

Course Overview, Intro to JavaScript

ITP 301
Spring 2021

General Information

Lectures	Tuesdays, 4pm - 5:50pm
-----------------	------------------------

Labs	See TA Office Hours
-------------	---------------------

Website	http://itpwebdev.com/spring-2021/itp-301/ <i>Announcements, Lectures, Assignments, etc.</i>
----------------	--

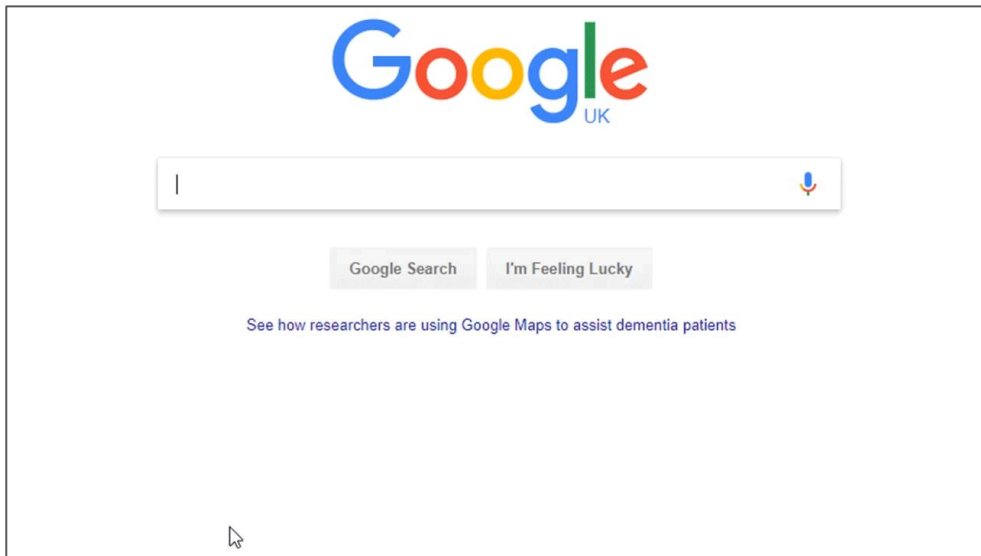
Zune Nguyen

Contact	zune@usc.edu
Education	B.S. in CECS from USC in 2014
Current	Teaching Web Development at USC since 2014
Past	Head of Technology @ NihaoPay CTO @ Bozway Inc. Intel, Dun & Bradstreet
Office Hours	Check Course Website

Course Goal

sit in the browser, wait to be triggered

- Learn how to build *interactive* websites using **client-side** programming.
- Real-time changes in the browser based on **events** or **actions**.
- Examples:



Course Goal

- Learn how to build *interactive* websites using client-side programming.
- Real-time changes in the browser based on **events** or **actions**.
- Examples:

Bootstrap Form

Full name

Username

Email address

Password

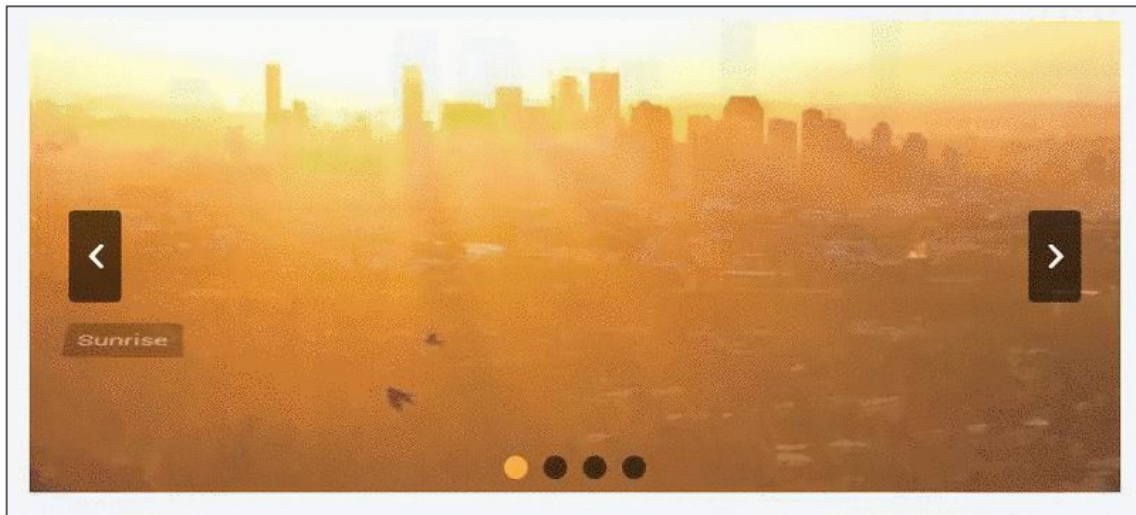
Gender ☐ Male
☐ Female
☐ Other

58 + 38 =

☐ Agree with the terms and conditions

Course Goal

- Learn how to build *interactive* websites using client-side programming.
- Real-time changes in the browser based on **events** or **actions**.
- Examples:



Syllabus

Midterm Exam

Date

Tuesday, 03/16/2021

Final Project

Main Prompt	Create an <i>interactive</i> website
Project Due Date	Wednesday, 05/05/2021 by 11:59pm
Past Examples	Bookmark Camping Central RecipMe Flickstagram Films, Books, Albums News Aggregator

Web Technologies



HyperText Markup Language

Creates elements on the page.

Tells browsers *what* to display.



Cascading Style Sheets

Defines element properties.

Tells browsers *how* to display elements.



JavaScript

HTML & CSS manipulation in *real-time*.

Tells browsers how to *behave*.

Web Technologies

HyperText Markup Language (HTML)

- Creates elements.

```
<!DOCTYPE html>
<html>
<head>
  <title>Lorem Ipsum</title>
  <style>
    #header {
      background-color: #900;
      color: #FFF;
    }
    img {
      display: inline-block;
      width: 200px;
      height: auto;
    }
  </style>
</head>
<body>
  <div id="header">
    <h1>Lorem ipsum</h1>
  </div> <!-- #header -->
  <div id="content">
    <div class="section">
      <h2>Dolor sit amet</h2>
      <p>Consectetur adipiscing elit. Vivamus ac sagittis sapien, vel venenatis neque.</p>
      <a href="nulla_at_risus">Nulla at risus</a>
    </div> <!-- .section -->
    <div class="section">
      
      
    </div> <!-- .section -->
  </div> <!-- #content -->
  <div id="footer">
    Posuere commodo turpis
  </div> <!-- #footer -->
  <script>
    document.querySelector('#header').onmouseenter = function(){
      this.style.backgroundColor = '#FC0';
      document.querySelector('#header h1').style.color = '#900';
    };
    var img = document.querySelectorAll('img');
    for (var i = 0; i < img.length; i++) {
      img[i].onclick = function(){
        this.style.width = '400px';
      }
    }
  </script>
</body>
</html>
```

Web Technologies

HyperText Markup Language (HTML)

- Creates elements.

Cascading Style Sheet (CSS)

- Defines element properties.

```
<!DOCTYPE html>
<html>
<head>
  <title>Lorem Ipsum</title>
  <style>
    #header {
      background-color: #900;
      color: #FFF;
    }
    img {
      display: inline-block;
      width: 200px;
      height: auto;
    }
  </style>
</head>
<body>
  <div id="header">
    <h1>Lorem ipsum</h1>
  </div> <!-- #header -->
  <div id="content">
    <div class="section">
      <h2>Dolor sit amet</h2>
      <p>Consectetur adipiscing elit. Vivamus ac sagittis sapien, vel venenatis neque.</p>
      <a href="nulla_at_risus">Nulla at risus</a>
    </div> <!-- .section -->
    <div class="section">
      
      
    </div> <!-- .section -->
  </div> <!-- #content -->
  <div id="footer">
    Posuere commodo turpis
  </div> <!-- #footer -->
  <script>
    document.querySelector('#header').onmouseenter = function(){
      this.style.backgroundColor = '#FC0';
      document.querySelector('#header h1').style.color = '#900';
    };
    var img = document.querySelectorAll('img');
    for (var i = 0; i < img.length; i++) {
      img[i].onclick = function(){
        this.style.width = '400px';
      }
    }
  </script>
</body>
</html>
```

Web Technologies

HyperText Markup Language (HTML)

- Creates elements.

Cascading Style Sheet (CSS)

- Defines element properties.

JavaScript (JS)

- Creates or removes elements, defines or re-defines their properties in **real-time**.

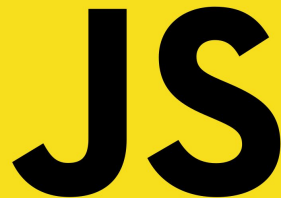
```
<!DOCTYPE html>
<html>
<head>
  <title>Lorem Ipsum</title>
  <style>
    #header {
      background-color: #900;
      color: #FFF;
    }
    img {
      display: inline-block;
      width: 200px;
      height: auto;
    }
  </style>
</head>
<body>
  <div id="header">
    <h1>Lorem ipsum</h1>
  </div> <!-- #header -->
  <div id="content">
    <div class="section">
      <h2>Dolor sit amet</h2>
      <p>Consectetur adipiscing elit. Vivamus ac sagittis sapien, vel venenatis neque.</p>
      <a href="nulla_at_risus">Nulla at risus</a>
    </div> <!-- .section -->
    <div class="section">
      
      
    </div> <!-- .section -->
  </div> <!-- #content -->
  <div id="footer">
    Posuere commodo turpis
  </div> <!-- #footer -->
  <script>
    document.querySelector('#header').onmouseenter = function(){
      this.style.backgroundColor = '#FC0';
      document.querySelector('#header h1').style.color = '#900';
    };
    var img = document.querySelectorAll('img');
    for (var i = 0; i < img.length; i++) {
      img[i].onclick = function(){
        this.style.width = '400px';
      }
    }
  </script>
</body>
</html>
```

JavaScript (JS)

The **only standardized** client-side scripting language.

Some features:

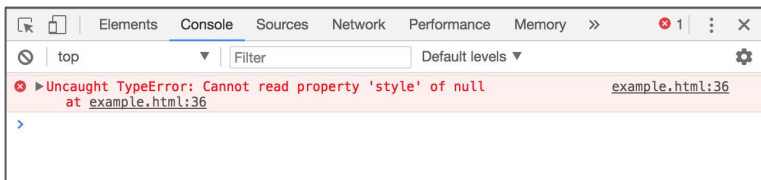
- Dynamic user interaction,
- DOM manipulation,
- Client-server communication without reloading pages,
- And more...



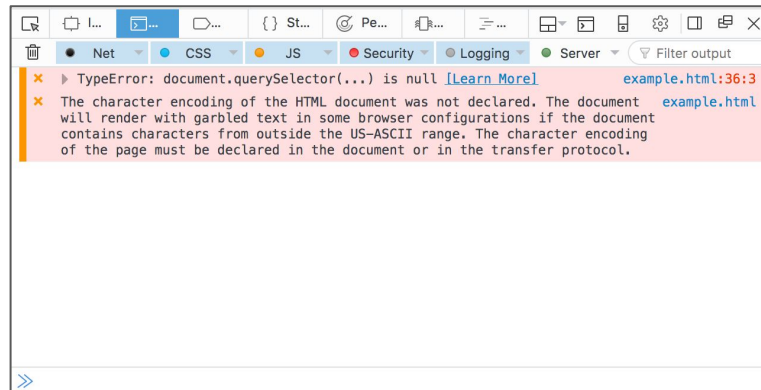
Debugging JS

JS errors and warnings are displayed in browser's developer console.

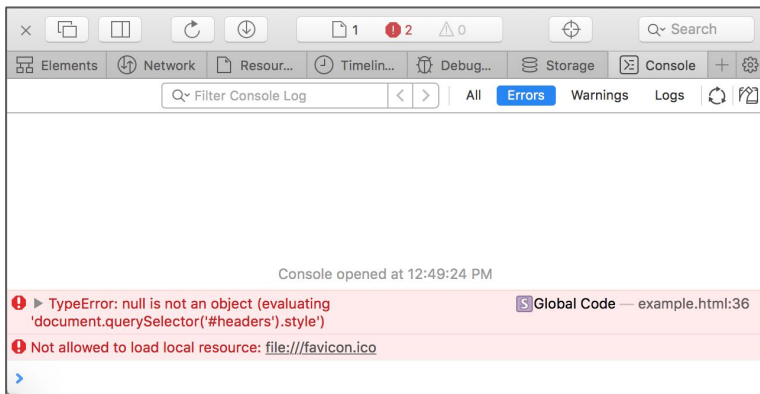
Chrome



FireFox



Safari



DOM

DOM – **D**ocument **O**bject **M**odel.

Cross-platform way of representing HTML objects in a document.

```
<!DOCTYPE html>
<html>
<head>
  <title>Lorem Ipsum</title>
</head>
<body>

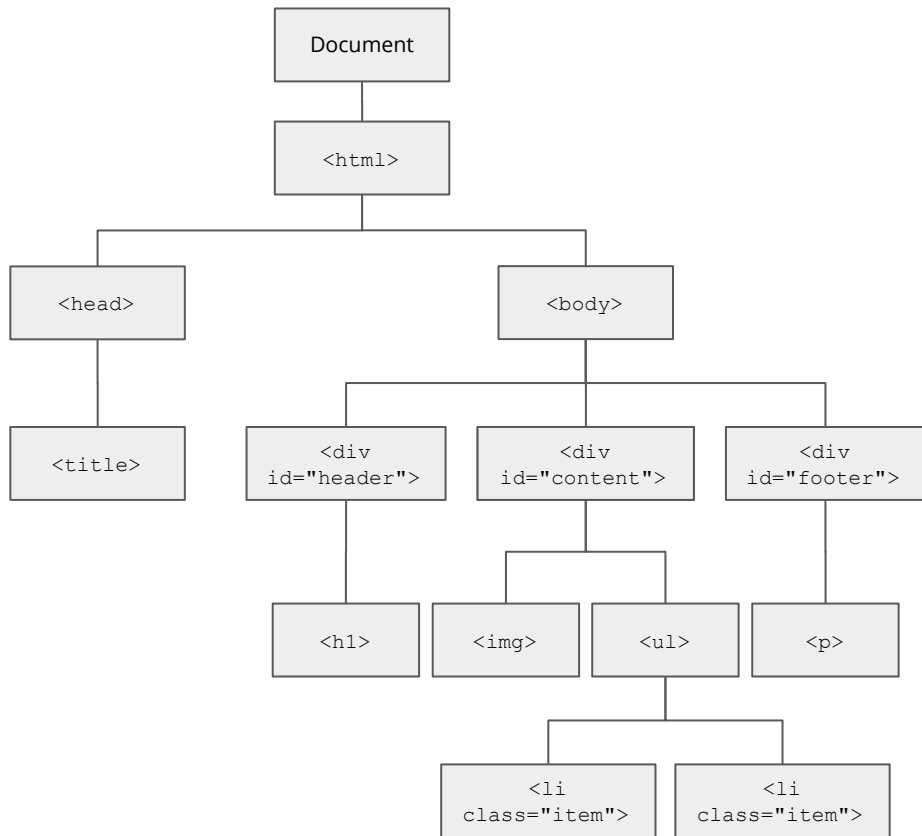
  <div id="header">
    <h1>Lorem Ipsum</h1>
  </div> <!-- #header -->

  <div id="content">
    
    <ul>
      <li class="item">Consectetur adipiscing elit.</li>
      <li class="item">Phasellus tempus posuere scelerisque.</li>
    </ul>
  </div> <!-- #content -->

  <div id="footer">
    <p>Suspendisse potenti.</p>
  </div> <!-- #footer -->

</body>
</html>
```


DOM



```
<!DOCTYPE html>
<html>
<head>
  <title>Lorem Ipsum</title>
</head>
<body>

  <div id="header">
    <h1>Lorem Ipsum</h1>
  </div> <!-- #header -->

  <div id="content">
    
    <ul>
      <li class="item">Consectetur adipiscing elit.</li>
      <li class="item">Phasellus tempus posuere scelerisque.</li>
    </ul>
  </div> <!-- #content -->

  <div id="footer">
    <p>Suspendisse potenti.</p>
  </div> <!-- #footer -->

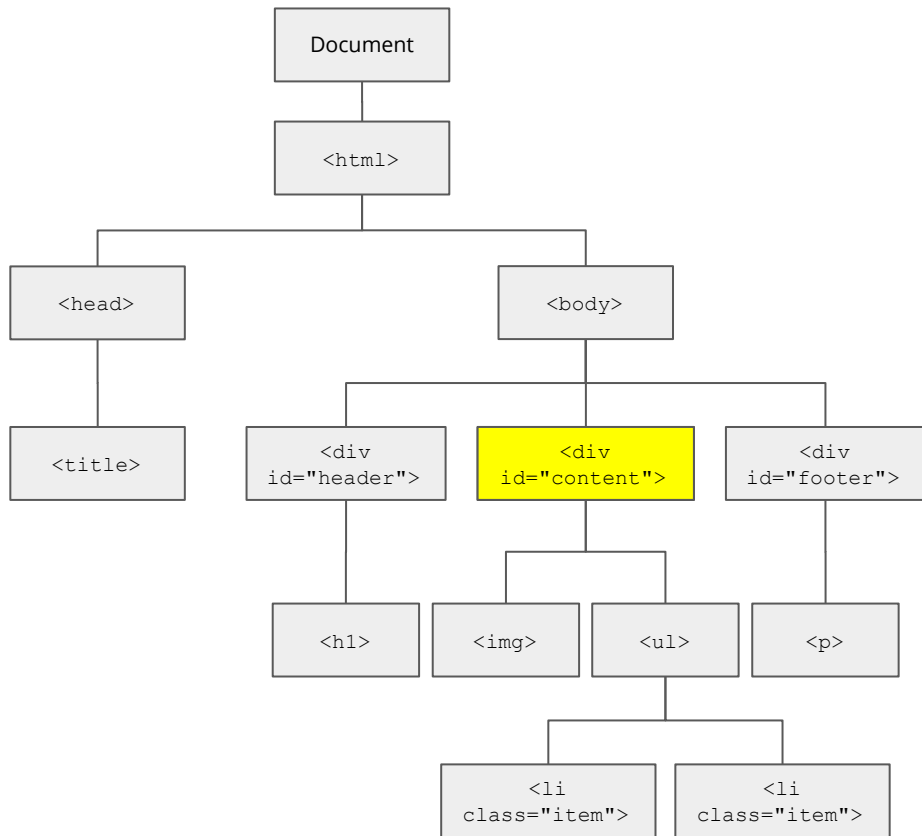
</body>
</html>
```

Accessing DOM Nodes

There are few ways to select DOM Nodes (HTML Elements) using JS:

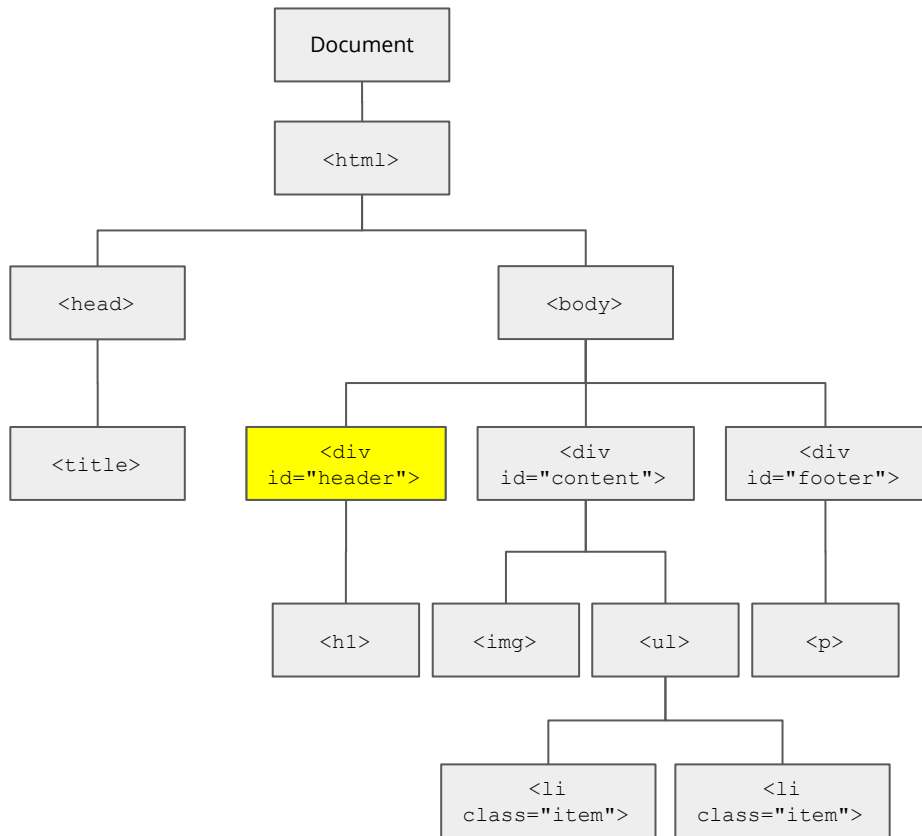
- `document.querySelector(CSS Selector)` *select the first element*
- `document.querySelectorAll(CSS Selector)` *select all the element*
- `document.getElementById(ID)`
- `document.getElementsByClassName(Class)`
- `document.getElementsByTagName(Tag)`

Accessing DOM Nodes



```
document.querySelector('#content');
```

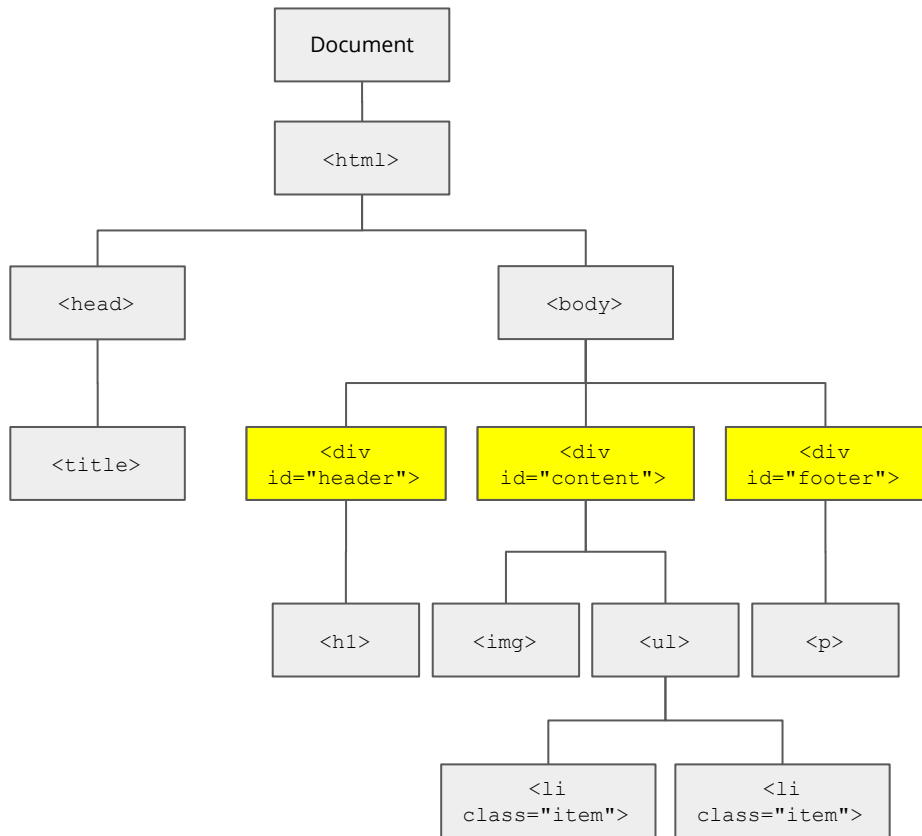
Accessing DOM Nodes



```
document.querySelector('#content');
```

```
document.querySelector('div');
```

Accessing DOM Nodes

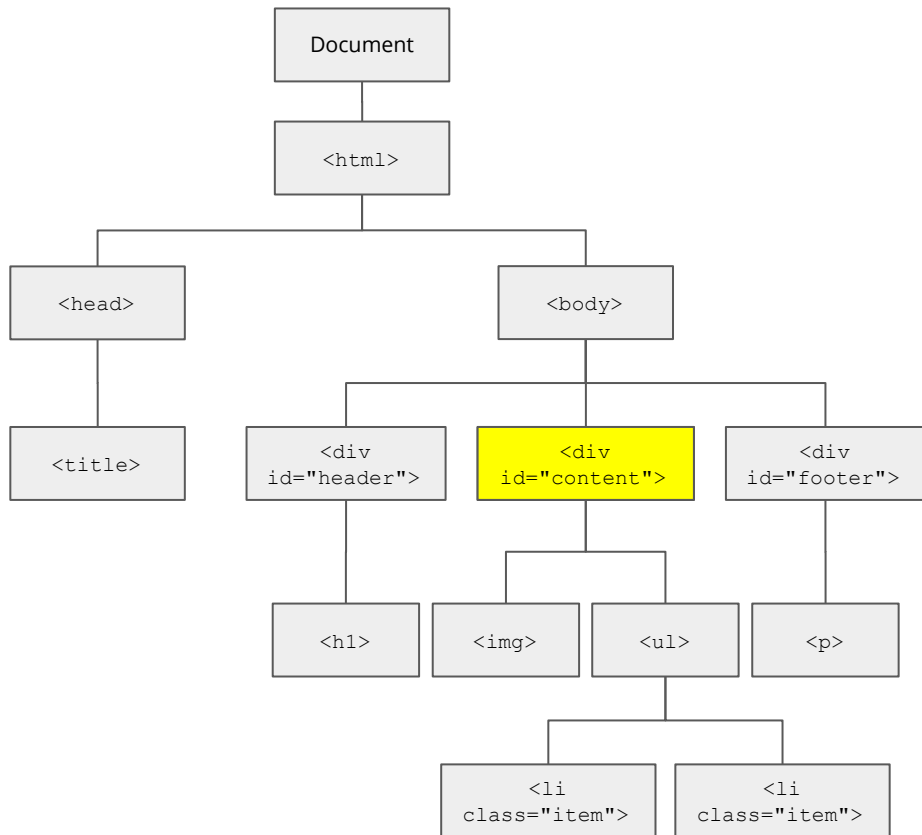


```
document.querySelector('#content');
```

```
document.querySelector('div');
```

```
document.querySelectorAll('div');
```

Accessing DOM Nodes



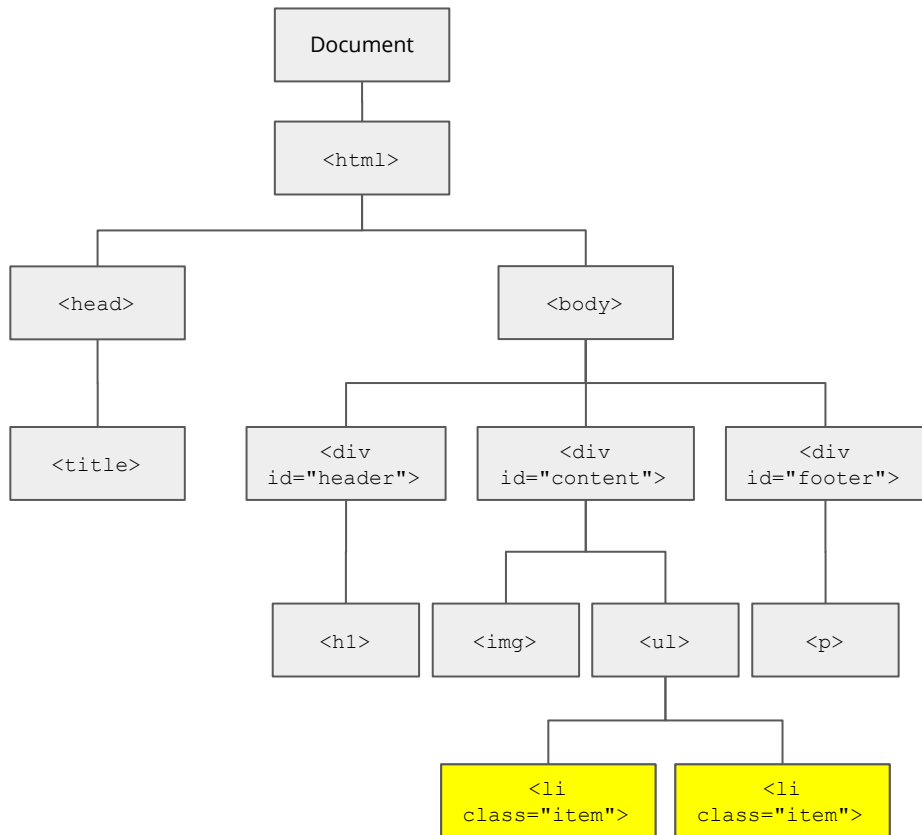
```
document.querySelector('#content');
```

```
document.querySelector('div');
```

```
document.querySelectorAll('div');
```

```
document.getElementById('content');
```

Accessing DOM Nodes



```
document.querySelector('#content');
```

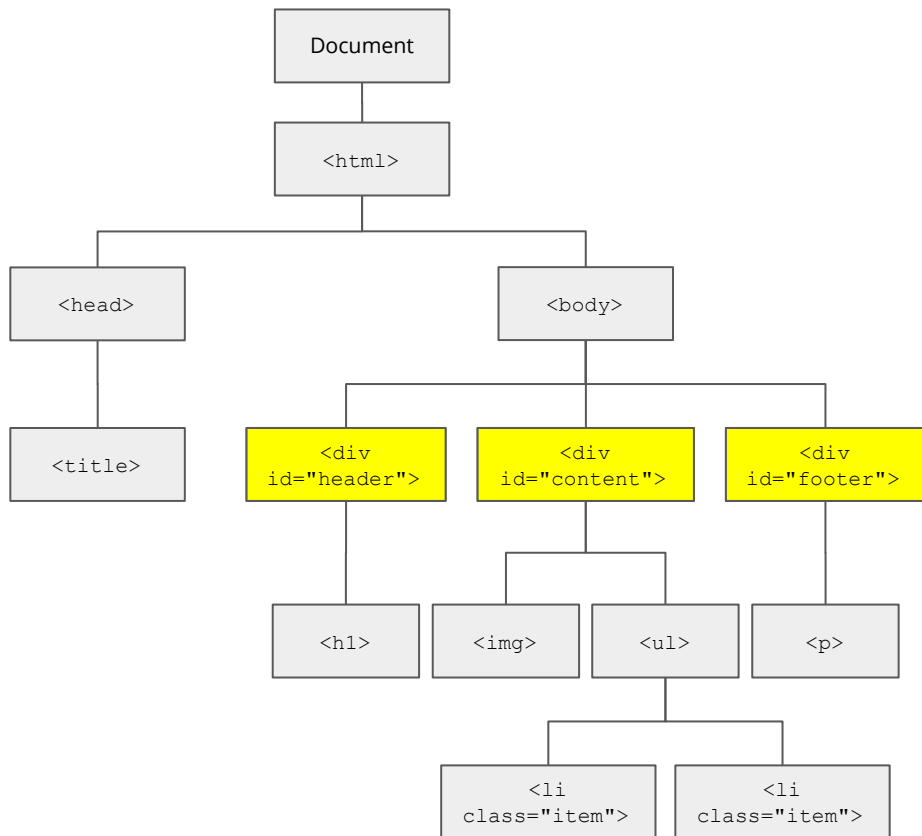
```
document.querySelector('div');
```

```
document.querySelectorAll('div');
```

```
document.getElementById('content');
```

```
document.getElementsByClassName('item');
```

Accessing DOM Nodes



```
document.querySelector('#content');
```

```
document.querySelector('div');
```

```
document.querySelectorAll('div');
```

```
document.getElementById('content');
```

```
document.getElementsByClassName('item');
```

```
document.getElementsByTagName('div');
```


Setting Object Properties

To set object properties:

1. Access them directly using dot notation:

```
object.propertyName
```

2. Set them equal to new value:

```
object.propertyName = newValue
```

All CSS properties are access through `style` nested object.

Change `src` attribute of `#my-img`

```
document.querySelector('#my-img').src = 'new-image.jpg';
```

Change `alt` attribute of `#my-img`

```
document.querySelector('#my-img').alt = 'New Text';
```

Change `href` attribute of `#my-url`

```
document.querySelector('#my-url').href = 'new-destination.html';
```

Change `background-color` CSS property of `#my-text`

```
document.querySelector('#my-text').style.backgroundColor = '#900';
```

Mouse Events

User's mouse interaction detected by JS.

<code>onclick</code>	Mouse click on an element.
<code>onmouseenter</code>	Mouse enters an element.
<code>onmouseleave</code>	Mouse leaves an element.
<code>onmouseover</code>	Mouse enters an element or its children.
<code>onmouseout</code>	Mouse leaves an element or its children.

Resource: [Full list of DOM Events.](#)

```
document.querySelector('#content').onclick = function(){  
  document.querySelector('#content').style.backgroundColor = '#FC0';  
}
```

```
document.querySelector('div').onmouseenter = function(){  
  document.querySelector('#name').innerHTML = 'Tommy';  
}
```