JAVASCRIPT
& JQUERY
interactive front-end
web development

CHAPTER 5

DOCUMENT OBJECT MODEL

The DOM specifies how:

The DOM specifies how:

1

Browsers
create a model of
an HTML page

The DOM specifies how:

1

Browsers
create a model of
an HTML page

2

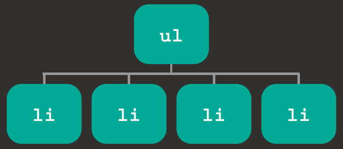
JavaScript
accesses / updates
an HTML page

THE DOM TREE

```
<ul>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
</ul>
```

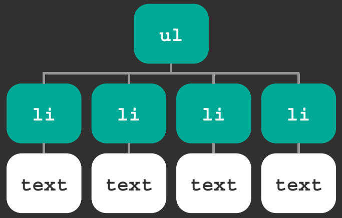
ELEMENT NODES

```
<ul>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
</ul>
```



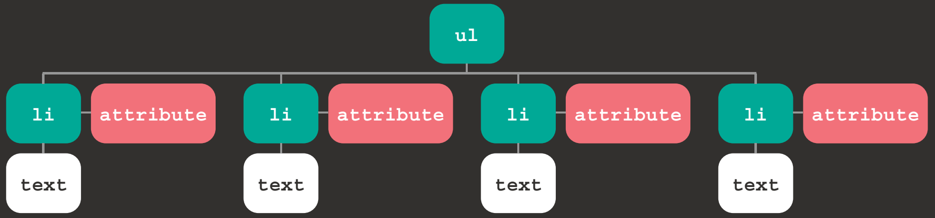
TEXT NODES

```
<ul>
  <li>fresh figs</li>
  <li>pine nuts</li>
  <li>honey</li>
  <li>balsamic vinegar</li>
</ul>
```



ATTRIBUTE NODES

```
<ul>
  <li id="one" class="hot">fresh figs</li>
  <li id="two" class="hot">pine nuts</li>
  <li id="three" class="hot">honey</li>
  <li id="four">balsamic vinegar</li>
</ul>
```



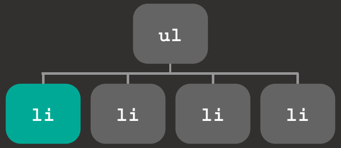
To access and update the HTML, first you select the element(s) you want to work with.

Here are some of the ways ways to select element nodes.

They are known as **DOM queries**.

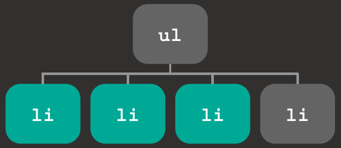
DOM QUERIES

```
<ul>
  <li id="one" class="hot">fresh figs</li>
  <li id="two" class="hot">pine nuts</li>
  <li id="three" class="hot">honey</li>
  <li id="four">balsamic vinegar</li>
</ul>
```



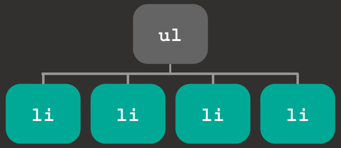
```
getElementById('one');
```

```
<ul>
  <li id="one" class="hot">fresh figs</li>
  <li id="two" class="hot">pine nuts</li>
  <li id="three" class="hot">honey</li>
  <li id="four">balsamic vinegar</li>
</ul>
```



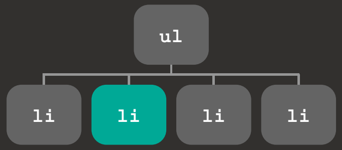
```
getElementsByClassName('hot');
```

```
<ul>
  <li id="one" class="hot">fresh figs</li>
  <li id="two" class="hot">pine nuts</li>
  <li id="three" class="hot">honey</li>
  <li id="four">balsamic vinegar</li>
</ul>
```



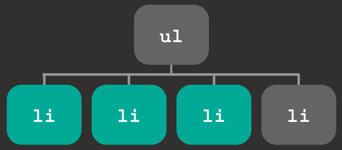
```
getElementsByTagName('li');
```

```
<ul>
  <li id="one" class="hot">fresh figs</li>
  <li id="two" class="hot">pine nuts</li>
  <li id="three" class="hot">honey</li>
  <li id="four">balsamic vinegar</li>
</ul>
```



```
querySelector('#two');
```

```
<ul>
  <li id="one" class="hot">fresh figs</li>
  <li id="two" class="hot">pine nuts</li>
  <li id="three" class="hot">honey</li>
  <li id="four">balsamic vinegar</li>
</ul>
```



```
querySelectorAll('li.hot');
```

NODELISTS

If a DOM query returns more than one element, it is known as a **NodeList**.

Items in a NodeList are numbered and selected like an array:

```
var elements;

elements = getElementsByClassName('hot');

var firstItem = elements[0];
```

You can check if there are elements before using a NodeList:

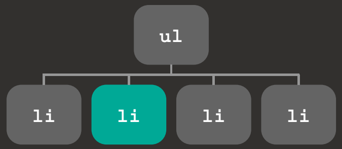
```
if (elements.length >= 1) {
  var firstItem = elements[0];
}
```

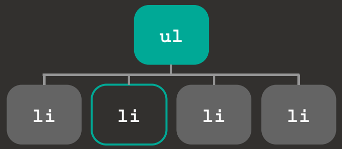
TRAVERSING THE DOM

You can move from one node to another if it is a relation of it.

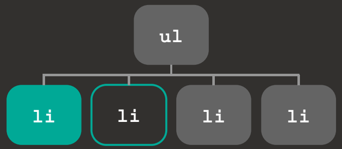
This is known as **traversing the DOM**.

STARTING ELEMENT

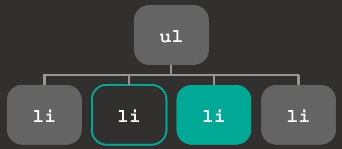




parentNode

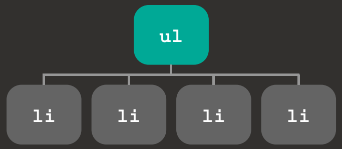


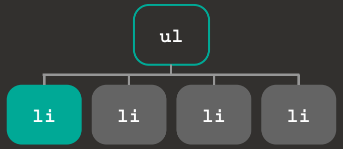
previousSibling



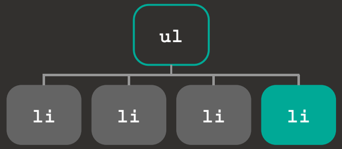
nextSibling

STARTING ELEMENT





firstChild

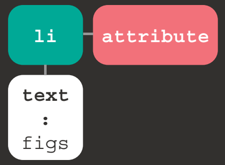


lastChild

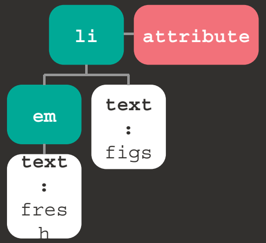
WORKING WITH ELEMENTS

Elements can contain:

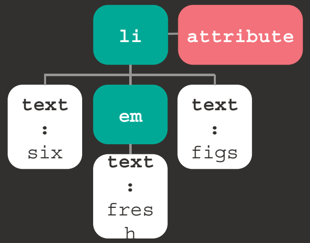
Text nodes
Element content
Attributes



```
<li id="one">figs</li>
```



```
<li id="one"><em>fresh</em> figs</li>
```

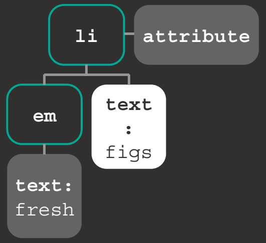


```
<li id="one">six <em>fresh</em> figs</li>
```

To access their content you can use:

`nodeValue` on text nodes
`textContent` for text content of elements
`innerHTML` for text and markup

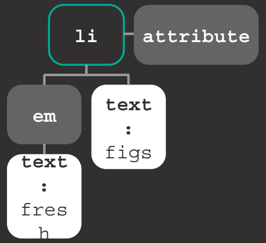
`nodeValue` works on text nodes



```
var el = document.getElementById('one');
el.firstChild.nextSibling.nodeValue;
```

returns: figs

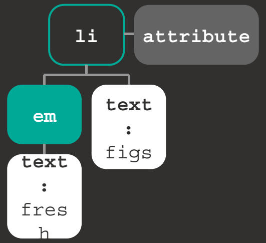
`textContent` just collects text content



```
document.getElementById('one').textContent;
```

returns: fresh figs

`innerHTML` gets text and markup



```
document.getElementById('one').innerHTML;
```

returns: `<em>`fresh`</em>` figs

DOM MANIPULATION VS `innerHTML`

```
createElement()
createTextNode()
appendChild()
```

- Builds up a string
- Contains markup
- Updates elements

CROSS-SITE SCRIPTING (XSS) ATTACKS

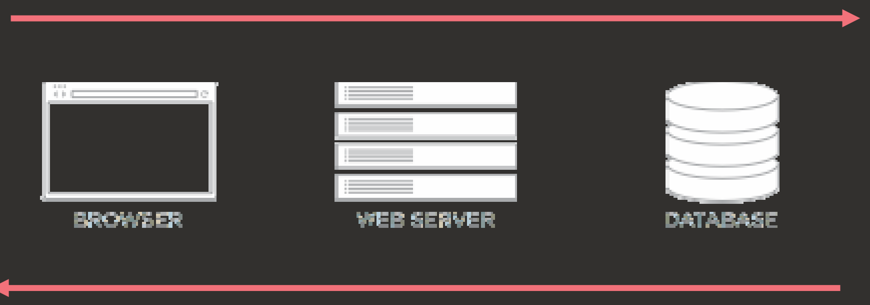
Untrusted data is content you do not have complete control over. It can contain malicious content.

Sources of untrusted data:

User creates a profile
Multiple contributors
Data from third-party sites
Files such as images / videos are uploaded

DEFENDING AGAINST XSS

Validate all input that is sent to the server



Escape data coming from the server

WORKING WITH ATTRIBUTES

ACCESSING AN ATTRIBUTE

1. Use a DOM query to select an element:

```
var el = document.getElementById(‘one’);
```

2. Method gets attribute from element:

```
el.getAttribute(‘class’);
```

UPDATING AN ATTRIBUTE

Check for attribute and update it:

```
var el = document.getElementById(‘one’);

if (el.hasAttribute(‘class’) {
  el.setAttribute(‘class’, ‘cool’);
}
```