



---

---

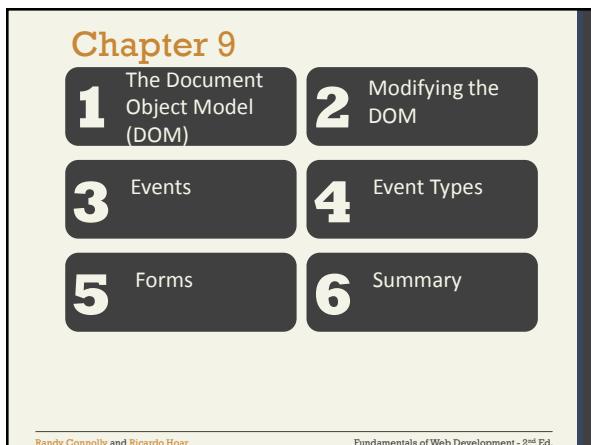
---

---

---

---

---



---

---

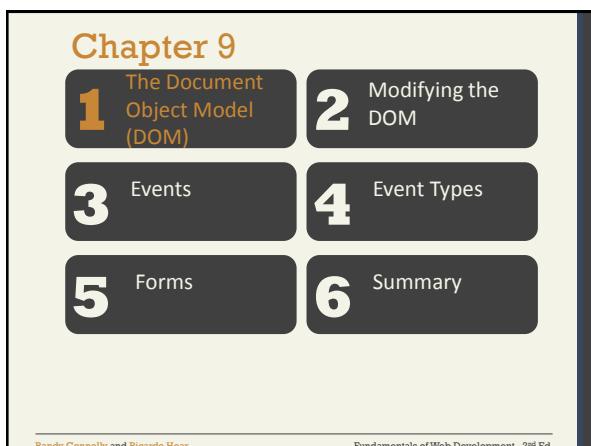
---

---

---

---

---



---

---

---

---

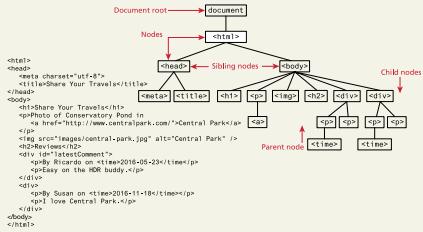
---

---

---

## The Document Object Model (DOM)

## Overview



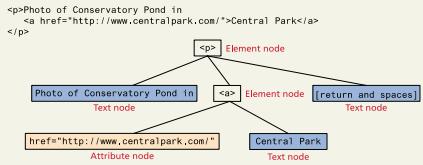
---

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed

## The Document Object Model (DOM)

## Nodes and NodeLists



---

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed

## 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;
```

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed

## The Document Object Model (DOM)

### Selection Methods

#### Classic

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

#### Newer

- querySelector() and
- querySelectorAll()

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed.

---



---



---



---



---



---



---



---



---



---



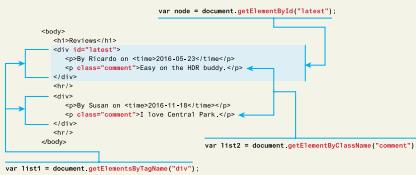
---



---

## The Document Object Model (DOM)

### Selection Methods



Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed.

---



---



---



---



---



---



---



---



---



---



---



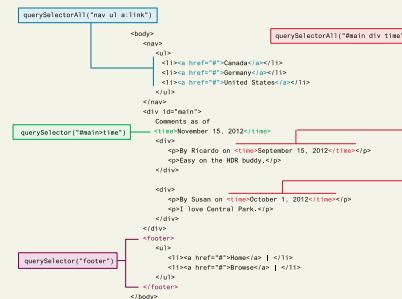
---



---

## The Document Object Model (DOM)

### Query Selector



Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed.

---



---



---



---



---



---



---



---



---



---



---



---



---



---

## 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

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed.

---



---



---



---



---



---



---



---



---



---

## The Document Object Model (DOM)

### More common (not universal) properties

- href
- name
- src
- value

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed.

---



---



---



---



---



---



---



---



---



---



---

## Chapter 9

**1** The Document Object Model (DOM)

**2** Modifying the DOM

**3** Events

**4** Event Types

**5** Forms

**6** Summary

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed.

---



---



---



---



---



---



---



---



---



---



---

# Modifying the DOM

## Changing an Element's Style

```
/*style
.box {
    border: 2px solid black;
    border: solid 1px black;
}
.yellowish { background-color: #FFFF00; }
#title { display: none; }
#content { margin-top: 10px; }
```

var box = document.querySelector("main div");  
box.classList.add("yellowish");  
node.classList.remove("yellowish");  
node.classList.toggle("yellowish");  
node.classList.toggle("box");  
node.classList.add("yellowish");  
node.classList.toggle("hide");  
node.classList.toggle("hide");

This replaces the existing class specification with the one, thus the `color` no longer has the box style applied.

Removes the specified class specification and adds the box class.

Adds a new class to the existing element.

If it's not in the class specification, then add it.

If it is in the class specification, then remove it.

Equivalent to:

- ① `<div class="yellowish">`
- ② `<div class="">`
- ③ `<div class="box yellowish">`
- ④ `<div class="box yellowish hide">`
- ⑤ `<div class="box yellowish hide">`

---

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed

## Modifying the DOM

## Meet the family

The diagram shows the DOM structure of a `<body>` element. It contains several child nodes:

- `<p>`: A sibling node.
- `<strong>text</strong>`: A child node.
- `<h1>Title goes here</h1>`: A child node.
- `<p>subtitle</p>`: A sibling node.
- `<div>`: A child node.

Annotations provide context:

- `parentNode` points to the `<body>` element.
- `firstChild` points to the first child node, `<p>`.
- `lastChild` points to the last child node, `<div>`.
- `nextSibling` points to the sibling node following the `<strong>` node.
- `previousSibling` points to the sibling node preceding the `<h1>` node.
- `...  
</div>` indicates the continuation of the `<div>` node's content.

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed

## Modifying the DOM

## Changing an Element's Content

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

---

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed

# Modifying the DOM

## Creating DOM elements

- 1 Create a new text node  

```
var text = document.createTextNode("this is dynamic");
```
  - 2 Create a new empty <p> element  

```
var p = document.createElement("p");
```
  - 3 Add the text node to new <p> element  

```
p.appendChild(text);
```
  - 4 Add the <p> element to the <div>  

```
var first = document.getElementById("first");
first.appendChild(p);
```

---

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed

## Modifying the DOM

## Creating DOM elements

- The diagram illustrates the state of the DOM tree after step 4. A green box contains the original code: `var first = document.getElementById("first"); first.appendChild(p);`. To its right, a blue box shows the resulting DOM structure. The root element is a red `<div>`. It contains three children: a blue `<h1>` node with the text "DOM Example", a blue `<p>` node with the text "Existing element", and a blue `<p>` node with the text "this is dynamic".

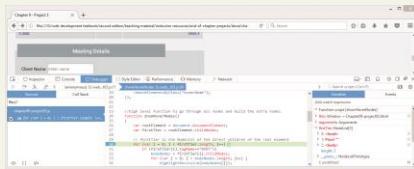
---

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed

## Modifying the DOM

## Tools - Debuggers



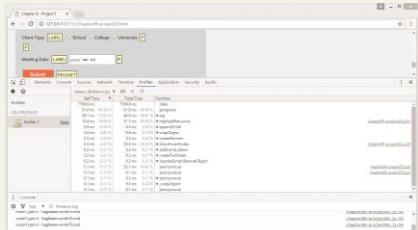
---

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed

## Modifying the DOM

## Tools – Performance checkers



---

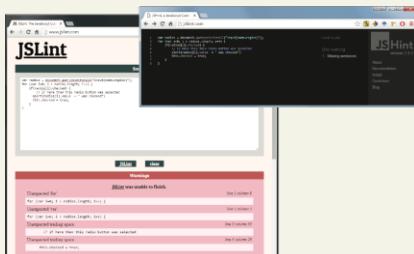
Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed



# Modifying the DOM

## Tools – Linters



---

Page 1 of 1

Environ Biol Fish (2008) 81:1–10



Chapter 9

# 1 The Document Object Model (DOM)

## 2 Modifying the DOM

# 3 Events

## 4 Event Types

5 Forms

**6** Summary

---

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed



## 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

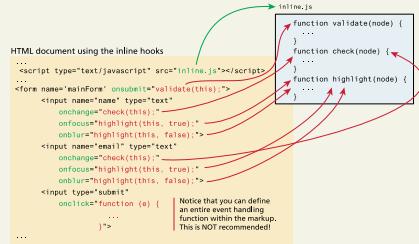
---

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed

## Events

## Event-Handling Approaches – Inline Hook



Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed

## Events

## Event-Handling Approaches – Event Property Approach

```
var myButton = document.getElementById('example');

myButton.onclick = alert('some message');
```

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed

## Events

## Event-Handling Approaches – Event Listener Approach

```
var myButton = document.getElementById('example');

myButton.addEventListener('click', alert('some
message'));

myButton.addEventListener('mouseout', funcName);
```

---

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed

## Events

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

---

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed

## 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;  
    ...
```

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

## 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 Section 9.4)

---

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed

## Chapter 9

# 1 The Document Object Model (DOM)

## 2 Modifying the DOM

3 Events

# 4 Event Types

# 5 Forms

6 Summary

---

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed

## 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

---

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

## 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)

---

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed

## 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).

---

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed

## Event Types

## Form Events

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

---

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

## 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

---

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed

---

---

---

---

---

---

---

---

## Chapter 9

# 1 The Document Object Model (DOM)

# 2 Modifying the DOM

## 3 Events

## 4 Event Types

# 5 Forms

6 Summary

---

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed

---

---

---

---

---

---

---

---

---

## Forms

## Responding to Form Movement Events



```

    // This function is going to get called every time the focus or blur events are
    // triggered on one of our three input elements.
    function onInputFocusOrBlur() {
        if (this.type === "text") {
            // Set the class="highlighted" style whenever the input has focus.
            this.classList.add("highlighted");
        } else if (this.type === "button") {
            // Set the class="highlighted" style whenever the button has focus.
            this.classList.add("highlighted");
        }
    }

    // Add the event listener immediately after the DOM is loaded.
    window.addEventListener("load", () => {
        var selector = "[type='text'], [type='button']";
        var elems = document.querySelectorAll(selector);
        elems.forEach((elem) => {
            elem.addEventListener("focus", onInputFocusOrBlur);
            elem.addEventListener("blur", onInputFocusOrBlur);
        });
    });
}

```

Now we can use the `this.classList` property instead of specifying the `highlighted` class name, because the `onInputFocusOrBlur` function overwrites all other definitions.

Selects the fields that will change.

Assigns the `onInputFocusOrBlur` function to each element in the array corresponding upon whether it has the focus.

---

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed

---

---

---

---

---

---

---

---

## Forms

Responding to Form Changes Events

```
// Depending on the state of the radio buttons
// and checkboxes, we can determine what
// the total amount of the bill is.
var total = document.getElementById("total");
var select = document.getElementById("select");
var selectValue = true;
var radioButtons = document.querySelectorAll("input[type='radio']");
var checkboxes = document.querySelectorAll("input[type='checkbox']");
for (var i = 0; i < radioButtons.length; i++) {
    // If all are off, then multiply the select
    // by the amount of the radio button.
    if (radioButtons[i].checked) {
        total.value = radioButtons[i].value;
        break;
    }
}
function onFormChange() {
    var checked = false;
    checkboxes.forEach(function(checkbox) {
        if (checkbox.checked) {
            checked = true;
        }
    });
    if (checked) {
        total.value = "Total Amount: " + "25";
    } else {
        total.value = "Total Amount: " + "15";
    }
}
addEvent(select, "change", onFormChange);
addEvent(radioButtons, "change", onFormChange);
addEvent(checkboxes, "change", onFormChange);
else if (checkboxes[0].checked) {
    total.value = "Total Amount: " + "15";
    total.value = "Total Amount: " + "15";
    total.value = "Total Amount: " + "15";
    addEvent(select, "change", onFormChange);
    addEvent(radioButtons, "change", onFormChange);
    addEvent(checkboxes, "change", onFormChange);
}
else if (checkboxes[1].checked) {
    total.value = "Total Amount: " + "25";
    addEvent(select, "change", onFormChange);
    addEvent(radioButtons, "change", onFormChange);
    addEvent(checkboxes, "change", onFormChange);
}
}
else if (checkboxes[2].checked) {
    total.value = "Total Amount: " + "35";
    addEvent(select, "change", onFormChange);
    addEvent(radioButtons, "change", onFormChange);
    addEvent(checkboxes, "change", onFormChange);
}
}
}
function addEvent(element, event, functionToCall) {
    if (element.addEventListener) {
        element.addEventListener(event, functionToCall);
    } else if (element.attachEvent) {
        element.attachEvent("on" + event, functionToCall);
    } else {
        element["on" + event] = functionToCall;
    }
}

```

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed.

## Forms

Validating a Submitted Form

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

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed.

## Forms

Submitting Forms

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

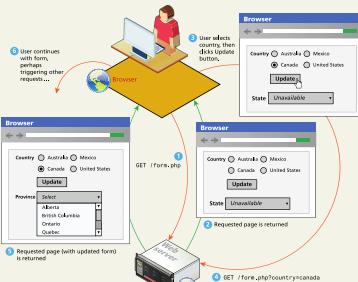
formExample.submit();
```

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed.

## AJAX

First the Normal Request Response Loop

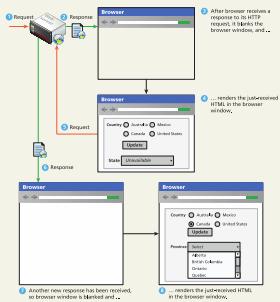


Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed.

## AJAX

The problem

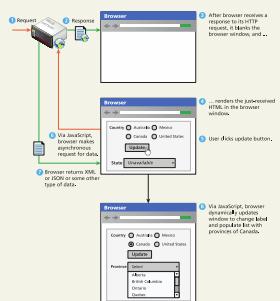


Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed.

## AJAX

The solution



Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed.

## Chapter 9

- 1 The Document Object Model (DOM)
  - 2 Modifying the DOM
  - 3 Events
  - 4 Event Types
  - 5 Forms
  - 6 Summary

---

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed

## **Summary**

## Key Terms

- blur
  - Document Object Model
    - event delegation
    - event handler
    - event listener
    - event object
    - event propagation
    - event type
    - focus
  - (DOM)
  - document root
  - DOM document object
    - node
    - selection methods
  - DOM tree
  - element node
  - event
    - form events
    - frame events
    - keyboard events
    - linter
    - mouse events
    - touch events

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed

## Summary

## Questions?

---

Randy Connolly and Ricardo Hoar

Fundamentals of Web Development - 2<sup>nd</sup> Ed

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---