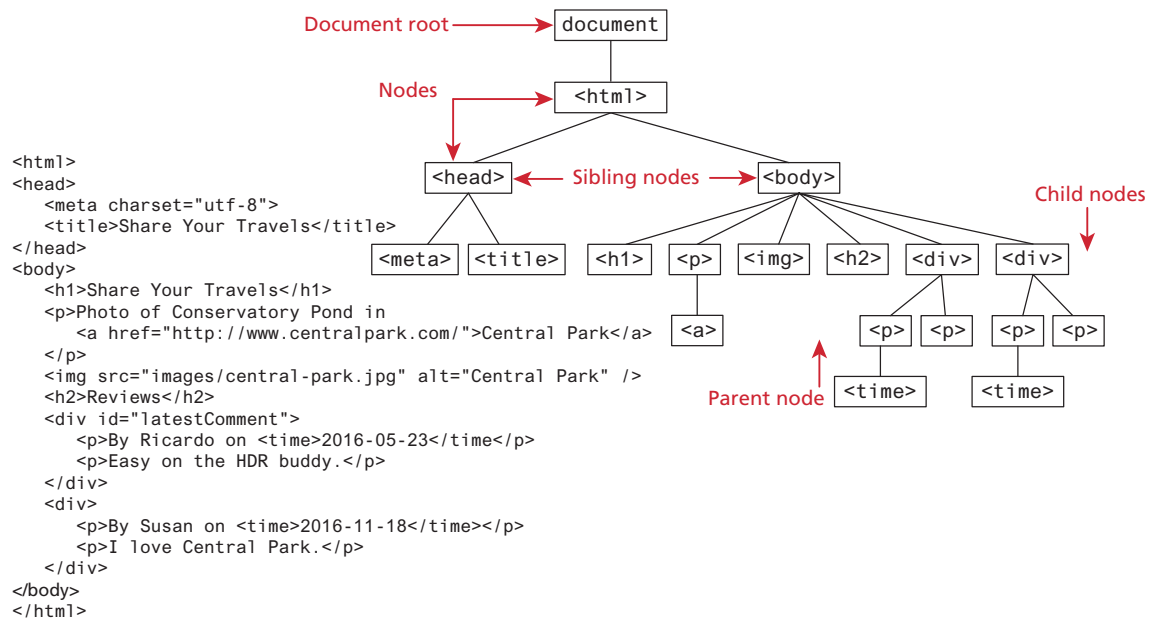


The DOM

47

The Document Object Model (DOM)

Overview

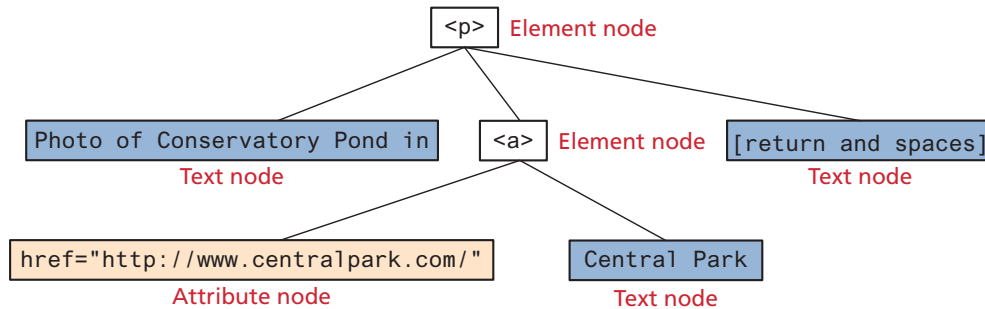


48

The Document Object Model (DOM)

Nodes and NodeLists

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



49

The Document Object Model (DOM)

Document Object

The DOM document object is the root JavaScript object representing the entire HTML document

```
// retrieve the URL of the current page  
  
var a = document.URL;  
  
// retrieve the page encoding, for example ISO-8859-1  
  
var b = document.inputEncoding;
```

50

The Document Object Model (DOM)

Selection Methods

Classic

- `getElementById()`
- `getElementsByTagName()`
- `getElementsByClassName()`

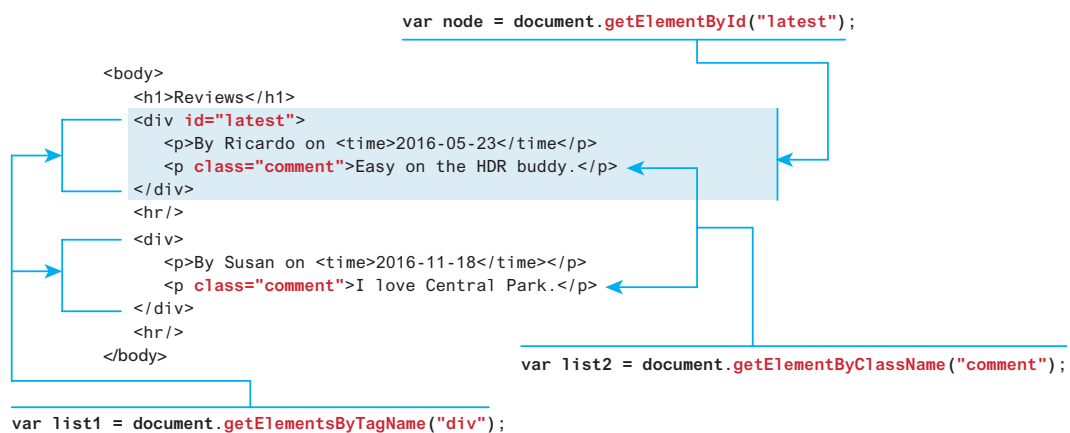
Newer

- `querySelector()` and
- `querySelectorAll()`

51

The Document Object Model (DOM)

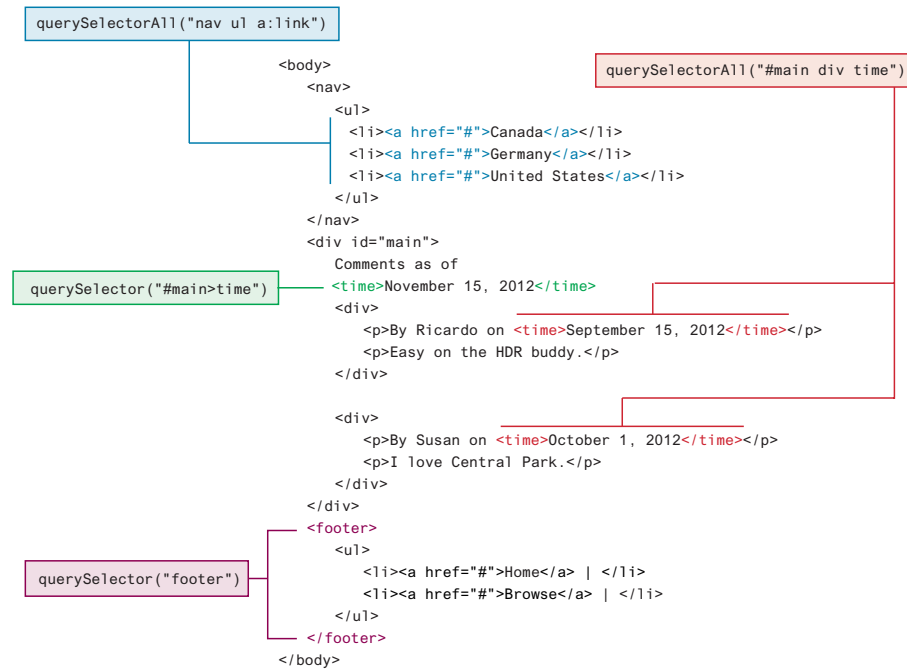
Selection Methods



52

The Document Object Model (DOM)

Query Selector



53

The Document Object Model (DOM)

Element Node Object

Element Node object represents an HTML element in the hierarchy, contained between the opening `<>` and closing `</>` tags for this element. Every node has

- `classList`
- `className`
- `id`
- `innerHTML`
- `Style`
- `tagName`

54

The Document Object Model (DOM)

More common (not universal) properties

- href
- name
- src
- value

55

Modifying the DOM

56

Modifying the DOM

Changing an Element's Style

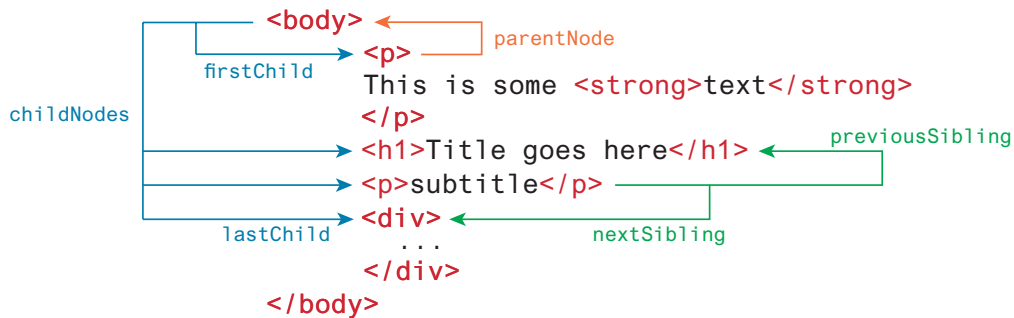
```
<style>
  .box {
    margin: 2em; padding: 0;
    border: solid 1pt black;
  }
  .yellowish { background-color: #EFE63F; }
  .hide { display: none; }
</style>
<main>
  <div class="box">
    ...
  </div>
</main>
```

<pre>var node = document.querySelector("main div");</pre>		Equivalent to:	
1	<pre>node.className = "yellowish";</pre> <p>This replaces the existing class specification with this one. Thus the <div> no longer has the box class</p>	1	<pre><div class="yellowish"></pre>
2	<pre>node.classList.remove("yellowish");</pre> <p>Removes the specified class specification and adds the box class</p>	2	<pre><div class=""> <div class="box"></pre>
3	<pre>node.classList.add("yellowish");</pre> <p>Adds a new class to the existing class specification</p>	3	<pre><div class="box yellowish"></pre>
4	<pre>node.classList.toggle("hide");</pre> <p>If it isn't in the class specification, then add it</p>	4	<pre><div class="box yellowish hide"></pre>
5	<pre>node.classList.toggle("hide");</pre> <p>If it is in the class specification, then remove it</p>	5	<pre><div class="box yellowish"></pre>

57

Modifying the DOM

Meet the family



58

Modifying the DOM

Changing an Element's Content

```
document.getElementById("here").innerHTML =  
    "foo<em>bar</em>";
```

1. Create a new Node
2. Add Node to an element in the DOM

59

Events

60

Events

JavaScript event is an action that can be detected by JavaScript

- Many of them are initiated by user actions
- some are generated by the browser itself.

We say that an event is *triggered* and then it is *handled* by JavaScript functions

61

Events

Event-Handling Approaches – Inline Hook

HTML document using the inline hooks

```
...  
<script type="text/javascript" src="inline.js"></script>  
...  
<form name='mainForm' onsubmit="validate(this);">  
  <input name="name" type="text"  
    onchange="check(this);"   
    onfocus="highlight(this, true);"   
    onblur="highlight(this, false);">  
  <input name="email" type="text"  
    onchange="check(this);"   
    onfocus="highlight(this, true);"   
    onblur="highlight(this, false);">  
  <input type="submit"  
    onclick="function (e) {  
      ...  
    }">  
...
```

inline.js

```
function validate(node) {  
  ...  
}  
function check(node) {  
  ...  
}  
function highlight(node) {  
  ...  
}
```

Notice that you can define an entire event handling function within the markup. This is NOT recommended!

The diagram illustrates the 'Inline Hook' approach to event handling. On the left, an HTML document snippet shows a form with three input fields and a submit button. Each input field has inline event handlers for 'onchange', 'onfocus', and 'onblur', all of which call functions named 'check', 'highlight', or 'validate'. The submit button has an 'onclick' handler that calls an anonymous function. A green arrow points from the 'inline.js' script tag in the HTML to a box on the right containing the definitions of these functions: 'validate', 'check', and 'highlight'. Red arrows point from each inline event handler in the HTML to its corresponding function definition in the 'inline.js' box. A note at the bottom right states: 'Notice that you can define an entire event handling function within the markup. This is NOT recommended!'.

62

Events

Event-Handling Approaches – Event Property Approach

```
var myButton = document.getElementById('example');  
myButton.onclick = alert('some message');
```

63

Events

Event-Handling Approaches – Event Listener Approach

```
var myButton = document.getElementById('example');  
  
myButton.addEventListener('click', alert('some  
message'));  
  
myButton.addEventListener('mouseout', funcName);
```

64

Events

Event-Handling Approaches – Event Listener Approach (anon function)

```
myButton.addEventListener('click', function() {  
    var d = new Date();  
    alert("You clicked this on "+ d.toString());  
});
```

65

Events

Event Object

When an event is triggered, the browser will construct an event object that contains information about the event.

```
div.addEventListener('click', function(e) {  
    // find out where the user clicked  
    var x = e.clientX;  
    ...
```

66

Events

Event Object

- bubbles Indicates whether the event bubbles up through the DOM
- cancelable Indicates whether the event can be cancelled
- target The object that generated (or dispatched) the event
- type The type of the event (see next slides)

67

Event Types

Mouse Events

- click The mouse was clicked on an element
- dblclick The mouse was double clicked on an element
- mousedown The mouse was pressed down over an element
- mouseup The mouse was released over an element
- mouseover The mouse was moved (not clicked) over an element
- mouseout The mouse was moved off of an element
- mousemove The mouse was moved while over an element

68

Event Types

Keyboard Events

- **keydown** The user is pressing a key (this happens first)
- **keypress** The user presses a key (this happens after keydown)
- **keyup** The user releases a key that was down (this happens last)

69

Event Types

Touch Events

Touch events are a new category of events that can be triggered by devices with touch screens

Limited Browser support (2017)

The different events (e.g., touchstart, touchmove, and touchend) are analogous to some of the mouse events (mousedown, mousemove, and mouseup).

70

Event Types

Form Events

- Blur
- Change
- Focus
- Reset
- select
- Submit

71

Event Types

Frame Events

- abort - An object was stopped from loading
- error - An object or image did not properly load
- load - When a document or object has been loaded
- resize - The document view was resized
- scroll - The document view was scrolled
- unload - The document has unloaded

72

Forms

73

Forms

Validating a Submitted Form

- Field Validation
- Number Validation
- Other (non JavaScript) Form validation reminder

74

Forms

Submitting Forms

```
var formExample =  
document.getElementById("loginForm");  
  
formExample.submit();
```