



Jumpstart Weekend

LEARN Academy

Weekend goal: Treasure Hunt

This weekend we are tasked with creating a web game. When developers are planning a job it is common to break down the steps into user stories.

- As a user, I can see a webpage with a treasure hunt game
- As a user, I can see the title and rules of the game
- As a user, I can see a board game on a 3x3 grid with a question mark in each square
- As a user, when I click on the square the question mark will turn into an icon
- As a user, if I click the correct square a treasure will appear
- As a user, if I click the wrong square a bomb icon will appear
- As a user, if I click any other square a tree icon will appear
- As a user, I can see a counter that tells me the number of turns I have left
- As a user, if I click on the square with the treasure I will see a notification that I won
- As a user, if I click on the square with the bomb I will see a notification that I lost
- As a user, I can continue clicking squares until I hit the treasure or the bomb

- As a user, once I click on the square with the treasure or bomb I can no longer play the game
- As a user, I can click a button to play a new game

Stretch goals (ice box):

- As a user, I can see a board game with a 5x5 grid
- As a user, I only have 20 chances to find the treasure
- As a user, I can navigate to the link on the internet



HTML

HTML (**H**ypertext **M**arkup **L**anguage) is the computer language that displays information to the user in a web browser

To create HTML computer code:

- Text editor – at LEARN we use Atom
- Save a file with the extension .html (example: index.html)
- Don't use spaces or capital letters while naming directories or files
- Setup the basic tags needed for your page called the DOCTYPE

```
1  <!DOCTYPE html>
2  <html lang="en" dir="ltr">
3    <head>
4      <meta charset="utf-8">
5      <title></title>
6    </head>
7    <body>
8      |
9    </body>
10 </html>
```

- HTML uses “tags” that are indications to the computer what information to display and how it should look
- A tag is surrounded by carrots < >
- A set of tags will open <div> and close </div>

- The text inside the opening and closing tags is displayed in the browser, this is called the innerHTML
- Attributes are extra pieces of information stored inside the opening tag to modify how the browser displays information
- Tags that just store information, like images don't have to close, these are called meta tags

```
8
9  <body>
10
11    <h1 id="page-title">Page Title Here</h1>
12    <img rel="https://www.web-pic.png" alt="detailed description of pic">
13
14  </body>
15
```

- The entirety of the code that appears on the page is nested in the <body> </body> tags in the HTML DOCTYPE
- Indentation matters!
- Save your work often (command s)
- Drag the icon in the top of your atom window into Google Chrome to view your page
- If you make changes to your code you must refresh the browser (command r)

HTML practice exercises:

Top 10 Favorite Animals

- Create a directory (folder) on the desktop called Jumpstart-March
- Open Atom
- Save a file inside the directory you created with an .html extension
 - Remember: **no** spaces or capital letters
- Set up the DOCTYPE
- Add a title for your page
- Add a subheader that list each of your top ten favorite animals
- Add a title to the browser tab
- Add a favicon to the browser tab

```
3  <head>
4    <meta charset="utf-8">
5    <title>Browser Tab Info Here</title>
6    <link rel="shortcut icon" href="https://www.picture-website.jpg">
7  </head>
8
```

- Add a picture of each of your animals
- Add a list of three interesting facts about each animal

ul – unordered list, (bullet points)

ol – ordered list (numbered)

li – list item, nested inside the ul/ol tag

```
9  <body>
10
11    <ul>
12      <li></li>
13      <li></li>
14    </ul>
15
16  </body>
```

- Add a link to an external website that will give you more information about each animal
- Add a navigation bar to the top of your page that provides links to all your favorite animals

```
<nav>
  <a href="#pangolin">Pangloin</a>
  <a href="#zebra">Zebra</a>
  <a href="#okapi">Okapi</a>
  <a href="https://www.google.com/" target="_blank">Google</a>
</nav>
```

TREASURE HUNT TO DO (HTML):

- Create a new HTML file in your Jumpstart-March directory called index.html
- Set up the DOCTYPE
- Add a header and subheader to your game
- Make a list of the game rules
- Make a table to create the game board, keep the indentation organized

tr – table row, nested inside the table

td – table delimiter, nested inside each table row

```
9      <body>
10
11      <table>
12      <tr>
13      <td>?</td>
14      <td>?</td>
15      </tr>
16      </table>
17
18      </body>
19
```

- Add a title to the browser tab
- Add a favicon to the browser tab
- Add an image tag with a picture to decorate your page, you may need to add a height attribute
- Add a footer to the page with your names <footer> </footer>
- Add a link to the footer to take your user to the top of the page
- Add a link to the footer to take your user to an external website
- Create a button to restart the game (your button will be useless for today, functionality will come later)

```
<button type="button">Restart Game</button>
```



JavaScript

JavaScript is a dynamic computer language that creates interactive experiences with the user

JavaScript basics:

- Chrome console has a platform to run JavaScript
 - Inspect the page (right click in the browser)
 - Click over to the next tab called Console
- Data types – defining information for the computer, there are lots of data types but we are only going to worry about three of them for now
 - Number - integer, float
 - String - any information stored in “ ”
 - Boolean - true/false
- Variables – empty buckets that hold information
 - Four important parts
 - Keyword var
 - Declaration
 - Assignment operator
 - Value
- Comparison operators

< =< > >=

- Equality operators

`== === !==`

- JS “vocabulary” – words and actions built in to the JS language
 - Reserved words – var, function, if, else
 - Built in methods

```
.alert()  
.prompt()  
.toLowerCase()  
.getElementById()
```

- Conditionals, decision structure

```
if(value === condition){  
|   perform this action  
} else {  
|   perform this action  
}
```

- Functions - little reusable machines that take in code and evaluate the code

```
var myAction = (placeholder) => {  
  if(placeholder === condition) {  
    perform this action  
  } else {  
    perform this other action  
  }  
}  
  
myAction(actualValue)
```

JavaScript practice exercises:

1) Joke – write a program that uses the alert() function to tell a joke

```
"Why did the chicken cross the road?"  
"To get to the other side."
```

2) Greeter – write a program using prompt that asks for your name, then says hello to you

```
"What is your name?"  
"Hello, " + yourName
```

3) Madlibs of the Amazon – write a program that tells you a funny story

- As a user, I see a series of prompts that asks me for an adjective, a noun, a liquid, a body part, a verb, noun, and a place
- As a user, I see an alert of a funny story that includes my answers

```
var adjective = prompt("Please enter an adjective.")  
var noun = prompt("Please enter a noun.")  
etc....
```

```
"Piranhas are more *adjective* during the day, so make sure you  
cross the *noun* at night. Piranhas are attracted to fresh *liquid* and  
will most likely take a bite out of your *body part* if you *verb*.  
Whatever you do, if you have an open *noun* try and find another  
way to get back to *place*. Good luck!"
```

4) World domination – write a program that determines the user’s ability to rule the world, use the JS method `.toLowerCase()` to help the user get the correct outcome

- As a user, I can see a prompt that asks me my name
- As a user, I can see a prompt that asks me if I know how to write code
- As a user, if I reply “yes” I will see the alert, “You will rule the world, *name*!” otherwise, “Well, good luck with that.”

5) Magic 8 Ball – write a program that asks the user a question and returns an answer

- As a user, I see a prompt where I can ask a question
- As a user, I see an alert that tells me a random Magic 8 Ball answer
 - o Better not tell you now
 - o It is decidedly so
 - o Don’t count on it
 - o Signs point to yes
 - o Outlook not so good

Hint....

```
var randomNum = Math.floor(Math.random() * 5)
```

TREASURE HUNT TO DO (JS):

- Create a new file in the same directory called treasurehunt.js
- Add a script tag to the bottom of the HTML body so the HTML page knows it has a dependency, notice there is no innerHTML

```
<script type="text/javascript" src="treasurehunt.js"></script>
</body>
```

- As a first step to connecting the HTML and JS, create a function called treasure that takes an argument of location and returns an alert displaying the id of the square clicked
- Add an onclick attribute to all the nested table tags so when our user clicks on the question mark it will trigger the treasure function to run in the JavaScript file

```
<td onclick="treasure(0)">?</td>
<td onclick="treasure(1)">?</td>
```

- Add a unique id to all <td></td> tags so we can identify which td was clicked and perform an action on the correct location
- Set up a variable to store the random location of the rainbow treasure
- Set up a variable to store the random location of the bomb
- Add a conditional statement that will alert either the location of the bomb, the location of the treasure or the id to alert the location of the treasure and bomb

- Replace the alert with an action that will change the question mark in the td innerHTML to the tree, bomb, or treasure icon

🌲 = tree icon

☠ = bomb icon

🌈 = rainbow icon

```
document.getElementById(location).innerHTML = "&#x1f308"
```

- Add an onclick to the button that will refresh the page, resetting the game board

```
<button type="button" onclick="location.reload()">Restart Game</button>
```

- Set up the counter to keep track of the number of clicks
 - Assign a variable to the base value
 - Reassign the variable every time the treasure function runs
 - Attach the counter to the HTML with an id tag
- What if the bomb location and the treasure location are the same?
- How does the game end?
- Stretch goals with expanding the board game



CSS

CSS (**C**ascading **S**tyle **S**heets) is a styling language that describes how elements should be rendered in the browser, the dressup of coding languages

CSS basics:

- CSS doesn't do much without HTML for it to act on, it is the skin covering the skeleton code of HTML
- CSS styling can be written once, and used by multiple tags in your HTML code
- Creating a styling sheet helps us achieve the coding philosophy of "separation of concerns" and keeping your code DRY
- Styling in CSS can be more difficult than you expect, it comes down to a lot of trial and error, don't be discouraged
- There are many wonderful online resources that help with finding the correct CSS attributes, check out W3Schools and
- CSS is set in *key: value* pairs targeting a particular HTML tag, id, or class

```
h1{  
  color: blue;  
  font-size: 50px;  
}
```

- To target a tag's id use `#the-id-name-here`
- To target a class use `.the-class-name-here`
- Particular CSS properties have a shortcut that can be listed together in one key: value pair as long as they are in a particular order

```
#header-section{  
  border-radius: 4px;  
  border-style: solid;  
  border-color: black;  
}
```

Can also be written as:

```
#header-section{  
  border: 2px solid black;  
}
```

TREASURE HUNT TO DO (CSS):

- Create a new file in the same folder called treasurehunt.css
- Add a link tag to the head of the HTML page

```
<head>
  <meta charset="utf-8">
  <title>My tab title here</title>
  <link rel="stylesheet" href="treasurehunt.css">
</head>
```

- This is an opportunity for you to explore styling
- There are many, many possibilities for creating a look for your game, have fun and be creative

Possible options include:

- Google fonts:
 - Select the font you like
 - Click the + sign
 - Copy the link tag from the popup and paste in the HTML head tag
 - Copy the CSS *key: value* pair from the popup and add to CSS file
- Background image – this is a great description of background image attributes from W3School using the HTML class selector hero-image

```
.hero-image {
  background-image: url("photographer.jpg"); /* The image used */
  background-color: #cccccc; /* Used if the image is unavailable */
  height: 500px; /* You must set a specified height */
  background-position: center; /* Center the image */
  background-repeat: no-repeat; /* Do not repeat the image */
  background-size: cover; /* Resize the background image to cover
the entire container */
}
```


- Hover attributes
 - There is a particular subset of CSS attributes called pseudo selectors
 - They act on other attributes to make modifications to the styling such a hover attribute that would only show the styling when the cursor is over the HTML tag
 - This attribute will not behave well on mobile apps as the “cursor” on your phone is your finger or stylus touching the screen
- Borders
- Font colors – CSS recognizes many color names, but to have a more specific hue/shade look for color #hex codes
- Animation
 - Uses an attribute @keyframes that breaks down the total length of your animation into as many frames as you choose
 - The difference in each frame creates the movement or action
 - This website breaks explains each of the animation attributes and gives great examples:

<https://thoughtbot.com/blog/css-animation-for-beginners>



Github

Github is a platform for git (**G**lobal **I**nformation **T**racker) a version control system developers use to save and share code

Github uses terminal commands to communicate directly from the computer to the git platform. Developers use the terminal to manipulate programs and access information directly from the computer without the GUI (graphical user interface).

TO DO (github):

- Create a github account (it's free)
- You will need to remember your github handle (username) and your password to deploy your application
- Together we will create a repository and use github pages to deploy our project through the command line interface