INTRODUCING JS

There are two ways to introduce JS to HTML, being embed or link. We'll explore the effects of each way.

UNDERSTANDING JS

JS stands for JavaScript. JavaScript is an interaction action based programming language, *dynamic behavior*. Js goes through one element at a time. Js is more complex in syntax because it has multiple actions of methods, functions properties etc of combinations. So for now let's focus on one.

The importance of structure and proper usage of id, class, and tags or names comes into place more-so in HTML because we use it as the basis of referencing in other programming languages. JS uses HTML's element names to connect visual transformations or actions. Content not defined in tags are least controllable to define, and it is floating without rules.

USING JS

To connect an element from HTML we use the type of tag or a unique created name which will be discussed later. Let's see how some tags can be styled or instructed to do an action.

In HTML we've seen that our data goes in between an opening and closing set of tags, <>info</>. In CSS the HTML's element is referenced and then information to attach to it goes in between the curly brackets, HTML element{info;}.

To achieve the same as we did in css, a few more add-on steps has to be taken. In this particular action, the document has to be called on first to instruct. Then following, append to document with a period symbol followed by a method you want to perform, and parentheses of the target or output desired inside of them.

So let's say document do this, get these html elements by this tag name inside of these parentheses inside of quotes. If that is all that is wanted, then you can end the line there. However if you want to do something with these elements after they are retrieved, you can add-on. Let's see below and then explain more afterwards.

Example:

<body> </body> in HTML

body{ background-color : green; } in CSS

tag name style property = value close/end
This HTML element or elements gets these properties in the curly brackets.

document.getElementsByTagName('body')[0].style.color = "green"; in JS

this document linked to append period get these HTML elements with the same tag name tag name individual element index number of that same tag name group append period style css append period property = value close/end
This specific index HTML element gets this appended property in the quotes after the equal sign.

JS is trying to attach actions to an element or elements but executing on only one element at a time. Unlike CSS, we can change elements with the same tag name all at once easily, if we wanted all of those to have the same appearance. Also in CSS, you can select each individual element with the same tag name to style each different. This is similar to the steps JS has to do to achieve one or all elements with the same tag name to have similar or different properties. So we have to be more specific in JS.

If it were just this code below it would be invalid without the index number inside the square brackets, [ ] .

document.getElementsByTagName('body').style.color = "green"; in JS

This code below is valid only in accessing & retrieving the body tag group itself, it is not considered an individual element.

document.getElementsByTagName('body'); //Important to remember accessing moving forward. Note element is plural

Why?

Tags in HTML are possible to have more than one of the same type. Any HTML element type or naming attribute possible to be considered capable of having more than one of the same type is considered a group, known as an array in JS. This is so regardless if there is only one thing of that type of tag. *Array is a list or group of elements of relative same type.* The order in which each element is placed in the array is based on the order it is written, and will be the order for its' access and retrieval. So again because JS is trying to attach actions to a single element at a time, an array is like a container, and JS is asking which element in this array container do you want to access. Let's take a look at a new example below.

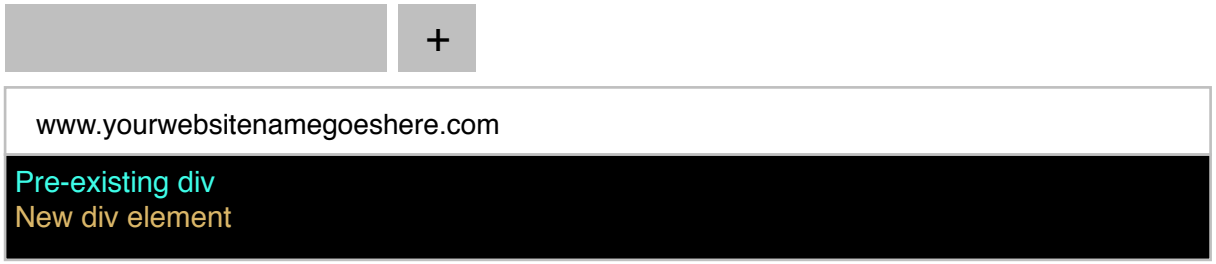
```
<html>
   <head> </head>
   <body>
        <header> </header>
        <nav> </nav>
        <div>Pre-existing div</div>
        <div>New div element</div>
        <img/>
        <footer> </footer>
      <script>
      //          /**/
      </script>
   </body>
</html>
```

New Example: Added Div Tag to HTML/ JS Syntax Access One El of Array:

Using HTML Script Tag to Code Inside HTML

```
<script>
  //See the new div tag added to the HTML Now there is 2 div(s).
 document.getElementsByTagName("body")[0].style.backgroundColor = 'black';
 document.getElementsByTagName("div")[0].style.color = "aqua";
 document.getElementsByTagName("div")[1].style.color = "tan";
 //Single line comment                    /* Multi-line comment */
</script>
```

HTML With JS Output to Browser Window:



Script tag represents JS in HTML HTML tags accessed from inside of JS

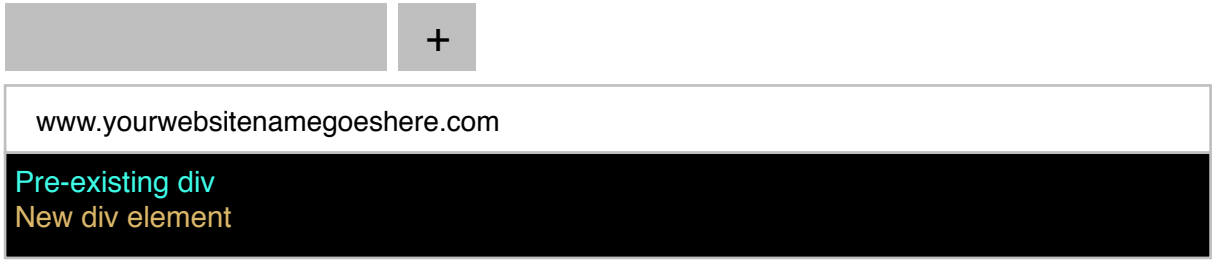
```
<html>
    <head> </head>
    <body>
        <header> </header>
        <nav> </nav>
        <div>Pre-existing div</div>
        <div>New div element</div>
        <img/>
        <footer> </footer>
      <script src="file location">
      //          /**/
      </script>
    </body>
</html>
```

New Example: Added Div Tag to HTML/ JS Syntax Access One El of Array:

Using HTML Script Tag to Connect JS File Code

```
//Coding in a JS file is the same, just no need for script tags inside file
//See the new div tag added to the HTML Now there is 2 div(s).
document.getElementsByTagName("body")[0].style.backgroundColor = 'black';
document.getElementsByTagName("div")[0].style.color = "aqua";
document.getElementsByTagName("div")[1].style.color = "tan";
//Single line comment                    /* Multi-line comment */
```

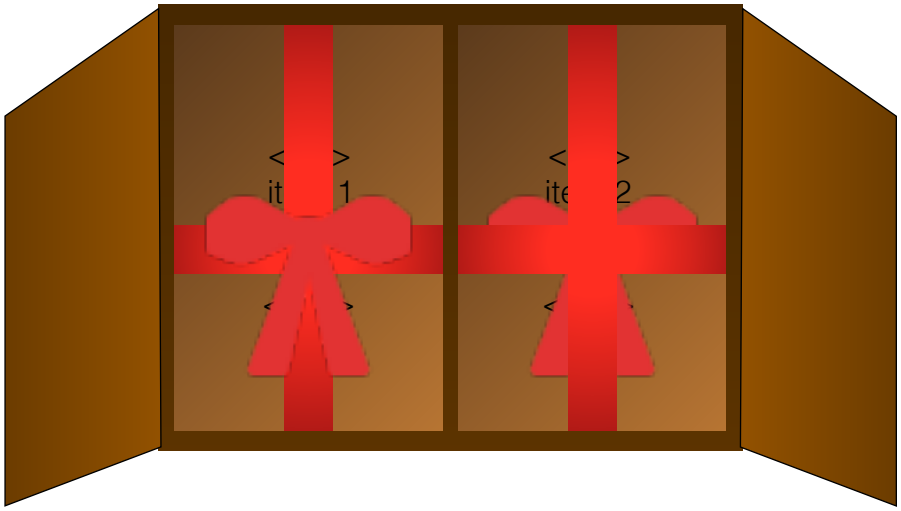
HTML With JS Output to Browser Window:



ARRAY EXAMPLE

DELIVERY CAME
ARRAY BOX CONTAINER OF ALL DIV TAGS

2 GIFTS: CANNOT SEE WHAT IT IS
THERE IS MORE THAN ONE DIV BOX TAGS HERE



STEP 1: OPEN IT
document.getElementsByTagName(‘’)

STEP 2: CHOOSE WHICH ITEM TO OPEN
POSITION LEFT = INDEX 0, POSITION RIGHT = 1
THE ORDERED IT WAS PACKAGED
[ ]

SHOWTIME, OPENED LEFT INDEX 0

WAIT TO SEE GIFT 2, CLOSING UP

STEP 3: ACTION SHOWS CONTENT INSIDE
style.visibility=“visible” or other action

STEP 4: THAT’S ALL THAT WAS WANTED TO DO
;

Note: An array starts counting from zero on position, even though actual element items inside are started counting from one.