




Apps Script Code Exercises

 Automate Your
Gmail with Google
Apps Script! 



Google Apps Script Mail Service Coding Exercises	1
Exercise 1: Send a Basic Email	2
Exercise 2: Send an Email with HTML Content	3
Exercise 3: Adding Attachments to an Email	3
Exercise 4: Fetching Emails from Inbox	4
Exercise 5: Searching Emails	5
Exercise 6: Labeling Emails	5
Exercise 7: Creating Email Filters	6
Exercise 8: Sending Emails with CC and BCC	6
Exercise 9: Auto-Reply to Emails	7
Exercise 10: Parsing Email Content	8

Google Apps Script Mail Service Coding Exercises

 10 innovative coding exercises to master Google Apps Script's Mail Service (GmailApp)! From sending emails with attachments to organizing your inbox with

Learn more about JavaScript with Examples and Source Code. Google Apps Script and Workspace Laurence Svekis Courses <https://basescripts.com/>

labels and filters, these exercises are designed to enhance your email automation skills.



Perfect for both beginners and advanced users, these exercises will guide you through the powerful capabilities of Gmail automation, improving your efficiency and productivity.



Elevate your email game and discover the endless possibilities with Google Apps Script!

Exercise 1: Send a Basic Email

Objective: Learn to send a basic email using GmailApp.

Explanation: This exercise introduces the essential function of sending an email. You will use `GmailApp.sendEmail` to send a simple text email.

Code:

```
function sendBasicEmail() {  
    var recipient = "example@example.com"; // Replace with actual recipient email  
    var subject = "Test Email";  
    var body = "Hello, this is a test email from Google Apps Script."  
    GmailApp.sendEmail(recipient, subject, body);  
}
```

Learn more about JavaScript with Examples and Source Code. Google Apps Script and Workspace Laurence Svekis Courses <https://basescripts.com/>

```
}
```

Exercise 2: Send an Email with HTML Content

Objective: Understand how to send an HTML-formatted email.

Explanation: This exercise focuses on sending emails containing HTML content for richer formatting.

Code:

```
function sendHtmlEmail() {  
    var recipient = "example@example.com"; // Replace with actual recipient email  
    var subject = "HTML Email";  
    var body = "<h1>Hello</h1><p>This is an HTML email.</p>";  
    GmailApp.sendEmail(recipient, subject, "", {htmlBody: body});  
}
```

Exercise 3: Adding Attachments to an Email

Objective: Learn to send an email with attachments.

Explanation: This exercise teaches how to add attachments to an email, a useful feature for sending documents and files.

Code:

```
function sendEmailWithAttachment() {
```

Learn more about JavaScript with Examples and Source Code. Google Apps Script and Workspace Laurence Svekis Courses <https://basescripts.com/>

```
var recipient = "example@example.com"; // Replace with actual recipient email
var subject = "Email with Attachment";
var body = "Please find the attached file.";
var file = DriveApp.getFileById('fileId'); // Replace with actual file ID
GmailApp.sendEmail(recipient, subject, body, {
  attachments: [file.getAs(MimeType.PDF)] // Change MimeType as required
});
}
```

Exercise 4: Fetching Emails from Inbox

Objective: Understand how to retrieve a list of recent emails from your inbox.

Explanation: This exercise focuses on using GmailApp to fetch and log emails from your inbox.

Code:

```
function fetchInboxEmails() {
  var threads = GmailApp.getInboxThreads();
  for (var i = 0; i < threads.length; i++) {
    Logger.log('Email Subject: ' + threads[i].getFirstMessageSubject());
  }
}
```

Learn more about JavaScript with Examples and Source Code. Google Apps Script and Workspace Laurence Svekis Courses <https://basescripts.com/>

Exercise 5: Searching Emails

Objective: Learn to search for specific emails using query strings.

Explanation: This exercise teaches how to use search queries to find specific emails in Gmail.

Code:

```
function searchEmails() {  
  var query = "from:example@example.com";  
  var threads = GmailApp.search(query);  
  for (var i = 0; i < threads.length; i++) {  
    Logger.log('Email Subject: ' + threads[i].getFirstMessageSubject());  
  }  
}
```

Exercise 6: Labeling Emails

Objective: Learn to programmatically apply labels to emails.

Explanation: This exercise demonstrates how to create and apply labels to organize emails in Gmail.

Code:

```
function labelEmails() {  
  var label = GmailApp.createLabel('My Label');
```

Learn more about JavaScript with Examples and Source Code. Google Apps Script and Workspace Laurence Svekis Courses <https://basescripts.com/>

```
var threads = GmailApp.getInboxThreads();
for (var i = 0; i < threads.length; i++) {
  threads[i].addLabel(label);
}
}
```

Exercise 7: Creating Email Filters

Objective: Understand how to create filters to automatically label emails.

Explanation: This exercise shows how to set up filters that automatically apply labels to incoming emails.

Code:

```
function createEmailFilter() {
  var label = GmailApp.createLabel('Filtered Label');
  var filter = GmailApp.createFilter("from:example@example.com", label);
}
```

Exercise 8: Sending Emails with CC and BCC

Objective: Learn to send emails with CC (carbon copy) and BCC (blind carbon copy).

Learn more about JavaScript with Examples and Source Code. Google Apps Script and Workspace Laurence Svekis Courses <https://basescripts.com/>

Explanation: This exercise teaches the importance of CC and BCC fields in email communication.

Code:

```
function sendEmailWithCcBcc() {  
    var recipient = "main@example.com"; // Replace with actual recipient email  
    var cc = "cc@example.com"; // Replace with actual CC email  
    var bcc = "bcc@example.com"; // Replace with actual BCC email  
    var subject = "Email with CC and BCC";  
    var body = "This email contains CC and BCC recipients.";  
    GmailApp.sendEmail(recipient, subject, body, {cc: cc, bcc: bcc});  
}
```

Exercise 9: Auto-Reply to Emails

Objective: Understand how to automatically reply to incoming emails.

Explanation: This exercise focuses on creating a script that sends an automatic reply to each email received.

Code:

```
function autoReplyEmails() {  
    var threads = GmailApp.getInboxThreads();  
    for (var i = 0; i < threads.length; i++) {  
        var messages = threads[i].getMessages();
```

Learn more about JavaScript with Examples and Source Code. Google Apps Script and Workspace Laurence Svekis Courses <https://basescripts.com/>

```
for (var j = 0; j < messages.length; j++) {  
  var message = messages[j];  
  message.reply("Thanks for your email. This is an automated response.");  
}  
}  
}
```

Exercise 10: Parsing Email Content

Objective: Learn to extract information from email messages.

Explanation: This exercise demonstrates how to read and process the content of an email.

Code:

```
function parseEmailContent() {  
  var threads = GmailApp.getInboxThreads();  
  for (var i = 0; i < threads.length; i++) {  
    var message = threads[i].getMessages()[0];  
    var content = message.getBody();  
    // Process the content of the email  
    Logger.log(content);  
  }  
}
```

Learn more about JavaScript with Examples and Source Code. Google Apps Script and Workspace Laurence Svekis Courses <https://basescripts.com/>

Learn more about JavaScript with Examples and Source Code. Google Apps Script and Workspace Laurence Svekis Courses <https://basescripts.com/>