Web Application Development

JavaScript

1

Assignment2

```
git status

git add .
git commit -m “message”

git init
git remote add origin “github Repository SSH link”

github.com vs. github.io
```

Upload to **github.io**

```
cd
pwd
ls -all
rm -rf .git
```

2

Introduction

- History
  - Originally developed by Netscape
  - Joint Development with Sun Microsystems in 1995
  - JavaScript was a trademark of Sun Microsystems, now is a trademark of Oracle.

- HTML to define the **content** of web pages
- CSS to specify the **styles** of web pages
- JavaScript to program the **behavior** of web pages

3

basic.html

Where does JavaScript go?

```
<head>
    <script >
      function foo(){
        document.getElementsByTagName("h1")[0].innerHTML="Hello World!”
        window.alert ("Hello World!");
      }
    </script>

    <script  src="basic.js"></script>

</head>
<body>
    <h1> Hello fellow Web programmers! </h1>
    <h1> Don't forget to submit assignment!</h1>
    <button type="button" onclick="foo()">Click me</button>
</body>
```

Embedded

External

Javascript can be either inside <head> or <body>

4

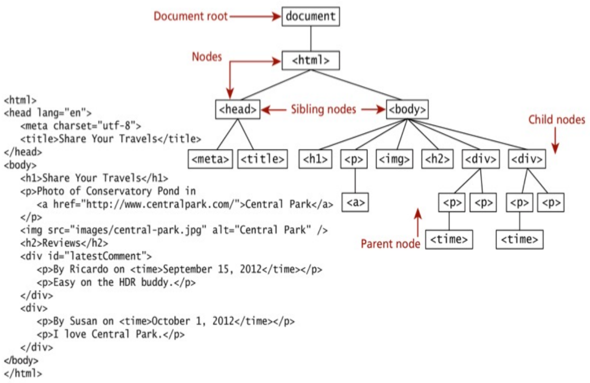
Document Object Model (DOM)

- JavaScript is **almost always** used to interact with the HTML document through Document Object Model.

- According to the W3C, the DOM is a:

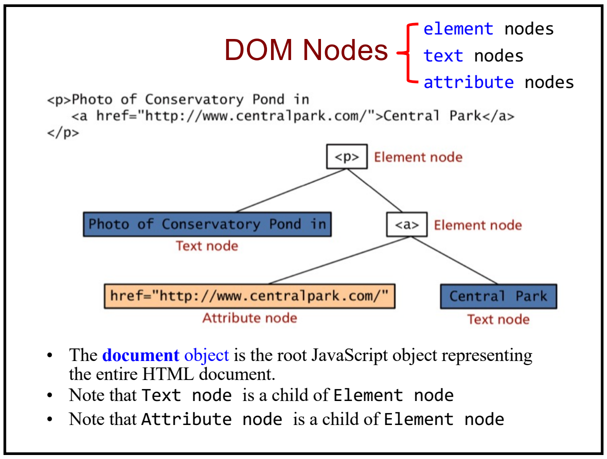
  Platform- and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of documents.

5



```
<html>
<head lang="en">
   <meta charset="utf-8">
   <title>Share Your Travels</title>
</head>
<body>
   <h1>Share Your Travels</h1>
   <p>Photo of Conservatory Pond in
      <a href="http://www.centralpark.com/">Central Park</a>
   </p>
   <img src="images/central-park.jpg" alt="Central Park" />
   <h2>Reviews</h2>
   <div id="latestComment">
      <p>By Ricardo on <time>September 15, 2012</time></p>
      <p>Easy on the HDR buddy.</p>
   </div>
   <div>
      <p>By Susan on <time>October 1, 2012</time></p>
      <p>I love Central Park.</p>
   </div>
</body>
</html>
```

6



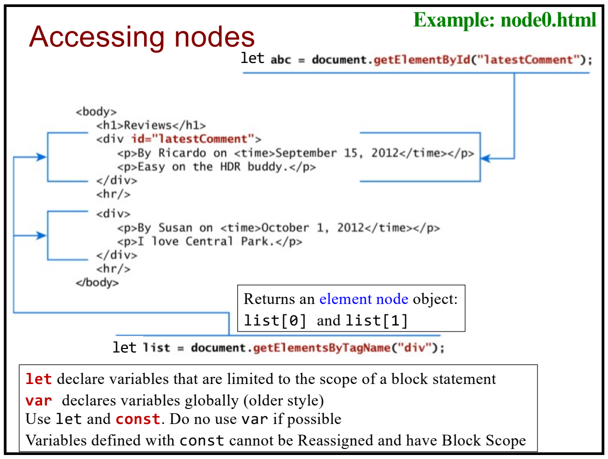
DOM Nodes

- `element` nodes
- `text` nodes
- `attribute` nodes

```
<p>Photo of Conservatory Pond in
   <a href="http://www.centralpark.com/">Central Park</a>
</p>
```

- The **document** object is the root JavaScript object representing the entire HTML document.
- Note that `Text node` is a child of `Element node`
- Note that `Attribute node` is a child of `Element node`

7



Accessing nodes

Example: node0.html

```
let abc = document.getElementById("latestComment");
```

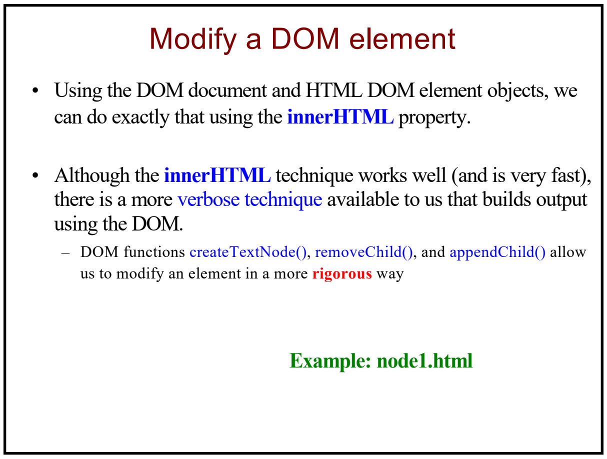
```
<body>
   <h1>Reviews</h1>
   <div id="latestComment">
      <p>By Ricardo on <time>September 15, 2012</time></p>
      <p>Easy on the HDR buddy.</p>
   </div>
   <hr/>

   <div>
      <p>By Susan on <time>October 1, 2012</time></p>
      <p>I love Central Park.</p>
   </div>
   <hr/>
</body>
```

Returns an element node object:
`list[0]` and `list[1]`

```
let list = document.getElementsByTagName("div");
```

`let` declare variables that are limited to the scope of a block statement
`var` declares variables globally (older style)
Use `let` and `const`. Do no use `var` if possible
Variables defined with `const` cannot be Reassigned and have Block Scope

8

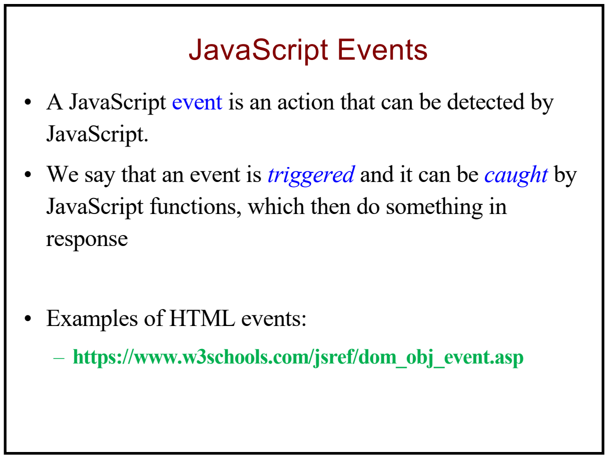


Modify a DOM element

- Using the DOM document and HTML DOM element objects, we can do exactly that using the **innerHTML** property.
- Although the **innerHTML** technique works well (and is very fast), there is a more verbose technique available to us that builds output using the DOM.
  - DOM functions createTextNode(), removeChild(), and appendChild() allow us to modify an element in a more **rigorous** way

Example: node1.html

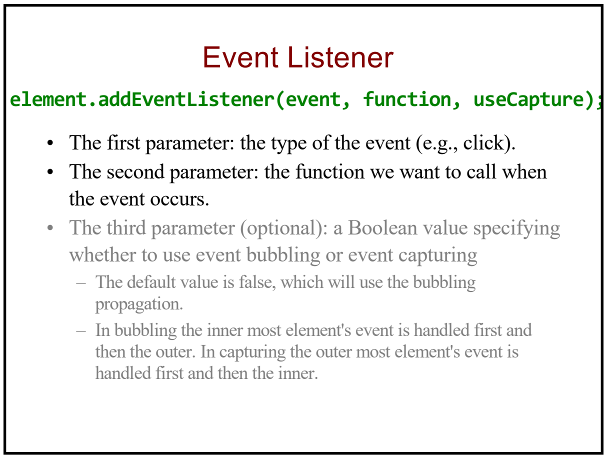
9



JavaScript Events

- A JavaScript event is an action that can be detected by JavaScript.
- We say that an event is *triggered* and it can be *caught* by JavaScript functions, which then do something in response
- Examples of HTML events:
  - **https://www.w3schools.com/jsref/dom_obj_event.asp**

10

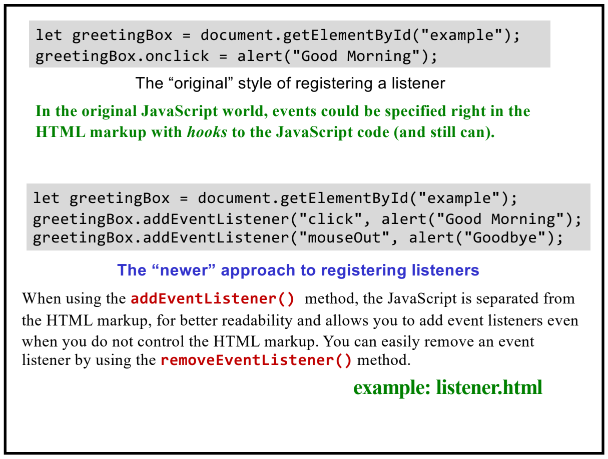


Event Listener

```
element.addEventListener(event, function, useCapture);
```

- The first parameter: the type of the event (e.g., click).
- The second parameter: the function we want to call when the event occurs.
- The third parameter (optional): a Boolean value specifying whether to use event bubbling or event capturing
  - The default value is false, which will use the bubbling propagation.
  - In bubbling the inner most element's event is handled first and then the outer. In capturing the outer most element's event is handled first and then the inner.

11



```
let greetingBox = document.getElementById("example");
greetingBox.onclick = alert("Good Morning");
```

The “original” style of registering a listener

In the original JavaScript world, events could be specified right in the HTML markup with *hooks* to the JavaScript code (and still can).

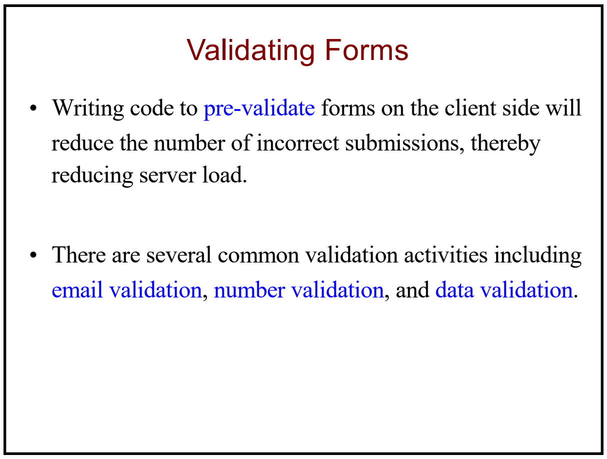
```
let greetingBox = document.getElementById("example");
greetingBox.addEventListener("click", alert("Good Morning");
greetingBox.addEventListener("mouseOut", alert("Goodbye");
```

The “newer” approach to registering listeners

When using the `addEventListener()` method, the JavaScript is separated from the HTML markup, for better readability and allows you to add event listeners even when you do not control the HTML markup. You can easily remove an event listener by using the `removeEventListener()` method.

example: listener.html

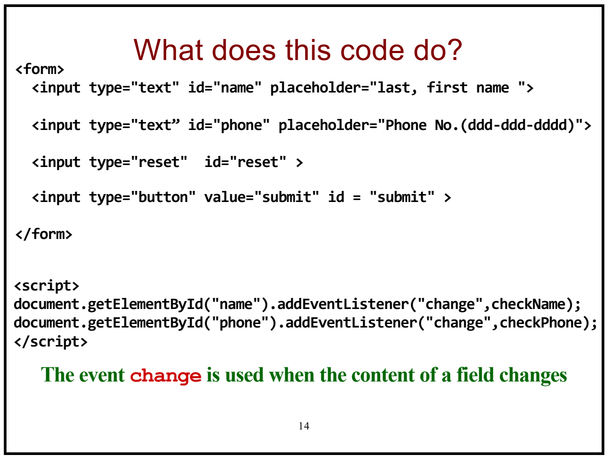
12



Validating Forms

- Writing code to pre-validate forms on the client side will reduce the number of incorrect submissions, thereby reducing server load.
- There are several common validation activities including email validation, number validation, and data validation.

13

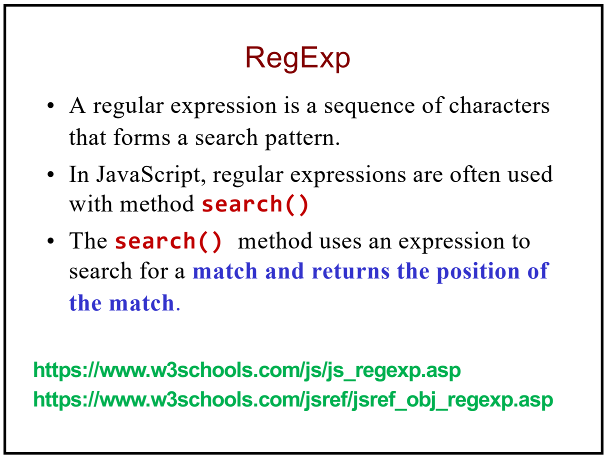


What does this code do?

```
<form>
  <input type="text" id="name" placeholder="last, first name ">

  <input type="text” id="phone" placeholder="Phone No.(ddd-ddd-dddd)">

  <input type="reset"  id="reset" >

  <input type="button" value="submit" id = "submit" >

</form>


<script>
document.getElementById("name").addEventListener("change",checkName);
document.getElementById("phone").addEventListener("change",checkPhone);
</script>
```

The event `change` is used when the content of a field changes

14

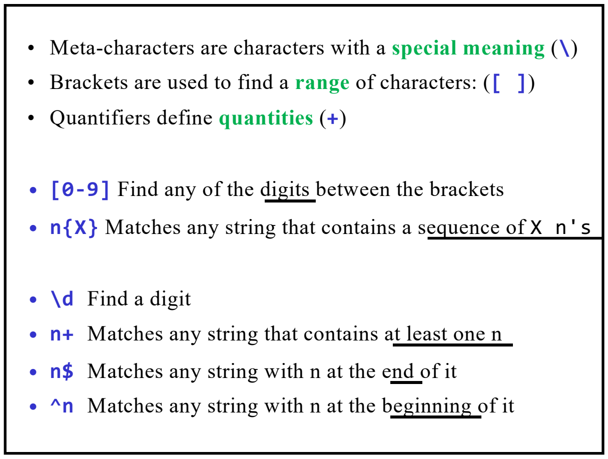


RegExp

- A regular expression is a sequence of characters that forms a search pattern.
- In JavaScript, regular expressions are often used with method `search()`
- The `search()` method uses an expression to search for a **match and returns the position of the match.**

https://www.w3schools.com/js/js_regexp.asp
https://www.w3schools.com/jsref/jsref_obj_regexp.asp

15

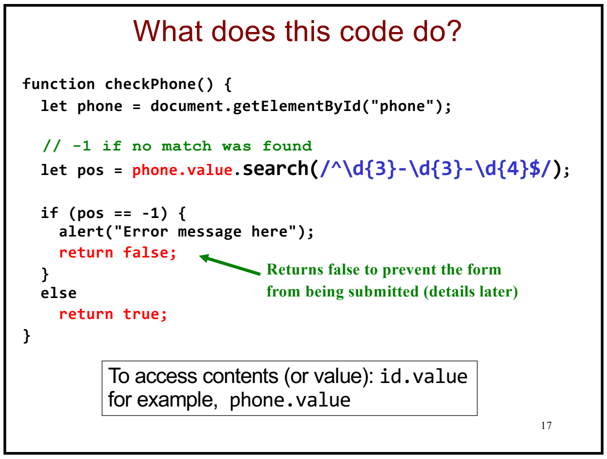


- Meta-characters are characters with a **special meaning** (****)
- Brackets are used to find a **range** of characters: (`[ ]`)
- Quantifiers define **quantities** (**+**)

- `[0-9]` Find any of the digits between the brackets
- `n{X}` Matches any string that contains a sequence of `X n's`

- `\d` Find a digit
- `n+` Matches any string that contains at least one n
- `n$` Matches any string with n at the end of it
- `^n` Matches any string with n at the beginning of it

16



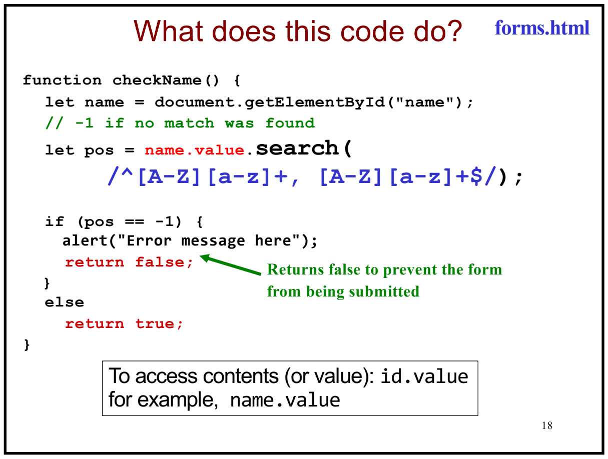
What does this code do?

```
function checkPhone() {
  let phone = document.getElementById("phone");

  // -1 if no match was found
  let pos = phone.value.search(/^\d{3}-\d{3}-\d{4}$/);

  if (pos == -1) {
    alert("Error message here");
    return false;
  }
  else
    return true;
}
```

Returns false to prevent the form from being submitted (details later)

To access contents (or value): `id.value`
for example, `phone.value`

17



What does this code do?

forms.html

```
function checkName() {
  let name = document.getElementById("name");
  // -1 if no match was found
  let pos = name.value.search(
       /^[A-Z][a-z]+, [A-Z][a-z]+$/);

  if (pos == -1) {
    alert("Error message here");
    return false;
  }
  else
    return true;
}
```

Returns false to prevent the form from being submitted

To access contents (or value): `id.value`
for example, `name.value`

18

What does this code do?

validation.html

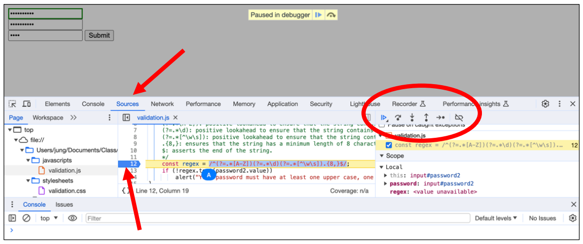
```
<form id="myForm">

<input type="password" id="password0"
             required minlength="8"
             pattern="^(?=.*[A-Z])(?=.*\d)(?=.*[\W_]).{8,}$"
             placeholder="password0"><br>
<button type="submit">Submit</button>

</form>
```

```
<style>
#password0:invalid{
    border: 2px solid red;
}
#password0:valid{
    border: 2px solid green;
}
</style>
```

https://developer.mozilla.org/en-US/docs/Learn/Forms/Form_validation
https://www.w3schools.com/js/js_validation_api.asp

19

Browser inspect - debugger



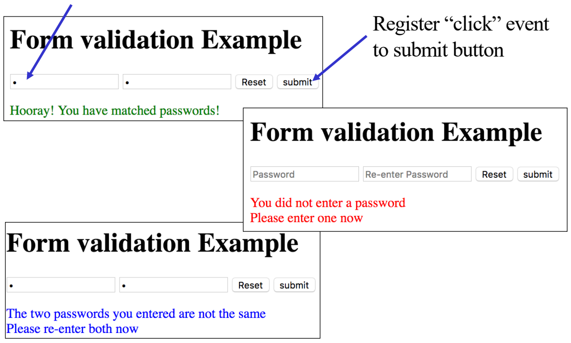
```
 7   (?=.*\d): positive lookahead to ensure that the string contains
 8   (?=.*[^\w\s]): positive lookahead to ensure that the string cont
 9   .{8,}: ensures that the string has a minimum length of 8 charact
10   $: asserts the end of the string.
11   */
12   const regex = /^(?=.*[A-Z])(?=.*\d)(?=.*[^\w\s]).{8,}$/;
13   if (!regex.test(password2.value))
14       alert("Your password must have at least one upper case, one
```

```
☑ const regex = /^(?=.*[A-Z])(?=.*\d)(?=.*[^\w\s])…   12
▼ Local
  ▶ this: input#password2
  ▶ password: input#password2
    regex: <value unavailable>
```

20

Exercise

```
<input type = "password" >
```



Register “click” event to submit button

Form validation Example

• | • | Reset | submit

Hooray! You have matched passwords!

Form validation Example

Password | Re-enter Password | Reset | submit

You did not enter a password
Please enter one now

Form validation Example

• | • | Reset | submit

The two passwords you entered are not the same
Please re-enter both now

21