

The background of the slide is a spiral-bound notebook with a light beige, textured cover. The spiral binding is visible on the left side, with the metal wire looping through a series of holes. The text is centered on the right side of the notebook cover.

# DOM and Scripting

CSE 264

J. Femister

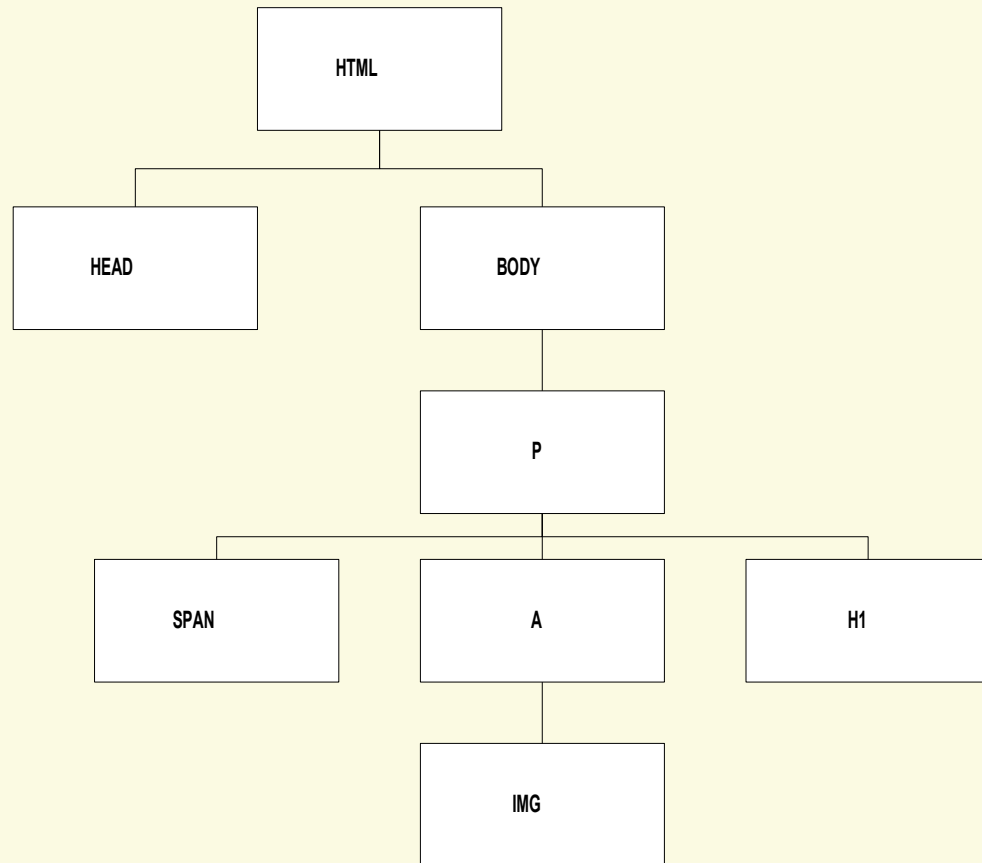
# Document Object Model (DOM)

---

- ✓ Exposes all elements in an HTML document to scripting
- ✓ Dynamic altering/reflowing of the document
- ✓ Animation
- ✓ Interaction - field validation

# Document Object Model

---



A silver metal spiral binding is visible on the left side of the page, looping through a series of holes.

# We need to know how to:

---

- ✓ Trap and respond to events.
- ✓ Modify the document through the DOM.

# HTML Event Model

---

- ✓ Two main features : event bubbling and default actions
- ✓ Events occur when:
  - User interacts with the page
  - Document's state changes

# Event Model

---

## ✓ Event Bubbling

- Simplifies writing of event handlers
- Events “bubble” up through the document hierarchy

## ✓ Default Actions

- can be overwritten at any level in the hierarchy

# Event Binding

---

## ✓ Event Attributes:

### – Example:

- `<input type="button" value="Click Here" onclick="foo()"/>`

# Event Binding

---

- ✓ “ubobtrusive Javascript” (preferred)
- ✓ Event Properties:
  - function clicked() { ..... }
  - document.getElementById("myElement").onclick = clicked;
- ✓ Event Listeners (DOM Level 2)
  - document.getElementById("myElement").  
addEventListener("click", clicked, false /\*  
useCapture \*/);
  - Allows more than one handler



# Events – Device Dependent

---

- ✓ mousedown
- ✓ mouseup
- ✓ mousemove
- ✓ keydown
- ✓ keyup
- ✓ keypress

# Events – Device Independent

---

- ✓ click – Triggered by mouse or keyboard
- ✓ textinput
  - Alternative to keypress
  - typing, pasting

# Events – User Interface

---

## ✓ Higher Level

- focus
- blur
- change
- submit

# Events – State-change

---

- ✓ Not triggered by user
  - load
  - unload

# Events – API Specific

---

## ✓ HTML5

- Drag and Drop
- Video
- Audio

# HTML Event Model

---

## ✓ The Event Object

- passed to every event handler
- initialized with event properties, including mouse and keyboard state
- can be used to control event bubbling and default actions
- re-initialized for each event

# Event Object

---

## ✓ Common Properties and Methods

- `event.target` = element that fired the event
- `event.cancelBubble = true` - cancels bubbling of event
- `event.returnValue = false` - cancels default action (see individual events for other uses)
- `event.type` = type of event (without “on”) Ex. if (`event.type == “click”`)

# Event Object

---

## ✓ Mouse Properties

- clientX, clientY - coords. relative to client area
- offsetX, offsetY - coords. relative to element/container
- screenX, screenY - coords. relative to screen
- button =
  - 0 - no buttons pressed
  - 1 - left
  - 2 - right
  - 4 - middle



# Event Object

---

## ✓ Mouse Properties (continued)

- ctrlKey = true     Control key pressed
- altKey = true     Alt key pressed
- shiftKey = true     Shift key pressed
- fromElement, toElement, onmouseover, onmouseout
- x, y - coords. Relative to parent positioned by CSS positioning attributes.

# Event Object

---

## ✓ Mouse Events

- onmousedown, onmouseup - button pressed/released
- onmousemove- mouse moved/moving
- onclick - clicked or default action of element
- ondblclick - button double clicked
- onmouseover, onmouseout - entered/exited scope of element

# Event Object

---

## ✓ Mouse Events

- ondragstart - start of drag-drop operation
- onselectstart - new selection started with mouse
- onselect - selection occurring

# Event Object

---

## ✓ Keyboard Properties

- keyCode - ASCII value of pressed key, can be changed (=0 in onkeypress cancels event)
- shiftKey, altKey, ctrlKey = true/false

# Event Object

---

## ✓ Scroll Event

- onscroll - fires when a scrollbar is scrolled

## ✓ Focus Events

- onfocus, onblur - fired when elements gain (lose) focus

## ✓ Help Event

- onhelp - fires when F1 keypressed

# Containership

---

- ✓ children collection
  - all of the child objects of a DOM object

# Identifying Elements/Attributes

---

✓ Attribute	Property
✓ (tag)	tagName (uppercase)
✓ ID	id
✓ CLASS	className
✓ NAME	name

# Dynamic Contents

---

- ✓ Properties
  - innerHTML