Document Object Model

Programming interface for HTML documents

Document Object Model

Dynamically access and manipulate:

- Structure
- Content
- Style

Document Object Model

In Browsers:
Implemented/exposed as a
JavaScript object (API)

https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model

Accessing the DOM

```
<script>
  document
  window.document
</script>
```

An object representing our document

Data Types

- document
- node / element
- nodeList / HTMLCollection
- attribute

Examples uses

Selecting nodes (single):

```
var node = document.getElementById('main');

var node = document.querySelector('#main');
```

node

```
var node = document.getElementById('main');

// properties
node.parentNode
node.childNodes

// methods
node.append
node.appendChild
node.removeChild
// etc..
```

https://developer.mozilla.org/en-US/docs/Web/API/Node

Examples uses

Selecting nodes (multiple):

```
var nodeList = document.getElementsByClassName('quote

var nodeList = document.getElementsByTagName('div');

var nodeList = document.querySelectorAll('.quote');
```

nodeList

```
var nodeList = document.querySelectorAll('.quote');

// Array-ish with some methods
nodeList.length
nodeList.forEach
nodeList.contains
// etc..
```

Nodelists are static, and do not update if the underlying document updates

https://developer.mozilla.org/en-US/docs/Web/API/NodeList

HTMLCollection

```
// childNodes is a HTMLCollection
var node = document.querySelector('.container');
var children = node.children;
```

HTMLCollections are "live" - they update if the underlying document updates

https://developer.mozilla.org/en-US/docs/Web/API/HTMLCollection

Examples uses

Getting and setting attributes:

```
var node = document.getElementById("image");

// getting
var src = node.getAttribute('src');
var classNames = node.getAttribute('class');

// setting
node.setAttribute('src', 'image.jpg');
node.setAttribute('class', 'large');
```

Examples uses

Changing styles:

```
node.style.color = 'red';
node.style.backgroundColor = 'blue';
node.style.width = '40px';
```

Camel-cased property names:

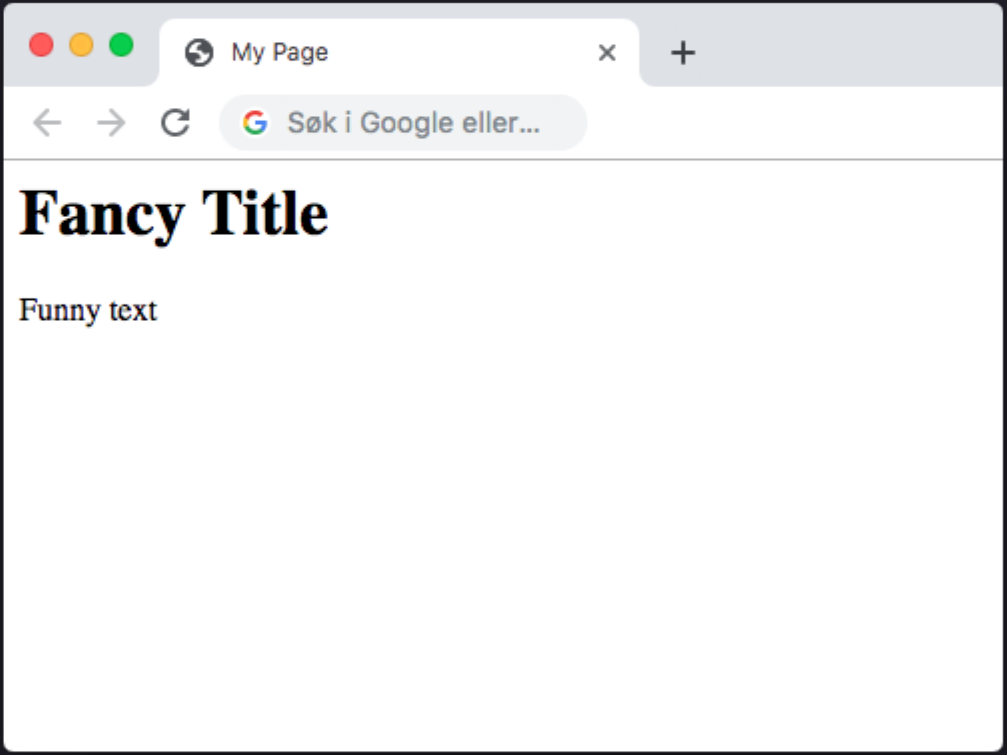
backgroundColor (JavaScript) vs. **background-color** (CSS)

Example

```html
<html>
  <head>
    <title>Old Title</title>
  </head>
  <body>
    <div id="main"></div>
  </body>
</html>
```

Example

Changing the document

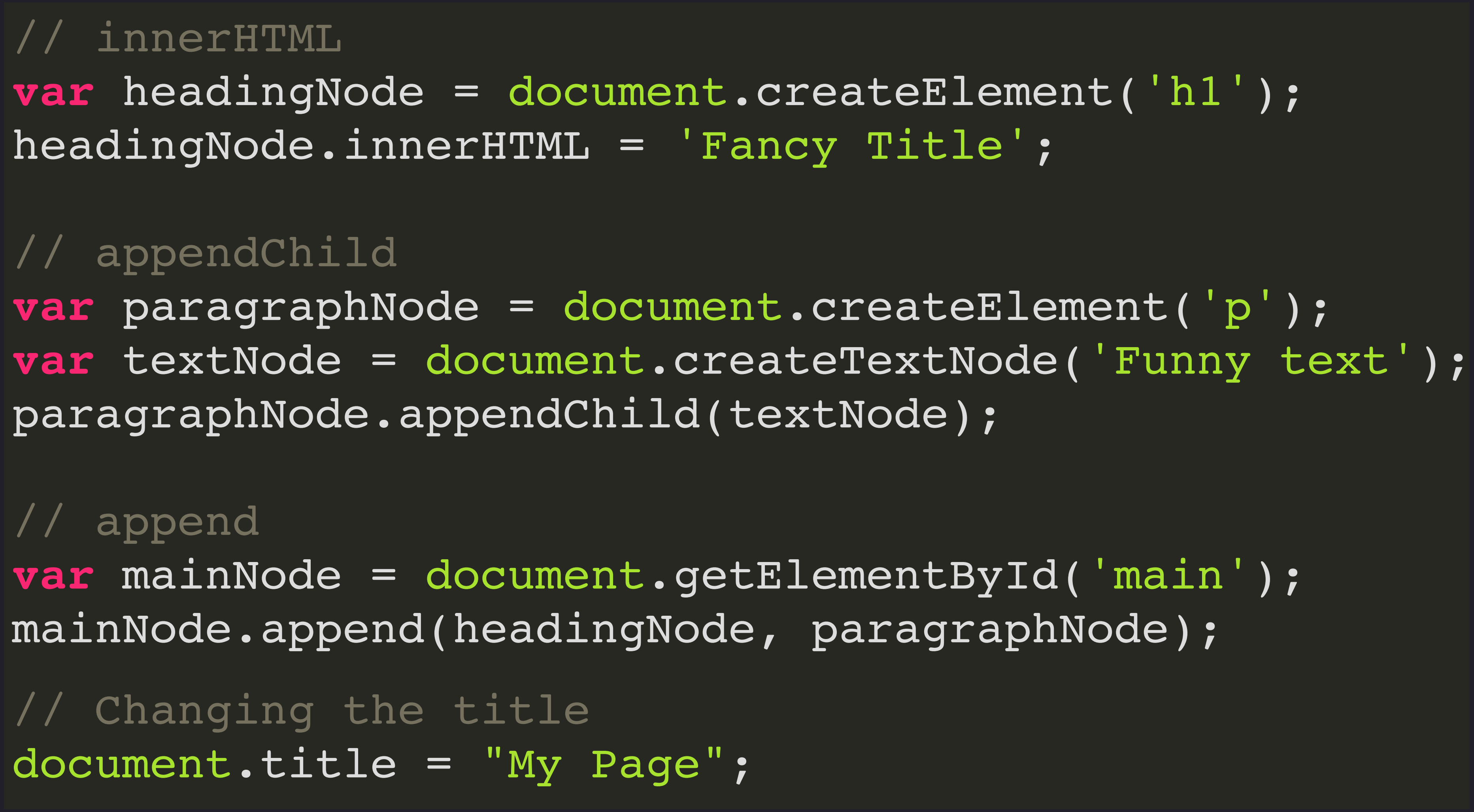


Fancy Title

Funny text

Example

Creating and appending nodes:



```
// innerHTML
var headingNode = document.createElement('h1');
headingNode.innerHTML = 'Fancy Title';

// appendChild
var paragraphNode = document.createElement('p');
var textNode = document.createTextNode('Funny text');
paragraphNode.appendChild(textNode);

// append
var mainNode = document.getElementById('main');
mainNode.append(headingNode, paragraphNode);

// Changing the title
document.title = "My Page";
```

Example

```
<html>
  <head>
    <title>My Page</title>
  </head>
  <body>
    <div id="main"></div>
    <script>
      var headingNode = document.createElement('h1');
      headingNode.innerHTML = 'Fancy Title';

      var paragraphNode = document.createElement('p');
      var textNode = document.createTextNode('Funny text');
      paragraphNode.appendChild(textNode);

      var mainNode = document.getElementById('main');
      mainNode.append(headingNode, paragraphNode);

      document.title = "My Page";
    </script>
  </body>
</html>
```