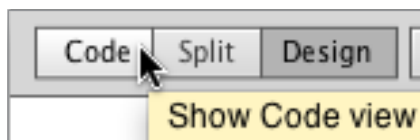**EXERCISE PREVIEW**

EXERCISE OVERVIEW

Some popular HTML email clients lack full CSS layout support, so we must use tables for consistent layout. In this exercise, we'll create a simple email using a single table, images, text and some basic CSS.

GETTING STARTED

1. In a text/code editor (such as Sublime Text, Dreamweaver, TextMate, Coda, etc.) go to **File > Open.**

2. Navigate to the **Desktop** and go into the **Class Files** folder, then **yourname-HTML Email Class** folder, then **Omniburger.**

3. Double–click on **birthday.html** to open it.

NOTE: If you are using Dreamweaver, in the upper left corner of the document, click the **Code** button to switch into Code view.

4. **Title** the document by editing the code as follows:

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Happy Birthday from Omniburger!</title>
</head>
<body>

</body>
</html>
```

5. Save the file.

   NOTE: The title won't show up in the email anywhere in most email clients. Outlook Mobile, however, displays the title in the text preview area. This page can also be put online, so it's best to add a title. To read more about how Outlook Mobile displays the title, go to: **tinyurl.com/hotmailtitle.** (Hotmail Mobile is now Outlook Mobile).

CODING UP THE TABLE

To see a finished design of the email you'll be coding, navigate to **Desktop > Class Files > yourname-HTML Email Class > Reference Files.**

1. Double–click on **birthday-coupon.pdf**. Notice the Happy Birthday heading, the simple message, the burger coupon image and the footer.

   While the fastest, easiest way to send this coupon out would be to simply pop a PNG or JPEG into an email and be done with the whole job, it's never a best practice to rely on one single image for an HTML email, for two important reasons:

   – A large percentage of email recipients do not enable images. If all your content is embedded in an image, your message is—essentially—never received. If you use HTML text for a portion of your message, the email can get your point across without images.
   – The balance of images to text is used as a gauge to predict whether an email is legitimate. A single-image email is more likely to be marked as spam than mixed content emails.

   As a best practice, let's create a simple one row, one column, table-based layout that includes some HTML text and a couple of images.

2. To create a table, add the following code (highlighted in bold) inside the <body> tag. In Sublime Text (with Emmet installed), you can simply type **table+** and hit **Tab** to expand to the full table code you see below.

```
<body>
    <table>
        <tr>
            <td></td>
        </tr>
    </table>
</body>
```

These tags are the basic building blocks of every table. Notice that there's a single row inside the <table> (indicated by the open and closed <tr> tags) and a single cell, or column, (indicated by the open and closed <td> tag) inside the row. The <td> stands for table data; all your content goes inside these tags. The row tags are purely structural.

3. Many people view their email through the narrow preview pane without ever opening up the email in a larger window. For best results across most email clients, the width of your HTML email should not exceed 700 pixels. Let's set a 600 pixel width for our email. Add the following width attribute to the <table> tag:

```
<table width="600">
    <tr>
        <td></td>
    </tr>
</table>
```

4. While we're here, let's remove all cellpadding (space inside the cells), cellspacing (space between the cells), and borders. Add the following attributes (in bold) to the <table> tag:

```
<table width="600" cellpadding="0" cellspacing="0" border="0">
```

5. Save your file.

6. Let's add some content to the table cell. Navigate to **Desktop > Class Files > yourname-HTML Email Class > Omniburger > snippets** and open **content-coupon.html.**

7. This is all the text for our page. **Select All** by pressing **Cmd–A** (MAC) or **Ctrl–A** (WINDOWS).

8. **Copy** it **(Cmd–C** (MAC) or **Ctrl–C** (WINDOWS)).

9. **Close** the file.

10. You should now be back in **birthday.html.**

11. **Paste (Cmd–V MAC or Ctrl–V WINDOWS)** the content inside the <td> tag like so:

```
<table width="600" cellpadding="0" cellspacing="0" border="0">
   <tr>
      <td>
         <h1>Happy Birthday!</h1>
         <p>The world is your burger. So have one with fries on us!</p>
      </td>
   </tr>
</table>
```

12. Let's next add the two images. First, the actual coupon. Add the following code (in bold) below the <p> tag inside the table cell, like so:

```
<td>
   <h1>Happy Birthday!</h1>
   <p>The world is your burger. So have one with fries on us!</p>
   <img src="img/free-burger.jpg" height="480" width="537" alt="Free Burger Coupon">
</td>
```

13. Save your file.

14. Let's add a footer image directly below the coupon image. The footer image has the company name, logo, address and store hours. Add the following code (in bold) to the table cell:

```
<img src="img/free-burger.jpg" height="480" width="537" alt="Free Burger Coupon">
<img src="img/footer.png" height="47" width="598" alt="Omniburger">
```

NOTE: We want this email to be 600px wide. Because we'll later be adding a 1px border around the entire content area, we had to make this footer image 598px wide. 598px + 1px left border + 1px right border = 600px total.

15. Save your file.

16. Let's see what we have so far. Launch your favorite **browser** (for example: **Safari** MAC, **Chrome**, **Firefox**, or **Internet Explorer** WINDOWS).

17. Hit **Cmd–O** MAC or **Ctrl–O** WINDOWS to open a file.

   – For browsers other than Internet Explorer: Navigate to the **Desktop** and go into the **Class Files** folder, then **yourname-HTML Email Class,** then **Omniburger** and find **birthday.html.** Hit **Open**.

   – If using **Internet Explorer**: Click **Browse**, then navigate to the file. Click **Open**. Then click **OK**.

18. Things look fine so far but it would be nice if the content were centered.

   TIP: Leave birthday.html open in the browser. This will make it easier for you to reload the page to see the changes you make in your code editor.

19. Return to **birthday.html** in your text/code editor to continue editing.

BROWSER PREVIEW SHORTCUTS

If you are using Sublime Text with **SideBarEnhancements** installed, you can hit **F12** (or **fn–F12** depending on your keyboard settings) and your saved HTML document will open in your default browser.

NOTE: This will typically not work on the Mac unless you change or disable the **Show Dashboard** keyboard shortcut in **System Preferences > Keyboard (or Mission Control).**

If you are using Dreamweaver, go to **File > Preview in Browser.**

20. Let's center the content inside the table cell. Add the following code to the
     <td> tag, like so:

```
<td align="center">
   <h1>Happy Birthday!</h1>
   <p>The world is your burger. So have one with fries on us!</p>
   <img src="img/free-burger.jpg" height="480" width="537" alt="Free Burger Coupon">
   <img src="img/footer.png" height="47" width="598" alt="Omniburger">
</td>
```

21. Save your file and preview the changes you made in a browser. The content is
     nicely centered inside the table cell but it'd be nice to center the entire table in
     the browser and/or the email preview pane. Return to your text/code editor.

22. Add the following attribute (in bold) to the <table> tag:

```
<table width="600" cellpadding="0" cellspacing="0" border="0" align="center">
```

23. Save your file and preview your page, once again, in a browser. Much better!

STYLING THE EMAIL

Sadly (and oddly), Google's Gmail neither supports external links to style sheets
nor embedded styles in the head or body of the document. This means that all
styles in HTML email must be inlined, element by element. Let's start by styling
the "Happy Birthday" heading.

1. Return to **birthday.html** in your text/code editor.

2. Add the following style attribute and property declaration (in bold) to the
    <h1> tag:

```
<h1 style="font-family:'Arial Black', Gadget, sans-serif;">Happy Birthday!</h1>
```

3. Save your file and preview your page in a browser. Nice. Let's make the text
    larger and orange, per the design specification.

4. Return to your text/code editor and add the following two new property
    declarations to the style attribute in the <h1> tag as follows:

```
<h1 style="font-family:'Arial Black', Gadget, sans-serif; font-size:58px;
color:#fb9a00;">
```

5. Save your file and preview your page in a browser. Looks good. Let's edit the
    paragraph underneath the heading.

6. Return to your text/code editor and add an inline style to the <p> tag
    as follows:

```
<p style="font-family:'Arial Black', Gadget, sans-serif; font-size:17px;">
```

7. Save your file and preview your page in a browser. The text closely resembles
    the text in the original design but the margins are way too big. Let's tighten
    them up.

1B
EXERCISE

8. Return to your text/code editor and edit the style attribute in the <h1> tag as follows:

```
<h1 style="font-family:'Arial Black', Gadget, sans-serif; font-size:58px;
color:#fb9a00; margin-top:15px; margin-bottom:0;">
```

9. Let's close up the space above the paragraph as well. Edit the style attribute in the <p> tag as follows:

```
<p style="font-family:'Arial Black', Gadget, sans-serif; font-size:17px;
margin-top:5px;">
```

10. Save your file and preview your page in a browser. Much improved!

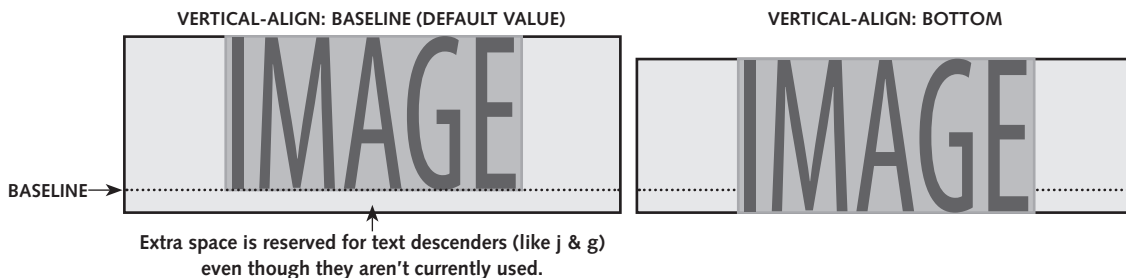### ADDING A BORDER AND FIXING THE IMAGE GAP

The original design calls for a border around the content. Let's add one now.

1. Return to your text/code editor and add the following inline style to the <table> tag:

```
<table width="600" cellpadding="0" cellspacing="0" border="0" align="center"
style="border:1px solid #825102;">
```

2. Save your file and preview your page in a browser. Looks pretty nice except for a strange gap between the footer image on the bottom and the border. What gives?

Images are displayed "inline" and rendered quite like text, even though they're obviously not text. Text (and images) are vertically aligned to the text baseline (where the bottom of each letter is positioned as you would when writing on lined paper) by default. The baseline makes sense for text, including lowercase characters like j and g. However, for images, which don't have descenders, it creates extra, unfilled space below them. The following diagram illustrates this for you.

VERTICAL-ALIGN: BASELINE (DEFAULT VALUE)   **VERTICAL-ALIGN: BOTTOM**

BASELINE→

**Extra space is reserved for text descenders (like j & g)
even though they aren't currently used.**

NOTE: This space below images shows up in **strict** DOCTYPES but not **transitional** DOCTYPES. We're using HTML5, which is a strict DOCTYPE because—while many email clients will strip out your DOCTYPE or leave it out entirely—some clients impose their own DOCTYPE and, invariably, these clients use a strict DOCTYPE.

Luckily, there's an easy fix for this issue. We can tell the browser to render images as "block" elements rather than "inline" elements. Block-level elements do not share the misfortune of being aligned to the text baseline.

3. Return to your text/code editor and add the following code to the <img> tag for the footer, like so:

```
<img src="img/footer.png" height="47" width="598" alt="Omniburger" style="display:block;">
```

4. Save your file and preview your page in a browser. Fixed!

   NOTE: The layout of this email is finished, but there are a few other steps required to get this ready to send out. They will be covered in the following exercises where we code and send out tests using a more complex newsletter design.

OPTIONAL BONUS: ADDING A BACKGROUND COLOR

1. Return to **birthday.html** in your text/code editor.

   We'd like to make the email pop a bit more by creating a different background for the page outside the bordered content of the email. This is typically accomplished in web design by creating a style for the <body> tag but many email clients ignore CSS background colors or strip out the <body> tag wholesale.

   In order to accomplish our goal, we need to wrap the content table with a 100% width single-row, one-column table and give the wrapper table's <td> a background color.

2. Wrap a new table around your current table by adding the following code (in bold):

```
<table cellpadding="0" cellspacing="0" border="0" width="100%">
   <tr>
      <td bgcolor="#f6e0b8">
         <table width="600" cellpadding="0" cellspacing="0" border="0" align="center"
```
(TO SAVE SPACE IN THE WORKBOOK, WE ARE NOT SHOWING THE CONTENT OF THE MAIN TABLE HERE)
```
         </table>
      </td>
   </tr>
</table>
```

3. Save your file and preview your page in a browser. It's a little much. We need to give the coupon content a white background. Return to your text/code editor.

4. Around line 13, find the <td> tag inside the 600 pixel-wide table. Edit the tag as follows (in bold):

    `<td align="center" `**`bgcolor="#ffffff"`**`>`

5. Save your file and preview your page in a browser.

6. There's still a white gutter around the page. In the actual email's preview pane, this won't be there, but we can get rid of it to create a nicer preview page for clients and colleagues. Return to your text/code editor.

7. Edit the <body> tag (around line 7) by adding the following style attribute and properties (in bold):

    `<body `**`style="background-color:#f6e0b8; margin:0;"`**`>`

8. Save your file and preview the page once more in a browser. Much nicer! Remember: these styles in the body tag are not supported by many email clients. We are simply adding them to give the email a "finished" look to replicate what it will look like in most preview panes.

    You're finished with this simple HTML email, so feel free to close **birthday.html** in the browser and in your text/code editor. Bear in mind that there are a few more steps you'd need to take before you can actually send out an email of this nature. We'll investigate those steps in the exercises going forward in the workbook.