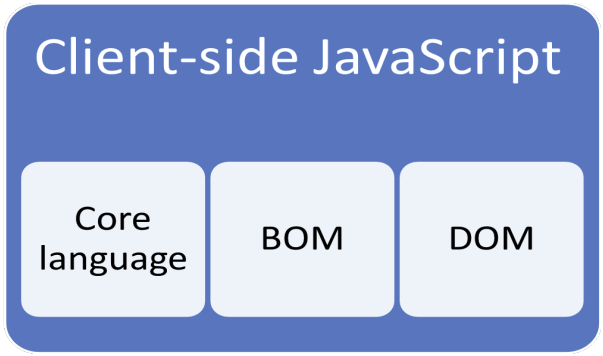
CS1116/CS5018
Web Development 2

Dr Derek Bridge
School of Computer Science & Information Technology
University College Cork

Client-side Javascript



The Browser Object Model (BOM)

- An **application programming interface** (API) for browsers:
  - `window` object;
    - Q: Where have we used this already?
  - `navigator` object
  - `location` object
  - `screen` object and
  - `history` object
- As part of HTML5, the BOM is increasingly standard across modern browsers

The Document Object Model (DOM)

- An API for HTML documents (and XML documents):
  - allows you to use JavaScript to change the content and appearance of a Web page without reloading it
- Treats the page as a **hierarchy** of different types of **node**:
  - allows you to use JavaScript to remove, replace, insert or change the nodes

Finding element nodes in JavaScript

- `element = document.querySelector('...');`
  Returns the first element in the document that matches the CSS selector, or `null` if no element matches the CSS selector
- `elements = document.querySelectorAll('....');`
  Returns an array-like collection of all elements in the document that match the CSS selector

Class exercise

```
<html lang="en">
    <head>
        <title>Lorem ipsum</title>
    </head>
    <body>
        <section id="sectA">
            <p class="opening">Lorem ipsum dolor sit amet consectetuer adipiscing.</p>
            <p class="closing">Lorem ipsum dolor sit amet consectetuer adipiscing.</p>
        </section>
        <section id="sectB">
            <p class="opening">Lorem ipsum dolor sit amet consectetuer adipiscing.</p>
            <p>Lorem ipsum dolor sit amet consectetuer adipiscing.</p>
            <p class="closing">Lorem ipsum dolor sit amet consectetuer adipiscing.</p>
        </section>
    </body>
</html>
```

Class exercise: which element nodes get retrieved?

1. `section_element = document.querySelector('#sectB');`
2. `p_elements = document.querySelectorAll('p');`
3. `p_elements = document.querySelectorAll('#sectB p');`
4. `opening_elements = document.querySelectorAll('.opening');`
5. `opening_elements = document.querySelectorAll('#sectB .opening');`

Creating new element nodes and new text nodes

- E.g. to create a new p element node:
  `new_p_element = document.createElement('p');`
- E.g. to create a new text node:
  `new_text_node = document.createTextNode('Lorem ipsum');`

Changing the tree's nodes and values

- Changing the nodes in the tree:
  - `some_node.appendChild(new_node);`
    Adds new_node to the end of the some_node's children
  - Others: `insertBefore`, `replaceChild`, `removeChild`
- Changing the text in a text node:
  - `some_text_node.nodeValue = 'Some new text';`
  - Others, e.g.: `appendData`, `insertData`, `deleteData`

A somewhat pointless example

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>Example A</title>
    <script src="exampleA.js" type="module">
    </script>
  </head>
  <body>
    <p>
      First paragraph.
    </p>
    <p>
      Second paragraph.
    </p>
  </body>
</html>
```

```
document.addEventListener('DOMContentLoaded', init, false);

function init() {
    let body_element = document.querySelector('body');
    let new_p_element = document.createElement('p');
    let new_text_node = document.createTextNode('Third paragr
    new_p_element.appendChild(new_text_node);
    body_element.appendChild(new_p_element);
}
```

A more concise approach

- Above we saw one way for JavaScript to put some text into a paragraph, i.e. create a text node and make it a child of the paragraph:

```
new_text_node = document.createTextNode('Third paragraph');
new_p_element.appendChild(new_text_node);
```

- A less cumbersome method is to assign to `innerHTML`:

```
new_p_element.innerHTML = 'Third paragraph';
```

- In fact, `innerHTML` is even more powerful than this example suggests

Properties of element nodes

- As you know, HTML elements may have attributes
  Q: Give examples
- JavaScript can retrieve and change attribute values by accessing and setting object properties
- E.g.

```
some_node.id = 'new_id';
```

Another pointless example

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>Example B</title>
    <script src="exampleB.js" type="module">
    </script>
  </head>
  <body>
    <section>
      <p>
        First paragraph.
      </p>
      <p>
        Second paragraph.
      </p>
    </section>
  </body>
</html>
```

```
document.addEventListener('DOMContentLoaded', init, false);

function init() {
    let section_element = document.querySelector('section');
    section_element.id = 'my_section';
    let p_elements = document.querySelectorAll('p');
    i = 0;
    for (let p of p_elements) {
        p.id = 'p' + i;
        i = i + 1;
    }
}
```

A proper example: the HTML

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8" />
        <title>Make me smile</title>
        <script src="rollover.js" type="module"></script>
    </head>
    <body>
        <p>
            <img id="face" src="frown.jpg">
        </p>
    </body>
</html>
```

A proper example: the JavaScript

```
let img_element;

document.addEventListener('DOMContentLoaded', init, false);

function init() {
    (new Image()).src = 'smile.jpg';
    img_element = document.querySelector('#face');
    img_element.addEventListener('mouseover', make_it_smile, false);
    img_element.addEventListener('mouseout', make_it_frown, false);
}

function make_it_smile() {
    img_element.src = 'smile.jpg';
}

function make_it_frown() {
    img_element.src = 'frown.jpg';
}
```

Changing the value of the class attribute

- We've seen how to change the `id`, `src` and other attributes
- To change the `class` attribute:
  - This doesn't work:

```
some_node.class = 'new_class';
```

  - You must write this:

```
some_node.className = 'new_class';
```

Yet another pointless example

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>Example C</title>
    <script src="exampleC.js" type="module">
    </script>
  </head>
  <body>
    <section>
      <p>
        First paragraph.
      </p>
      <p>
        Second paragraph.
      </p>
    </section>
  </body>
</html>
```

```
document.addEventListener('DOMContentLoaded', init, false);

function init() {
    let p_elements = document.querySelectorAll('p');
    for (let p of p_elements) {
        p.className = 'paras';
    }
}
```