

JavaScript

Getting started with JS in the browser

Progressive Enhancement

What is it?

Why is it a good practice?

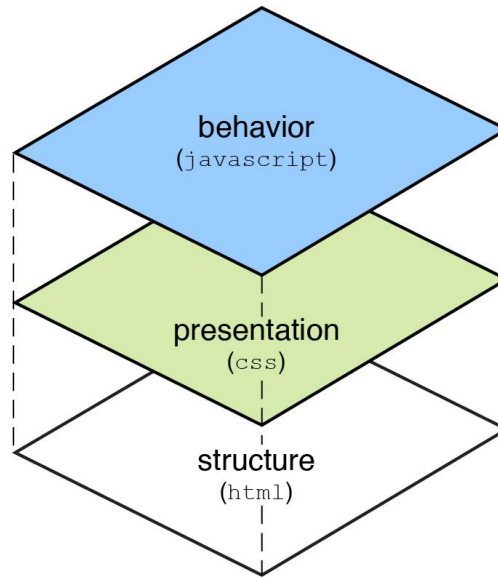


Figure 1.1 Graphical representation of progressive enhancement

From Learning JavaScript by Tim Wright (ISBN: 0321832744) Copyright © 2013 Pearson Education, Inc. All rights reserved.

JavaScript Fallbacks

What does it mean to have fallbacks for JS?

Why is it important?

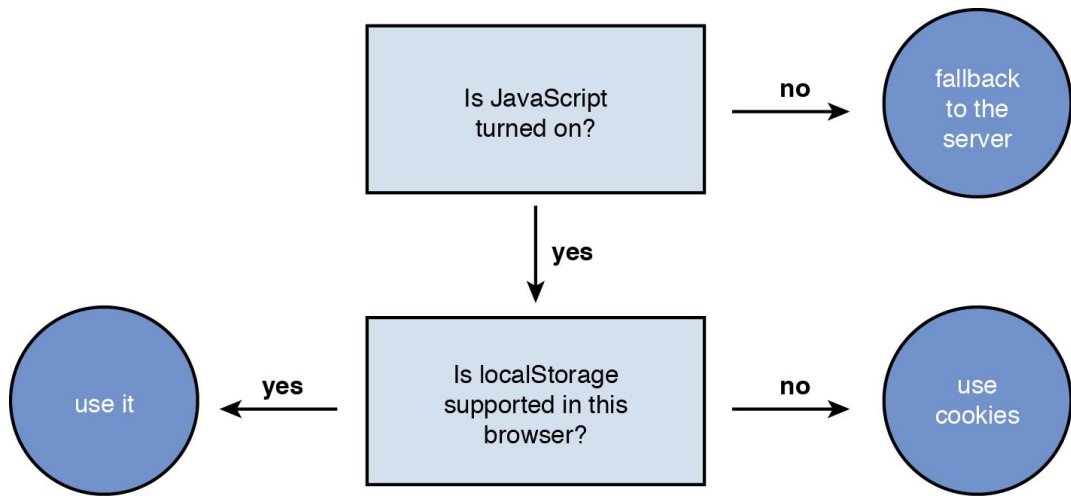


Figure 2.3 Flowchart - using proper fallbacks in JavaScript

From Learning JavaScript by Tim Wright (ISBN: 0321832744) Copyright © 2013 Pearson Education, Inc. All rights reserved.

Performance

Why is performance important?

What are some ways we can improve it?

DOM

What does DOM stand for?

What is the DOM?

```
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5      <meta charset="utf-8">
6      <title>Basic DOM Example</title>
7  </head>
8
9  <body>
10     <h1>Hello World</h1>
11     <p>
12         While this is a <strong>very basic HTML document</strong>,
13         it actually serves as a detailed example of the document object model.
14     </p>
15 </body>
16
17 </html>
```

Listing 4.1 Basic HTML Structure to illustrate the DOM

From Learning JavaScript by Tim Wright (ISBN: 0321832744) Copyright © 2013 Pearson Education, Inc. All rights reserved.

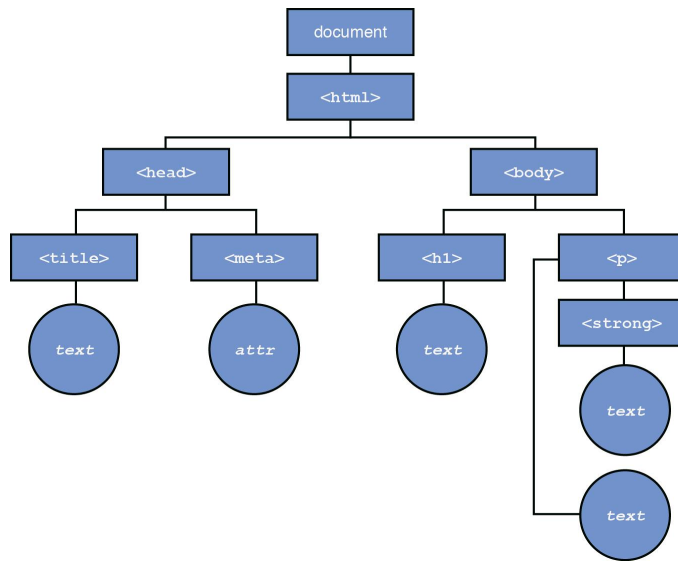


Figure 4.1 A graphical representation of the Document Object Model (DOM)

From Learning JavaScript by Tim Wright (ISBN: 0321832744) Copyright © 2013 Pearson Education, Inc. All rights reserved.

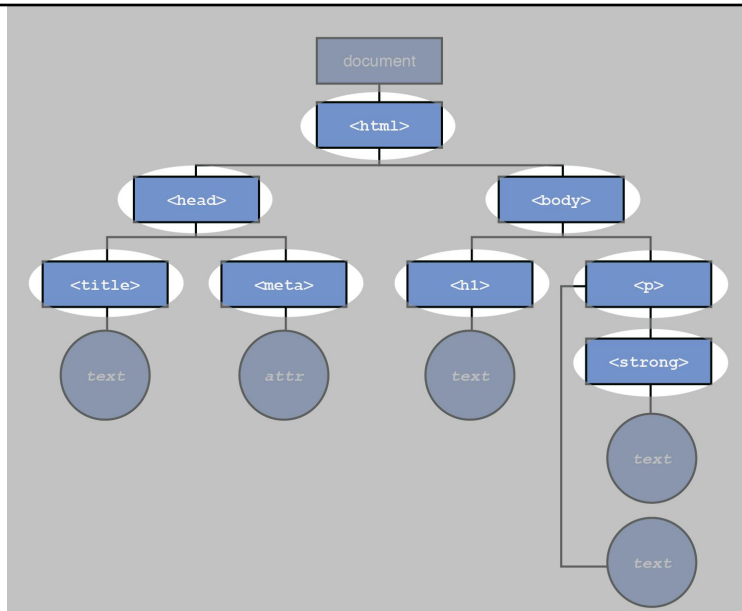


Figure 4.2 Highlighting the element nodes from our DOM.

From Learning JavaScript by Tim Wright (ISBN: 0321832744) Copyright © 2013 Pearson Education, Inc. All rights reserved.

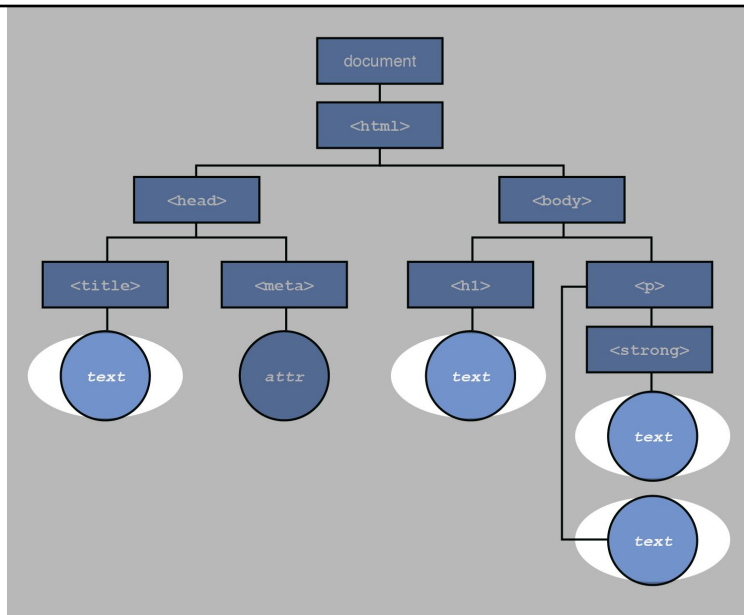


Figure 4.4 Highlighting the text nodes in our DOM.

From Learning JavaScript by Tim Wright (ISBN: 0321832744) Copyright © 2013 Pearson Education, Inc. All rights reserved.

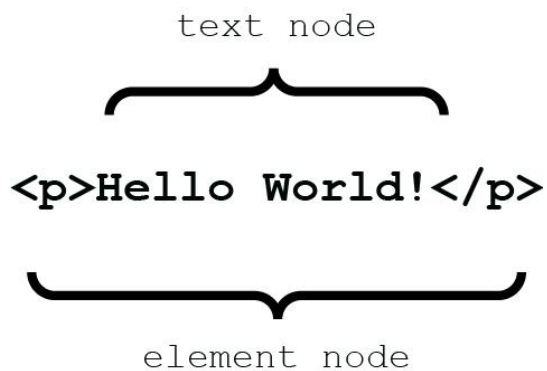


Figure 4.3 Showing the difference between an element node and a text node

From Learning JavaScript by Tim Wright (ISBN: 0321832744) Copyright © 2013 Pearson Education, Inc. All rights reserved.

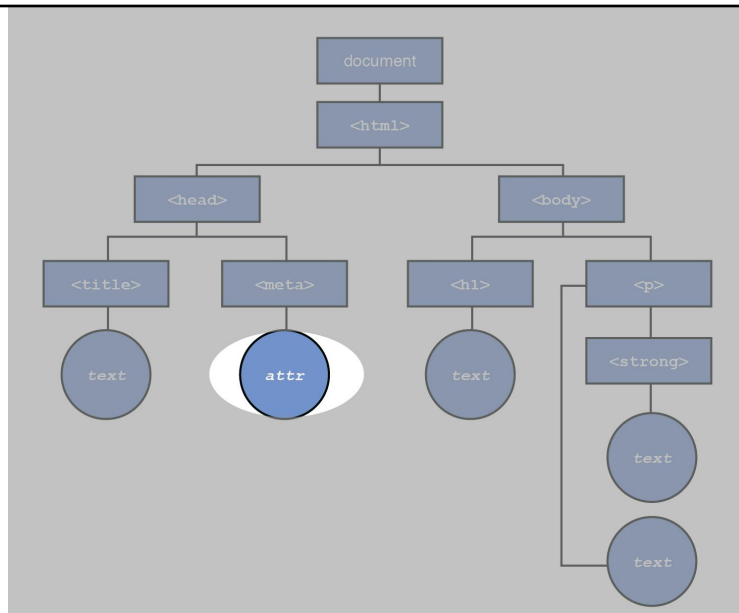


Figure 4.6 Highlighting the only attribute node in our DOM example.

From Learning JavaScript by Tim Wright (ISBN: 0321832744) Copyright © 2013 Pearson Education, Inc. All rights reserved.

Accessing Elements

Target by tag name or ID:

```
document.getElementsByTagName("h1");
```

```
document.getElementById("someID");
```

Moving around the DOM

Remember that relationships in the DOM are like a family tree.

- `parentNode`
- `previousSibling`
- `nextSibling`
- `firstChild`
- `lastChild`

Modifying Attributes

Does it have the attribute?

- `hasAttribute("class")`

Use JavaScript to add/remove attribute

- `setAttribute("class", "someClass")`
- `removeAttribute("class")`

Add / Remove Elements

We can modify the DOM to add / remove elements, changing the look of the page.

- createElement()
- createTextNode()
- appendChild()
- removeChild()

Create a New Paragraph

Add a new paragraph with text:

```
var target = document.getElementById("target");
var p = document.createElement("p");
var text = document.createTextNode("text here");
p.appendChild(text);
target.appendChild(p);
```

Assignment

Fix up Moonflower's Coffee Shop:

<https://github.com/htc-ccis2591/dom-access>

Due next week before class.

Fork & Submit a Pull Request.