JavaScript and jQuery Course

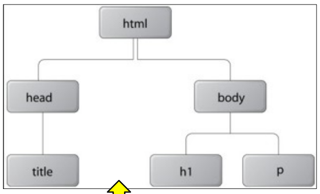
Session 08 DOM

DOM

DOM – hierarchical tree representation of document

- collection of *nodes* that represent the web page
- JavaScript can modify the DOM

- descendants
- parents
- children
- siblings



nodes

DOM – commonly used node types

`document`	Only has element nodes as child node
`element`	Can have element, text, comment node as a child node
`attr`	Attribute of an element Can have text node as a child node
`text`	Text for an element, attribute, comment No child nodes

Node Properties

`nodeValue` ←	Returns the text for a text, attribute, comment node
`parentNode`	Returns the parent of that node
`childNodes`	Returns an array of node objects
`firstChild` ←	Returns a node object for the first child node
`lastChild`	Returns a node object for the last child node
`nextElementSibling`	Returns node object for next sibling

If no node exists, returns `null`

Methods of document interface

getElementById(id) ←

returns the element object that matches the id

getElementsByTagName(tag_name)

Elements = plural

returns an array of elements that matches the tag_name value

```
<input name="fname" id="fname" type="text">
<input name="lname" id="lname" type="text">
```

all <input> elements

Methods of document interface

getElementsByTagName(tag_name)

returns an array of elements that matches the tag_name value

```
<input name="fname" id="fname" type="text">
<input name="lname" id="lname" type="text">
```

```
// clear out text box values
  for (var i = 0;  i < document.getElementsByTagName('input').length;  i++) {

      document.getElementsByTagName('input')[i].value = " ";

  }
```

array

Methods of document interface

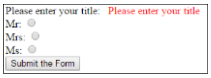
getElementsByName(name_value)

returns an array of elements that matches the name_value value

Elements = plural

```
<input name="title" type="radio" id="Mr" value="Mr">
<input name="title" type="radio" id="Mrs" value="Mrs">
<input name="title" type="radio" id="Ms" value="Ms">
```

all elements with name attribute
–used by the web server



Please enter your title: Please enter your title
Mr: ○
Mrs: ○
Ms: ○

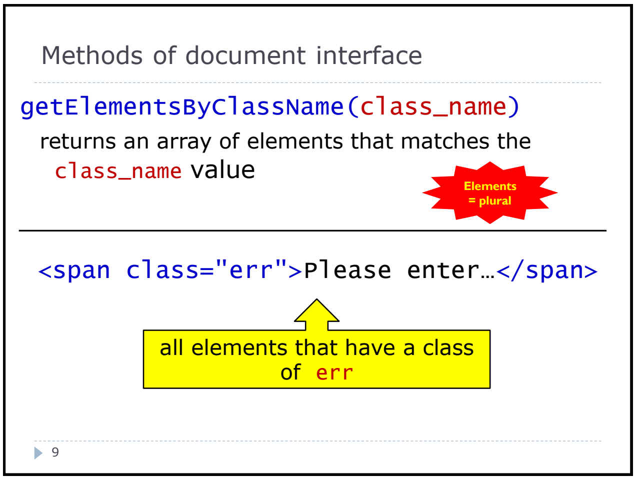
Methods of document interface

getElementsByName(name_value)

returns an array of elements that matches the name_value value

```
// determine if any title was checked

for (var i = 0;   i < document.getElementsByName('title').length;    i++) {

    if (document.getElementsByName('title')[i].checked) {
           titleOK = true;
    }

}
```

array



Methods of document interface

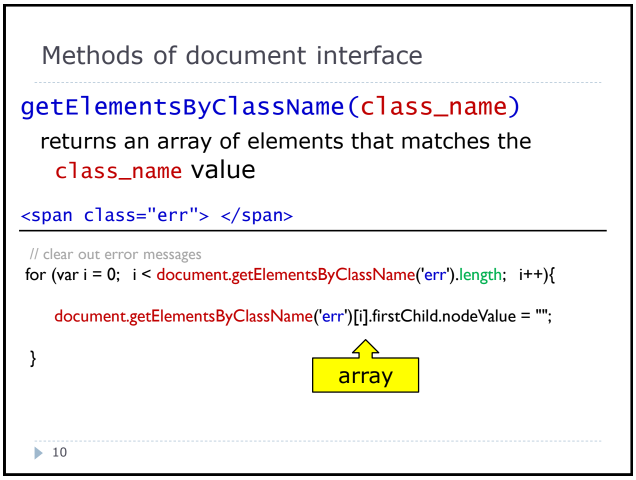
`getElementsByClassName(class_name)`

returns an array of elements that matches the `class_name` value

Elements = plural

```
<span class="err">Please enter…</span>
```

all elements that have a class of `err`



Methods of document interface

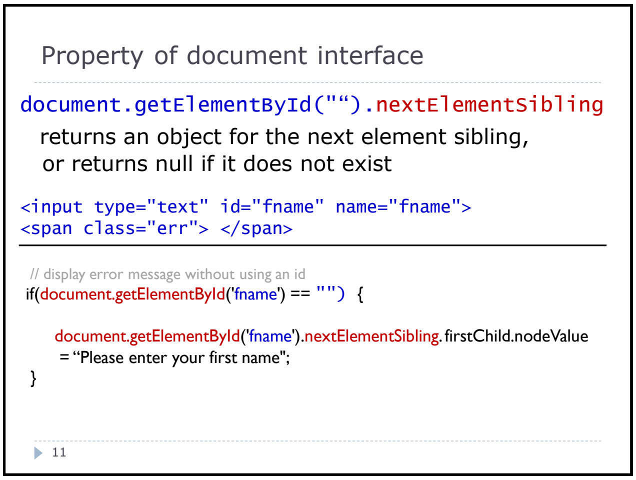
`getElementsByClassName(class_name)`

returns an array of elements that matches the `class_name` value

```
<span class="err"> </span>
```

```
// clear out error messages
for (var i = 0;  i < document.getElementsByClassName('err').length;  i++){

    document.getElementsByClassName('err')[i].firstChild.nodeValue = "";

}
```

array



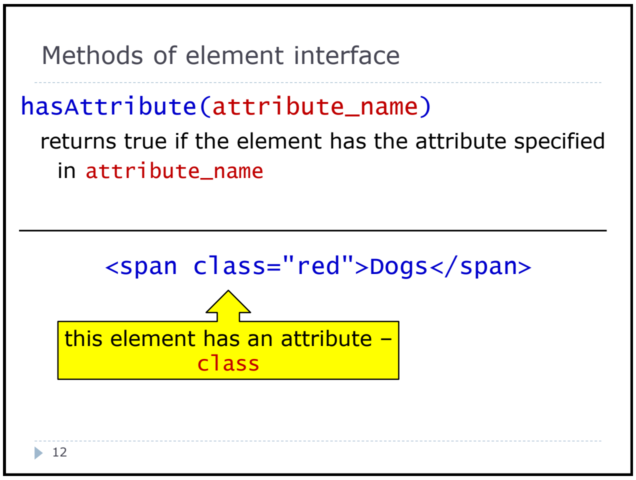
Property of document interface

`document.getElementById("“).nextElementSibling`

returns an object for the next element sibling, or returns null if it does not exist

```
<input type="text" id="fname" name="fname">
<span class="err"> </span>
```

```
// display error message without using an id
if(document.getElementById('fname') == "") {

    document.getElementById('fname').nextElementSibling.firstChild.nodeValue
    = “Please enter your first name";
}
```



Methods of element interface

`hasAttribute(attribute_name)`

returns true if the element has the attribute specified in `attribute_name`

```
<span class="red">Dogs</span>
```

this element has an attribute – `class`

Methods of element interface

```
getAttribute(attribute_name)
```

returns the value of the attribute specified in `attribute_name`

```
<span class="red">Dogs</span>
```

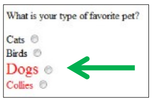
this attribute value is red

Methods of element interface

```
setAttribute(attribute_name, value)
```

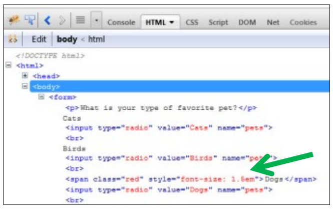
sets the attribute to the value,
or sets a new attribute

```
<span class="red">Dogs</span>
```



What is your type of favorite pet?

Cats ○
Birds ○
Dogs ○
Collies ○



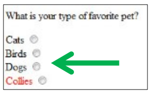
```
<!DOCTYPE html>
<html>
  <head>
  <body>
    <form>
      <p>What is your type of favorite pet?</p>
      Cats
      <input type="radio" value="Cats" name="pets">
      <br>
      Birds
      <input type="radio" value="Birds" name="pets">
      <br>
      <span class="red" style="font-size: 1.5em">Dogs</span>
      <input type="radio" value="Dogs" name="pets">
      <br>
```

Methods of element interface

```
removeAttribute(attribute_name)
```

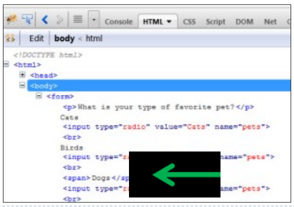
removes the attribute

```
<span class="red">Dogs</span>
```



What is your type of favorite pet?

Cats ○
Birds ○
Dogs ○
Collies ○



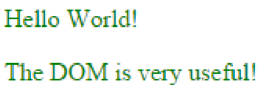
```
<!DOCTYPE html>
<html>
  <head>
  <body>
    <form>
      <p>What is your type of favorite pet?</p>
      Cats
      <input type="radio" value="Cats" name="pets">
      <br>
      Birds
      <input type="r                name="pets">
      <br>
      <span>Dogs</sp
      <input type="r                name="pets">
      <br>
```

Methods of document interface - not in text
querySelector(), querySelectorAll()

```
// Access the first .myClass class name
document.querySelector('.myClass');

// Return a NodeList of all instances of .myClass
document.querySelectorAll('.myClass');

// Accessb the myID id
document.querySelector('#myID');

// Return a NodeList of all 'div' instances
document.querySelectorAll('div');

//Access the last list Node of .someList unordered list
document.querySelector('ul.someList li:last-child');
```

Methods of document interface - not in text querySelectorAll()

```
<body>

<p>Hello World!</p>

<p>The DOM is very useful!</p>


<script>
   var ps = document.querySelectorAll('p');

   for (var i = 0; i < ps.length; ++i) {
     ps[i].style.color = "green";
   }
</script>

</body>
```



Hello World!

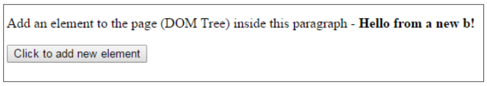
The DOM is very useful!

Methods of document interface - not in text createElement(), createTextNode(), appendChild()

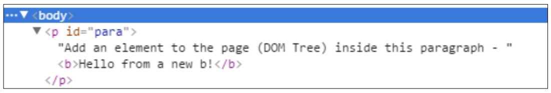
```
// create a new div element
var newB = document.createElement("b");

// and give it some content
var newContent =
      document.createTextNode("Hello from a new b!");

//add the text node to the newly created <b>
newB.appendChild(newContent);

// add the newly created element and its content
// into the DOM
Var thePara = document.getElementById("para");

thePara.appendChild(newB);
```

Methods of document interface - not in text createElement(), createTextNode(), appendChild()

```
<p id="para">
Add an element to the page (DOM Tree) inside this paragraph –
</p>
```



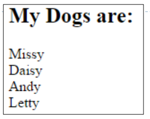
Add an element to the page (DOM Tree) inside this paragraph - **Hello from a new b!**



```
… ▼ <body>
  ▼ <p id="para">
      "Add an element to the page (DOM Tree) inside this paragraph - "
      <b>Hello from a new b!</b>
    </p>
```

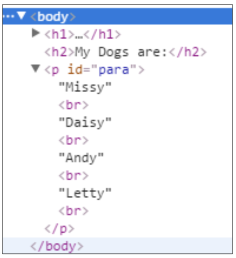
Methods of document interface - to loop through array createElement(), createTextNode(), appendChild()

```
 var myPara  = document.getElementById('para');
// loop through the array
//  create elements, nodes, append them
for (var i = 0; i < myArray.length; i++) {

var text = document.createTextNode(myArray[i]);

var br   = document.createElement('br');

myPara.appendChild(text);

myPara.appendChild(br);
}
```



My Dogs are:

Missy
Daisy
Andy
Letty



```
… ▼ <body>
    ▶ <h1>…</h1>
      <h2>My Dogs are:</h2>
    ▼ <p id="para">
        "Missy"
        <br>
        "Daisy"
        <br>
        "Andy"
        <br>
        "Letty"
        <br>
      </p>
  </body>
```