

As of this lab, you have completed the vast majority first pillar of CS 105: you have learned how to *understand a problem like a computer*. In doing so, you have created **fifteen different real programs** (amazing, isn't it?) and took an exam over the basics of the JavaScript language.

Now that this course is at a turning point between understanding how a computer is programmed and using that understanding to process big data sets, we wanted to get some feedback on your experience in CS 105 so far this semester.

**For the first 12 minutes of lab**, you will not have access to the files for this lab assignment. Instead, you will take an **anonymous survey** about your experience with CS 105 so far. Since you don't have access to your lab section files, please take the time to give meaningful feedback to us.

**Complete the survey now.**

*Read the rest of this document once you have completed the survey.*

## Overview

In lecture last week, you learned about the basics of **HyperText Markup Language** or **HTML**. Throughout the entire semester, you have been provided HTML as part of your MPs, Weekly Readings, and Lab Assignments. Today, you will start to interact with the HTML.

As we have done in the past, you will find that we have prepared a ZIP file for you on the course website. This week, you will find the ZIP file does not contain a JavaScript file -- instead, you will be writing the JavaScript right into the HTML!

## The Tic-Tac-Toe Game

The game of **tic-tac-toe** is a classic paper-based game that is played on a 3x3 grid between two players. The first player traditionally takes on the symbol **x** and places an **x** in one of the nine cells. The second player, using the symbol **o**, places her symbol in a second cell that is currently not occupied. This process repeats itself until either no cells are left (the game ends with no winner) or until one player has three symbols in a row (virtually, horizontally, or diagonally).

X	O	
O	X	X
X		O

When you open up `game.html` in a web browser, you will find that the HTML is already set up with a simple 3x3 board. However, if you click on a cell or try and interact with the game, the game does nothing. You should open up `game.html` in a **plain text editor** like Notepad++ to view the HTML code behind the website.

The HTML is made up of elements that you are familiar with from lecture: `<html>`, `<body>`, lots of `<div>`s, and an empty `<script>` element. There are nine key `<div>` elements in the HTML that represent the nine cells on the grid, all of which look similar in code:

```
<div id="cell_top_left" onclick="clicked('cell_top_left');"></div>
```

You should be able to identify the nine different cells by looking for similar elements.

## Understanding the HTML Structure

---

Looking carefully at the nine `<div>` elements that correspond to the nine cells on the HTML web page, you should notice they all have an `onclick` attribute. As introduced in lecture, this HTML attribute will run some JavaScript when a user clicks on the HTML element.

Specific to our web page, the JavaScript in the `onclick` attribute calls a function called `clicked`. The `clicked` function takes in one parameter – a **string**. Each cell passes in a different string into the function's parameter, but the string that is passed in is always the value of the `id` of the cell.

Finally, there is one last element in the HTML that is important. At the bottom of the HTML, there is a `<span>` element (an element that works similarly to a `<div>` tag to divide up content within a web page) that contains the current turn: `<span id="current_turn">X</span>`. The value in the element with the `id` attribute value of `"current_turn"` should always contain the player who is currently about to make a turn.

## Programming the Game

---

To complete this lab, you simply need to program the JavaScript so that **when a cell is clicked, the current player's symbol appears in that cell**. This can be divided up into several small sub-tasks that each represents about one line of code:

- Define the `clicked` JavaScript function inside the `<script>` area of the HTML web page. Remember, the `clicked` function takes in one parameter – the `id` of an element that was just clicked.
- Find out which player is the current player. You will want to store their symbol into a variable so that you can use it later. You may find `document.getElementById("x").innerHTML` to be useful here, where `"x"` is the id of some element in the HTML.
- Replace the innerHTML of the cell that was clicked with the current player's symbol. You may find `document.getElementById(id).innerHTML` to be useful here, too.
- Finally, you need to update the current player. If the player was `x`, the new player is `o`; if the player was `o`, the new player is `x`.

Once these four steps are complete, you can a team member can play a game of tic-tac-toe! You may notice there are several bugs still in your game (the game does not stop when someone wins, you can play on top of another player instead of only in an empty cell, and others) – **you do not need to worry about these problems for the lab**.

## Submitting the Lab

---

As usual, submit the lab via the CS 105 website. Remember, **every person in your group must submit the lab**. It is your job to make sure everyone who is in your group who is in lab has a complete program as one group member will randomly be chosen to be graded.