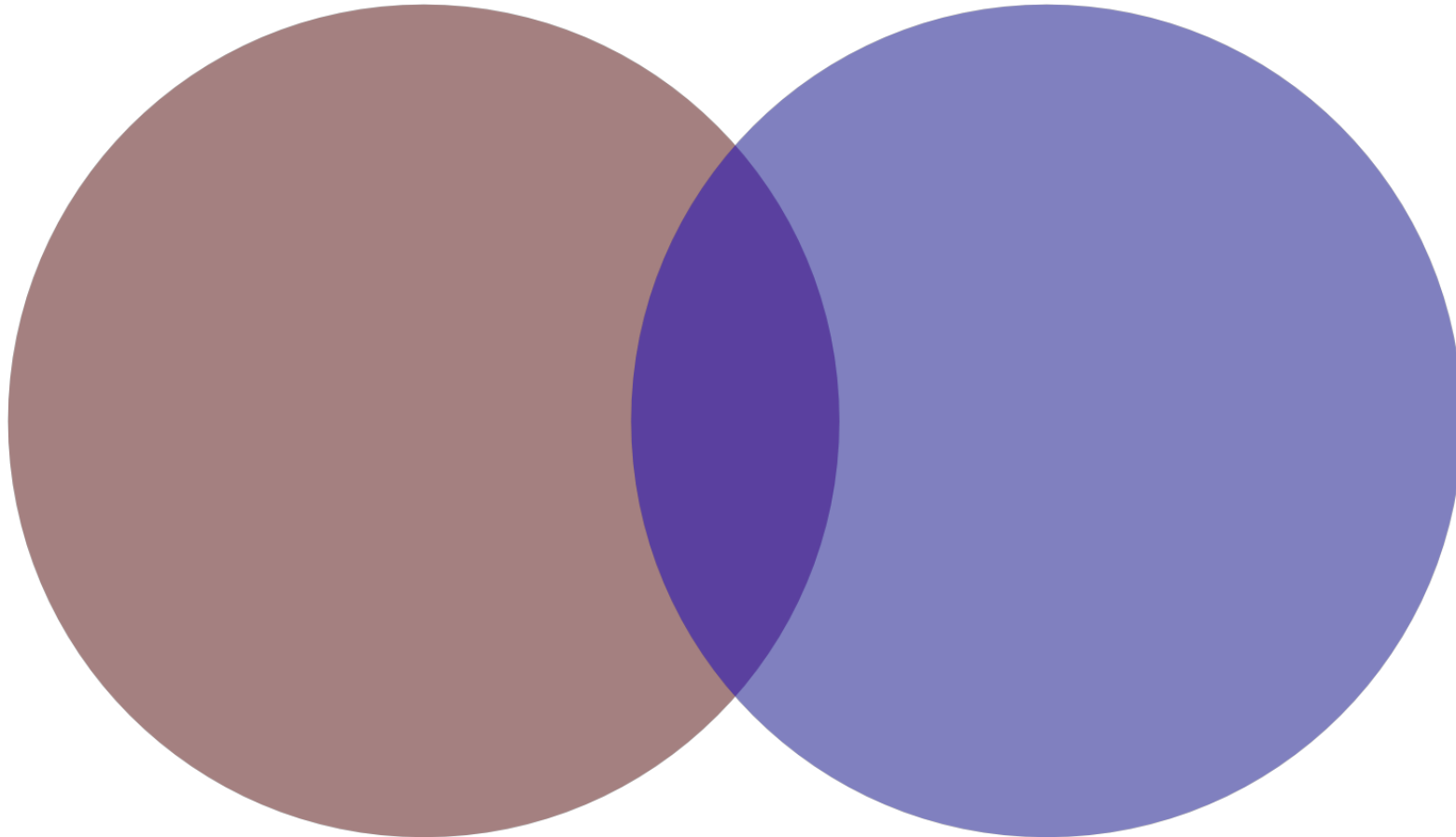
More about Data Silos

Disjoint Silos



More about Data Silos

Conjoined Silos



Where Are The Tables?

- Siloing by table-like attribute

```
db.query({table: "bricks", ...});
```

APPS SCRIPT

- Siloing by attribute presence

```
db.query({minifig_type: db.anyValue(), ...});
```

APPS SCRIPT

- Siloing by flag

```
db.query({is_translucent: true, ...});
```

APPS SCRIPT

Attribute Naming Consistency

Best Practice: Always use the same attribute name for the same kind of thing.

Mitigating Attribute Naming Inconsistency

Mitigating Attribute Naming Inconsistency

```
var result = db.query({"color,colour": "Black"});
```

APPS SCRIPT

Discovering Unplanned Silos

This is why consistent naming is important!



Joins

Joins

APPS SCRIPT

```
var result = db.query({
  short_side: 2,
  long_side: 2,
  height: 3,
  count: db.greaterThanOrEqualTo(50)});
var colors = {}
while (result.hasNext()) {
  colors[result.next().color] = 1;
}

result = db.query({
  minifig_type: "torso",
  color: db.anyOf(colors.keys())});
var result_colors = {};
while (result.hasNext()) {
  result_colors[result.next().color] = true;
}
```

Joins

APPS SCRIPT

```
var result = db.query({
  short_side: 2,
  long_side: 2,
  height: 3,
  count: db.greaterThanOrEqualTo(50)});
var colors = {}
while (result.hasNext()) {
  colors[result.next().color] = 1;
}

result = db.query({
  minifig_type: "torso",
  color: db.anyOf(colors.keys())});
var result_colors = {};
while (result.hasNext()) {
  result_colors[result.next().color] = true;
}
```

Limits

- Query result can be at most 50k rows
- Max database size governed by quota:
 - Consumer: 50MB
 - Google Apps: 100MB
 - Google Apps For Business: 200MB

Summary

- ScriptProperties
- Spreadsheets
- JDBC
- ScriptDb

<Thank You!>



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

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

Lego is a trademark of Lego A/S and Lego A/S was not affiliated with the talk

