1. Establish a Date Object Name

The Date Object

There are many pre-defined objects in JavaScript, and the Date object is one of them. Many methods are associated with the Date object, including getDay, getMonth, and getSeconds. Because methods are something an object does, you can probably guess what these methods do. We will try them out soon.

Note

For a list of other methods for Date, consult the appendix of the optional JavaScript textbook:

Instances

Before we can use an object in JavaScript, we must first create a variable to hold the object. When we talk about an object in Javascript, we must name the particular instance of the object, just as we must identify one particular book of the many that exist in real life. Using a variable name allows us to uniquely identify a particular instance of the object.

Format

\[ \text{variblename} = \text{new Object}() \]

A. Adding onto lab7.html, establish the beginning and end of your script. You will put this script in the body of your HTML above the \(<hr>\) tag and below the \(<br>\) tag. Reference the first step of Lab 7 if needed.

Note

Remember to protect your script from showing up with older, non-JavaScript browsers by using comments.

B. Choose a name to hold your date variable, then assign that name a new date instance, using the format described above.
Example

YourVariableNameForTheDate = new Date()

2. Use the `getDate()` Method to Assign the Current Day to a New Variable

Assignment

To get today’s date, you must use the `getDate()` method on the instance of the object (Date) you created. Once you know how to get the date, you must assign that information to a variable that holds the value for you. As you know from the assignment on variables, you can assign a variable a specific value, such as "dog" or "5," or you can assign it values based on the results of any statement. For example:

```javascript
modified = document.lastModified
```

document.lastModified is equal to the value of the property lastModified, and will be stored in the variable modified.

Note
When you use a method of an object, you use the following syntax: `object.method()`

Remember that lastModified is a property and therefore does not have the trailing `()`.  

A. Create (declare) another new variable to hold the day of the month.

B. Assign this new variable returned by the `getDate` method:

```javascript
YourVariableNameForTodaysDate =
YourVariableNameForTheDate.getDate()
```

Remember!
Like everything in JavaScript, variable names and methods are case sensitive. Note that `getDate` needs to be capitalized as shown, while your variable name should always have the exact capitalization you used when you declared it.
3. Test your Variable by Printing It to Your Web Page

A. **Print the current date** to your Web page, using the `document.write()` method.

B. **Save and test** your page. Does it print the day of the month to your Web page?

**Note**
The location you choose to place your script will determine where the `document.write()` method will print the day of the month. If it is not in the place you would like it to be, try adding another `<script>` tag somewhere else on your page, and print the current date in that location.

C. **Use the `document.write()` method** to add formatting and text before the current date. Two examples are:

   ```javascript
   document.write("Today’s date is: ")
   ```

   and

   ```javascript
   document.write("<p>")
   ```

**Note**
To write HTML onto your Web page from within the `<script>` tag, you must print it to your document using the `document.write()` method. Remember to use quotes around any formatting or text.

4. Create a Conditional in Plain English

In this step we are going to write, in plain English, what we want *JavaScript* to help us do in our next step. We know from the lessons that **conditionals** allow a program to make decisions. To get a taste of conditionals, we are going to write *JavaScript* code that will cause our Web page to have different background colors, depending on the time of day. While this may not be a terribly useful conditional, it is easy to test.
We know from our earlier work with the `Date` object that there are several methods that will retrieve specific information about the current date and time. One of these methods allows us to get the current second. Because we know computers are best able to work with "True" or "False" decisions, we need to phrase our condition so it depends on a "True" or "False" outcome. For example:

If the cost of a piece of cake is less then $3.00, buy a piece of cake.

"Is the cake less then $3.00?" True or False? If "True," then the next part of the statement should be executed, and the cake is purchased. If "False," nothing happens. The end of the statement is not executed.

A. Just after the code you have written in Step 3, start a multi-line comment in JavaScript.

   ```javascript
   Note
   Remember: the start of a multi-line comment is /*
   and the end is */
   ```

B. In this comment, in plain English, write a conditional that will check what second it is, and then change the background of the page to a specific color. You can use whatever comparison operators you want here. Review Lesson Five for more information, if needed. Also, remember to use specifics, such as the exact colors of the background change.

C. Read over the statement you just wrote. Is the outcome predictable? That is, if you know what second it is, will you always know what color the background is going to be?

5. Write a JavaScript Conditional to Change Your Background Color

Now that you have thought out your conditional, we are going to set about writing it. As you know from Lesson Five, the basic conditional form is:

```javascript
if [T/F Expression]
{
    code statements;
}
```
A Quick Note about Curly Brackets and Semicolons in *JavaScript*

If your T/F expression is "True," your conditional will run all code statements within the following **curly brackets**. **Semicolons**, the ; characters, allow you to separate each code statement so you may do several things in response to a "True" result of your conditional. Because *JavaScript*, like *HTML*, does not take spacing into consideration when it reads your code, semicolons are important when you want to do two or more separate things in specific order.

A. **Construct the first line** of your conditional. This can be placed directly after the code from Step 4.

   For example:

   ```javascript
   if (dateobjectvariable.getSeconds() = = 12)
   {
   }
   ```

   **Note**

   It would be silly to use the is equal to operator here, because of the unlikely event the page would be loaded when the second equaled 12. We suggest using a greater than or less than operator.

   B. **Add the code statement** that will change the background color between the curly brackets. Remember: first, identify the object you want to affect (the document), and then the property.

   C. **Save your page and test it.** Does the color change?

   D. **Having trouble?** Are you lost? Go back to the main page for the lab, and click on "Step 5 Conditional."

6. **Add an else Statement to your Conditional**

   In the last step, you changed the background of your form if your conditional was "True." If your conditional T/F expression was "False," then your screen stayed white. Sometimes when you write a conditional, you want to specify
an alternate thing to happen if your T/F statement is not "True." Remember our earlier example of buying cake? Take a look at the following conditional:

If the cost of a piece of cake is less then $3.00, buy a piece of cake. If it isn’t, then buy a brownie.

The syntax to achieve this result involves using an else statement. The syntax is as follows:

```javascript
if (T/F expression) {
    code statements;
} else {
    code statements;
}
```

A. **Add an else statement** to your conditional that will change the background color to something else. It should look something like this:

```javascript
if (dateobjectvariable.getSeconds() = = 12) {
    document.bgColor="blue";
} else {
    document.bgColor="red";
}
```

Remember: your T/F expression is going to be different then the one listed above!

B. **Save** your work, and **reload** your document. Did it change color? If you continue to reload your document, does it eventually change color back and forth?

C. **Add a statement** that will print the current second to your Web page. This will help you test your code. You may add this statement after the end of your if/then/else statement:
**Document.write(document.getSeconds())**

**Note**

You may want to add additional formatting to put this statement on a new line. Add the `<p>` tag by using `document.write("<p>")` before the `<p>` tag writing the current second.

7. **Replace `document.getSeconds()` with a Variable**

In Step 2 you created a variable that holds the value of the current day. You then used that variable to write the current day to your Web page. In Steps 5 and 6, you referenced the current second using the `getSeconds()` method. Now you are going to replace this reference with a variable.

A. **Create a variable** that holds the current second.

   ```javascript
   var YourVariableName
   ```

B. **Assign** the current second to the variable.

C. In your conditional, replace the first instance of `document.getSeconds()` with your new variable name.

D. **Save** your work, and **refresh** your Web page. Did it work the same way?

E. **Replace the other instance** of `document.getSeconds()` with your variable. Again, your program should work as before.

8. **Use an Expression to Add a Label Before Your Current Second**

**Expressions**

"An **expression** is any valid set of literals, variables, operators, and expressions that evaluates to a single value; the value can be a number, a string, or a logical value."

—Netscape Core JavaScript Guide

What does this definition mean? Here are some examples to help:
1 + 1  |  Expression with two literals and an operator (+)  |  Evaluated, it equals 2

For the next examples, $y_{2k}$ holds the value 2000:

$y_{2k} + 1$  |  Expression with one variable ($y_{2k}$) and a literal (1)  |  Evaluated, it equals 2001

An example of an expression that results in a string might be:

"The year was " + $y_{2k}$  |  Evaluated, it equals "The year was 2000"

Note
Adding a number to a string always results in a string.

A. Modify your `document.write()` method that prints the current second, with an expression to add "The current second is " before you print the current second.

```javascript
Document.write("The current second is " + YourVariableNameForTheSecond)
```

B. Remember: the addition operator + adds the variable directly after the last character of the string. It is often useful to put a space as the last character in a string.

9. Use an Expression to Print the Complete Date on Your Web Page

This is a Challenge! You are not required to do this step.

A. Print the current date on your Web page in the following format:

Month-Day-Year

Example: 3-15-2004

B. Important
The method `getMonth()` counts from 0, not 1. January is month 0,
February is month 1, March is month 2, and so on. The way to correct for this is to use an expression to add “1” to the month.