The background features a complex, abstract design composed of several concentric circles and arrows. The circles are rendered in white and light gray, creating a sense of depth and motion against a dark blue gradient background. Some circles have arrows pointing clockwise, while others have arrows pointing counter-clockwise, suggesting a dynamic or cyclical process.

JS

JAVASCRIPT

PRESENTED BY STEVE LOURENCO

LAST WEEK

CSS Layouts

CSS Framework: Bootstrap 4

WHY DO WE NEED JAVASCRIPT?

- HTML defines the content
- CSS specifies the layout of a webpage
- JavaScript allows us to program the behavior of a web page

WHERE CAN WE USE JAVASCRIPT?

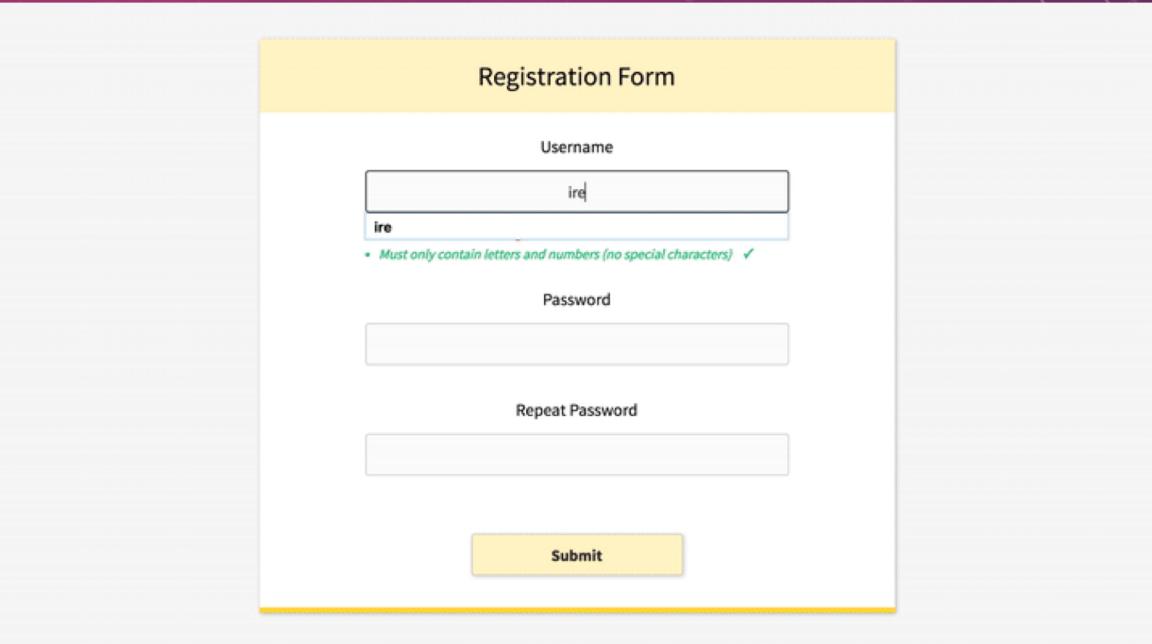
- **Client side validation** (verify any user input)
- **Manipulating HTML Pages** (change the content of an HTML tag)
- **User Notifications** (raise dynamic pop-ups on the webpages to give different types of notifications to your website visitors)
- **Back-end Data Loading (AJAX)** : (Instead of loading content by loading the whole page, we can see AJAX to load asynchronously content)
- **Server Applications** (Nodejs, is a Javascript Server. We can compare with Rails. Rails is a ruby server.)

CLIENT-SIDE JAVASCRIPT

- JavaScript is most used in the client-side
 - This means the browser loads HTML, CSS and **JavaScript**
After finishing loading it will execute the JavaScript in the user side and not in the server side
 - We can execute JS (JavaScript) with help of user-initiated events (Explained in the last slides)
 - Example: User clicks on a HTML button, HTML link

ADVANTAGES OF JAVASCRIPT

- **Less server interaction** (validate user input before sending the page off to the server)
- **Immediate feedback to the visitors** (don't have to wait for a page reload to see if they have forgotten to enter something)



A screenshot of a registration form titled "Registration Form". The form has three fields: "Username", "Password", and "Repeat Password". The "Username" field contains the value "ire" and has a validation message below it: "Must only contain letters and numbers (no special characters) ✓". The "Submit" button is located at the bottom right of the form.

- **Richer interfaces** (You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors)
- <https://apvarun.github.io/toastr-js/#>

WHERE TO INSERT

- In HTML, JavaScript code must be inserted between `<script>` and `</script>` tags.
 - https://www.w3schools.com/js/tryit.asp?filename=tryjs_whereto
- Scripts can be placed in the `<body>`, or in the `<head>` section of an HTML page
 - Example with the script in the head tag:
https://www.w3schools.com/js/tryit.asp?filename=tryjs_whereto_head
 - Example with the script in the body tag:
https://www.w3schools.com/js/tryit.asp?filename=tryjs_whereto_body
- Scripts can also be placed in external files
 - Example: `<script src="myScript.js"></script>`
 - https://www.w3schools.com/js/tryit.asp?filename=tryjs_whereto_external
 - External scripts cannot contain `<script>` tags

DISPLAY POSSIBILITIES (MOST IMPORTANT)

- Writing into an HTML element, using **innerHTML**
- Writing into an alert box, using **window.alert()** → You can use an alert box to display data:
- Writing into the browser console, using **console.log()**

innerHTML

- To access an HTML element, JavaScript can use the **document.getElementById(id)** method
 - **Document** → Means the webpage
 - **getElementById** → Allow us to find an HTML element by ID
 - **ID** → Enter the ID name of your element
- **innerHTML** → Defines the HTML content
- https://www.w3schools.com/js/tryit.asp?filename=tryjs_output_dom

STATEMENTS

- alert("Wake up!") is a statement
- No limits of statements
- Add a semicolon at the end of each executable statement:
- JavaScript Code Blocks: JavaScript statements can be grouped together in code blocks

```
function myFunction() {  
    document.getElementById("demo1").innerHTML = "Hello Dolly!";  
    document.getElementById("demo2").innerHTML = "How are you?";  
}
```

- JavaScript Keywords: statements often start with a **keyword** to identify the JavaScript action to be performed:
https://www.w3schools.com/js/js_reserved.asp

VARIABLES

- Javascript allows us to work with data. Here are three primitive data types:
 - Number
 - String
 - Boolean
- JS allows us to store those data types in **variables**
 - **var name = "Ali";** → Store the string ALI in the variable name

Another example:

- **var money;**
- **money = 2000.50;**

VARIABLE SCOPES: GLOBAL VS LOCAL

- **Global variable:** A global variable has global scope which means it can be defined anywhere in your JavaScript code.
- **Local Variables** – A local variable will be visible only within a function where it is defined. Function parameters are always local to that function.

You can call a global variable inside a function but you can't call a local variable outside of the function.

OPERATORS

- Arithmetic Operators
- Comparison Operators
- Logical (or Relational) Operators
- Assignment Operators
- Example: https://www.tutorialspoint.com/javascript/javascript_operators.htm

IF ... ELSE

- With help of operators, we can now create conditions which can only be executed if true

```
• var age = 20;  
  
if( age > 18 ) {  
    document.write("<b>Qualifies for driving</b>");  
}
```

```
var age = 15;  
  
if( age > 18 ) {  
    document.write("<b>Qualifies for driving</b>");  
} else {  
    document.write("<b>Does not qualify for driving</b>");  
}
```

- https://www.tutorialspoint.com/javascript/javascript_ifelse.htm

- Others:

https://www.tutorialspoint.com/javascript/javascript_switch_case.htm

FOR LOOP

- The for loop has 3 parameters
 - **loop initialization** → We initialize our counter
 - **test statement** → test if a given condition is true or not
 - **iteration statement** → increase or decrease rule of the counter

```
var count;  
  
document.write("Starting Loop" + "<br />");  
  
for(count = 0; count < 10; count++) {  
    document.write("Current Count : " + count );  
    document.write("<br />");  
}  
  
document.write("Loop stopped!");
```

WHILE LOOP

- The purpose of a **while** loop is to execute a statement or code block repeatedly as long as an **expression** is true. Once the expression becomes **false**, the loop terminates.
- Possible to create a infinite loop → while (true) { }

```
var count = 0;  
document.write("Starting Loop ");  
  
while (count < 10) {  
    document.write("Current Count : " + count + "<br />");  
    count++;  
}  
  
document.write("Loop stopped!");
```

FUNCTIONS

- group of reusable code which can be called anywhere in your program
- eliminates the need of writing the same code again and again
- allow a programmer to divide a big program into a number of small and manageable functions

```
function sayHello() {  
    alert("Hello there");  
}
```

- To invoke the created function somewhere later in the code, you just need to write the name
- → sayHello();

EVENTS

- JavaScript's interaction with HTML is handled through **events** that occur when the user or the browser manipulates a page.
 - When the page loads, it is called an event. When the user clicks a button, that click too is an event. Other examples include events like pressing any key, closing a window, resizing a window, etc.
 - https://www.tutorialspoint.com/javascript/javascript_events.htm

ONCLICK EVENT TYPE

- most frequently used event type

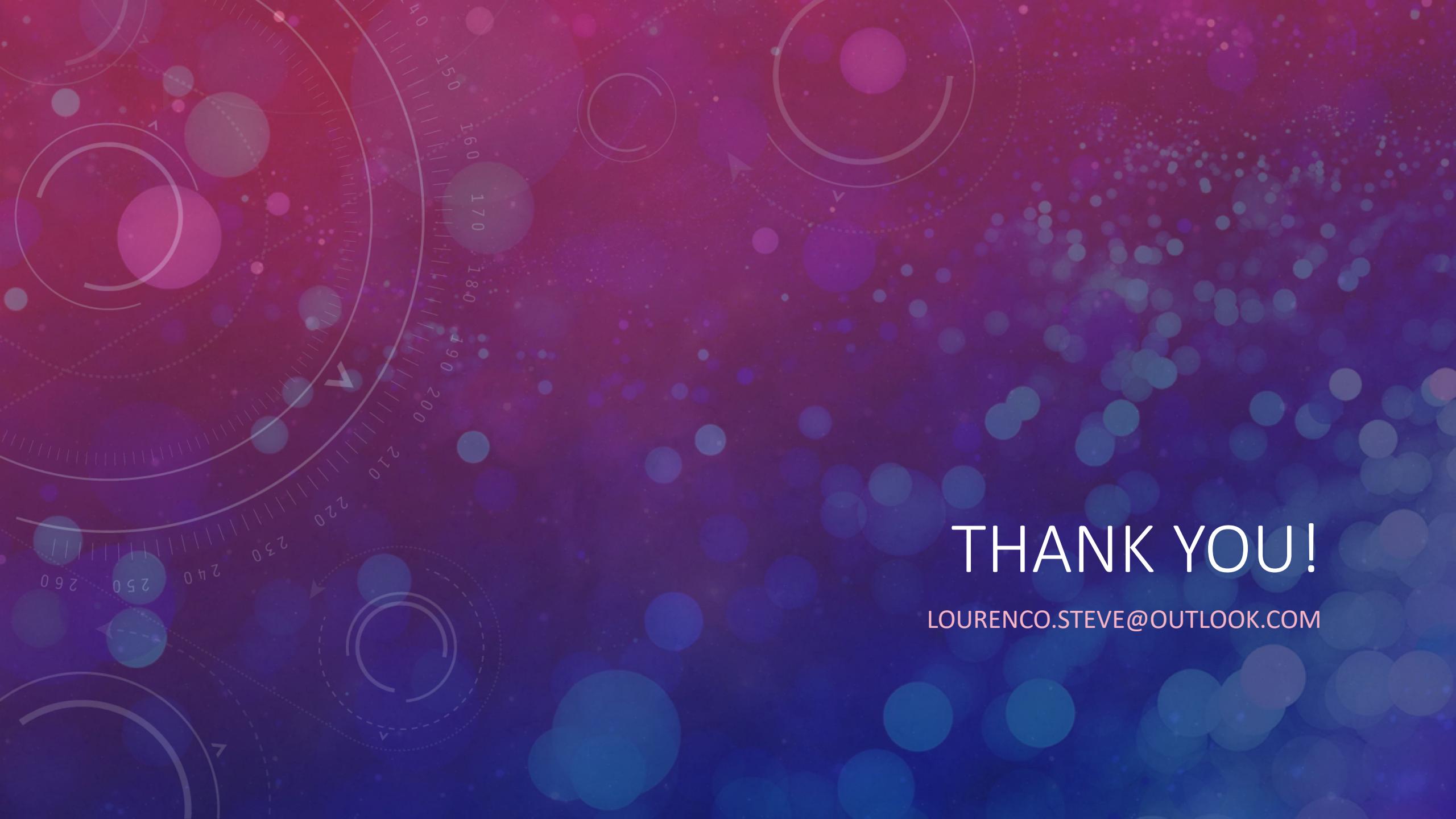
```
<html>
  <head>
    <script type = "text/javascript">
      <!--
        function sayHello() {
          alert("Hello World")
        }
      //-->
    </script>
  </head>

  <body>
    <p>Click the following button and see result</p>
    <form>
      <input type = "button" onclick = "sayHello()" value = "Say Hello" />
    </form>
  </body>
</html>
```

Live Demo

MORE EVENTS

- https://www.tutorialspoint.com/javascript/javascript_events.htm

The background features a complex, abstract design. On the left side, there are several concentric circles in white and light gray, with numerical values ranging from 40 to 260 marked along their outer edges. These circles are partially obscured by numerous small, semi-transparent blue and white circular bokeh lights of varying sizes, creating a sense of depth and motion.

THANK YOU!

LOURENCO.STEVE@OUTLOOK.COM