CS1116/CS5018

Web Development 2

Dr Derek Bridge

School of Computer Science & Information Technology

University College Cork

Recap

- To find nodes: `querySelector`, `querySelectorAll`
- To create new element nodes: `document.createElement`
- To create new text nodes: `document.createTextNode`
- To insert an extra child node: `appendChild`
- To change the text of a text node: assign to `nodeValue`
- To change the value of an attribute of an element node: assign to an element node's properties

Changing an element's CSS in JavaScript

- JavaScript can retrieve and change an element's CSS by accessing and setting properties of its `style` object
- E.g.

```
some_node.style.color = 'blue';
```

  The visual display of the element is automatically changed

- CSS often uses hyphens in CSS property names but JavaScript uses camel case, e.g.

CSS	JavaScript
`background-color`	`style.backgroundColor`
`font-family`	`style.fontFamily`

Another of my pointless examples

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>Example D</title>
    <link rel="stylesheet"
          href="exampleD.css">
    <script src="exampleD.js" type="module">
    </script>
  </head>
  <body>
    <p>
      First paragraph.
    </p>
    <p>
      Second paragraph.
    </p>
  </body>
</html>
```

```
p {
    color: blue;
    background-color: yellow;
}
```

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

function init() {
    let first_p_element = document.querySelector('p:first-child');
    first_p_element.style.color = 'yellow';
    first_p_element.style.backgroundColor = 'blue';
}
```

Slideshow case study

```html
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8" />
        <title>Apartment For Sale</title>
        <link rel="stylesheet" href="styles.css">
        <script src="slideshow.js" type="module"></script>
    </head>
    <body>
        <header>
            <h1>Sherlock Homes – Estate Agents</h1>
        </header>
        <main>
            <h1>Apartment for sale – 12A Ivory Towers, Cork</h1>
            <p>
                We at <i>Sherlock Homes</i> are proud to present 12A Ivory Towers …
            </p>
            <figure>
                <img src="photo.jpg" alt="Front view" />
            </figure>
            <section>
                <h1>Accommodation</h1>
                <p>
                    This property is a spacious 70 square metres apartment …
                </p>
            </section>
            <section>
                <h1>Price</h1>
                <p>
                    The price is available upon request.
                </p>
            </section>
        </main>
        <aside>
            <ul id="slideshow">
                <li><img src="photo1.jpg" alt="Dining Area" /></li>
                <li><img src="photo2.jpg" alt="Living Room" /></li>
                <li><img src="photo3.jpg" alt="Bathroom" /></li>
                <li><img src="photo4.jpg" alt="Bedroom" /></li>
                <li><img src="photo5.jpg" alt="Exterior" /></li>
            </ul>
        </aside>
        <footer>
            &copy; Sherlock Homes Ltd.
        </footer>
    </body>
</html>
```

Slideshow case study

```css
* {
    margin: 0;
    padding: 0;
}

html {
    color: white;
    background-color: seagreen;
}

header {
    text-align: center;
    padding: 1em;
}

main {
    color: black;
    background-color: white;
    float: left;
    width: 70%;
    padding: 2.5%;
}

figure {
    text-align: center;
}

aside {
    text-align: center;
    margin-left: 75%;
    font-family: monospace;
}

aside ul {
    list-style: none;
}

aside img {
    width: 100%;
}

footer {
    clear: both;
    padding: 1em;
}
```

Slideshow case study

```
let img_elements;
let num_of_images;
let current_image_index;
let prev_link;
let next_link;
let count_display;

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

function init() {
    // Get the images
    img_elements = document.querySelectorAll('#slideshow img');
    num_of_images = img_elements.length;
    // Hide them all
    for (let image of img_elements) {
        image.style.display = 'none';
    }
    // Display the controls
    let control_div = create_control_div();
    let aside_element = document.querySelector('aside')
    aside_element.appendChild(control_div);
    // Show the first image
    current_image_index = 0;
    goto_image(null);
}

function create_control_div() {
    // Create the prev link
    prev_link = document.createElement('a');
    prev_link.appendChild(document.createTextNode(' < '));
    prev_link.addEventListener('click', goto_image, false);
    // Create the next link
    next_link = document.createElement('a');
    next_link.appendChild(document.createTextNode(' > '));
    next_link.addEventListener('click', goto_image, false);
    // Create the span to display the counter
    count_display = document.createTextNode('');
    // Create a div containing the prev link, counter display and next link
    let control_div = document.createElement('div');
    control_div.appendChild(prev_link);
    control_div.appendChild(count_display);
    control_div.appendChild(next_link);
    return control_div;
}

function goto_image(event) {
    // Hide the current slide
    img_elements[current_image_index].style.display = 'none';
    // Show the new current slide
    if (! event) {
        // don't change slide
    } else if (event.target === prev_link) {
        current_image_index -= 1;
    } else if (event.target === next_link) {
        current_image_index += 1;
    }
    img_elements[current_image_index].style.display = 'inline';
    // Hide the prev link if there is no previous slide
    if (current_image_index === 0) {
        prev_link.style.visibility = 'hidden';
    } else {
```

```
        prev_link.style.visibility = 'visible';
    }
    // Hide the next link if there is no next slide
    if (current_image_index === num_of_images - 1) {
        next_link.style.visibility = 'hidden';
    } else {
        next_link.style.visibility = 'visible';
    }
    // Update the counter
    count_display.nodeValue = (current_image_index + 1) + ' of ' + num_of_images;
}
```

Slideshow case study: CSS

`visibility`	whether an element is visible or not	
	`visible`	the element is visible
	`hidden`	the element is invisible, but it still takes up space
`display`	the type of box used for the element	
	`inline`	takes up as much width as necessary, and does not force line breaks
	`block`	takes up the full width available, and has a line break before and after it
	`none`	not displayed at all with no effect on layout (i.e. does not take up any space)
	…	lots of other variations

Questions about the case study

- Why `document.querySelectorAll('#slideshow img')`?
  Why not `document.querySelectorAll('img')`?
- Why does it make images invisible using `display: none` but make the '<' and '>' symbols invisible using `visibility: hidden`?
- Give a major limitation of this JavaScript program

An observation

- People overuse JavaScript — especially for simple changes of style
- Sometimes CSS alone suffices
    - E.g. hover effects can be done in CSS
    - E.g. animations can be done in CSS