# Assignment07 – Hidden Hippos Game Part 1

In this assignment, students are going to start work on a simple game called “Hidden Hippos”. ***Be sure to save your work because you will need it in future assignments.***

When you are done with this assignment you won’t (yet) have a working, completed game. But you will be ready for a future assignment where you will make something a bit more satisfying.

You will use your knowledge of functions, strings, numbers, and scope to write this code.

## Overview

Important notes:

* In this assignment you will write JavaScript code that renders an HTML table. You will do this using a **for** loop, nested inside another **for** loop.
* If you need help understanding HTML table syntax. See this video from earlier in the semester: <https://youtu.be/6xldLvon-yI>
* The size of the table will be controlled by the user. That is: is it a 3 x 3 table? 6 x 6? 10 x 10? Bigger?
* You will need to implement error trapping.
* The table will contain random number of hippos. The maximum number of hippos is controlled by a **global constant** called maxhippos. There is a second global variable called numhippos used to keep track of how many hippos you have already shown. No other global constant or global variable should be used. ( As a general principle, it is best to use global variables sparingly.)
* You will use concatenation to build a string. The string of characters represents the HTML table.

**CONTINUED**

## Instructions

1. Download assignment07.zip as provided by your instructor.
2. Start your work by declaring a global **constant** near the global variable at the top of the script.
   1. You will need a constant called maxhippos with a value of 5
   2. Notice that the variable called numhippos with an initial value of 0 has already been created.
3. You must complete two functions that have been started for you. Those functions are: hippoBuildGrid() and hippoCellContent().
4. Start your work by completing hippoCellContent.

***You do not need a loop to complete this function.***

***This function does not need error trapping.***

This function randomly returns one of two things.

It returns either an empty string, or the HTML tag for a hippo icon. Here is the HTML tag for a hippo:

<i class="fa fa-hippo"></i>

**Careful!** If the maximum number of hippos has been reached, then this function will only return an empty string.

This function accepts one parameter, N. The higher N is, the lower the chance of getting a hippo.

* If N is 1, then there is a 50% chance of getting a hippo.
* If N is 2, then there is a 25% chance of getting a hippo.
* If N is 3, then there is a 16.67% chance of getting a hippo.
* If N is 4, then there is a 0.125% chance of getting a hippo.

… and so on for any value of N. It’s up to you to figure out the math there.

One more:

* If N is 10, then there is a 0.05% chance of getting a hippo.

Write your code for hippoCellContent.

**CONTINUED**

1. Test hippoCellContent using the Web Developer Console in Chrome.

You can see screenshots of what testing looks like below. Notice that, once 5 hippos have been returned, then the function stops returning hippos. Also, because there is an element of chance, there’s no guarantee that you will get the exact same results.

|  |  |
| --- | --- |
| A screenshot of a cell phone  Description automatically generated |  |

**CONTINUED**

1. Now complete the function hippoBuildGrid.

This flowchart diagram illustrates how your code should render the HTML table.

A screenshot of a diagram

Description automatically generated

## Check your work

1. Compare your work to these videos. These videos have no audio.

* Test each function. <https://youtu.be/dWLqBuQ9Jhc>
* Test for different values of N. <https://youtu.be/V1TaoIjUWLo>

## Notes for later / Things to not worry about

1. Our solution tends to favor the top of the grid. That is, because we limit the total number of hippos, and loops render one cell at a time, hippos are more likely to appear sooner in the grid rather than later. That’s OK for now. We will address this in a future assignment.
2. Some grids will have fewer hippos than the maximum allowed hippos. That’s ok.
3. When N is small (for example, N = 2) it is possible to have a table with no hippos at all in it. This is a problem that we will address in a future assignment.

## Turn in your work

1. Publish your work to misdemo. Check your URL.
2. Go to canvas and turn in your URL there. A correct URL will start with **https://misdemo** and end in **assignment07/hippogame.html**

**CONTINUED**

## How will this assignment be graded?

|  |  |
| --- | --- |
| The function hippoCellContent works as expected. | 40 |
| Correct declaration and use of maxhippos | 10 |
| Correct formula used in hippoCellContent | 10 |
| The function hippoBuildGrid has good error trapping. | 10 |
| The function hippoBuildGrid runs as expected | 20 |
| The behavior of the solution matches what is demonstrated in the video | 10 |
| **TOTAL** | **100** |