# DOM

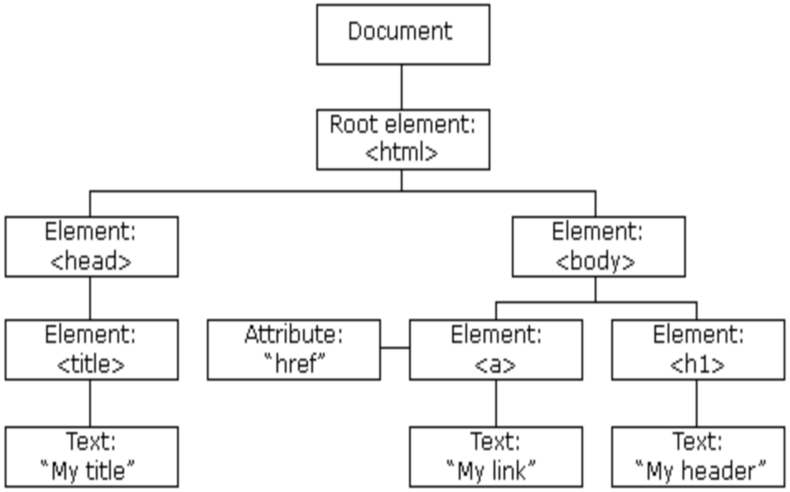
document.

# DOM

- javascript - Netscape (1995)
- Browser console
- window
- document

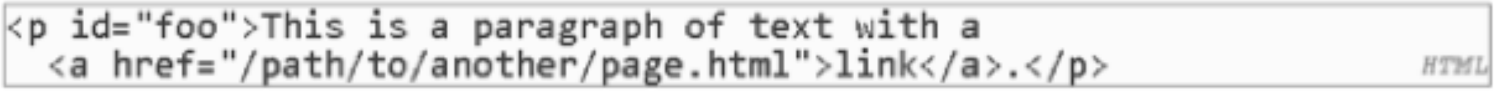
# DOM

```
<!doctype html>
<html>
<head>
   <title>My title</title>
</head>
<body>
      <a href=“xyz.html”>My link</a>
       <h1>My header</h1>
</body>
</html>
```

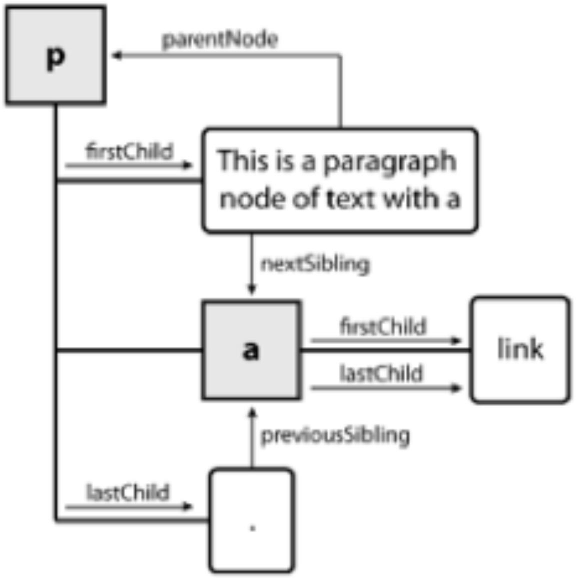


- JS objects represent elements on a page
- Use JS to traverse, inspect, add, remove elements
- Aside from “tag”, use  “id” or “class” to find elements

# DOM Traversal



```html
<p id="foo">This is a paragraph of text with a
  <a href="/path/to/another/page.html">link</a>.</p>
```



# document.

- *getElementById()*
- *getElementsByTagName()*
- *createElement()*

# element.

- *innerHTML*
- *getAttribute()*
- *addEventListener()*

# Event Driven

- *mouse*
  - *onclick, ondblclick, onmousedown/up/over/move/out*
- *keboard*
  - *onkeypress/down/up*
- *form*
  - *onblur/focus/select/change/reset/submit*
- *frame*
  - *onload/abort/resize/scroll/unload*

# Where ?

- *inline*

```
<div id="example1" onclick="alert('hello')">Click for pop-up</div>
```

- *embedded*

```
<script type=“text/javascript”>
     console.log(“Hello World”)
</script>
```

- *external*

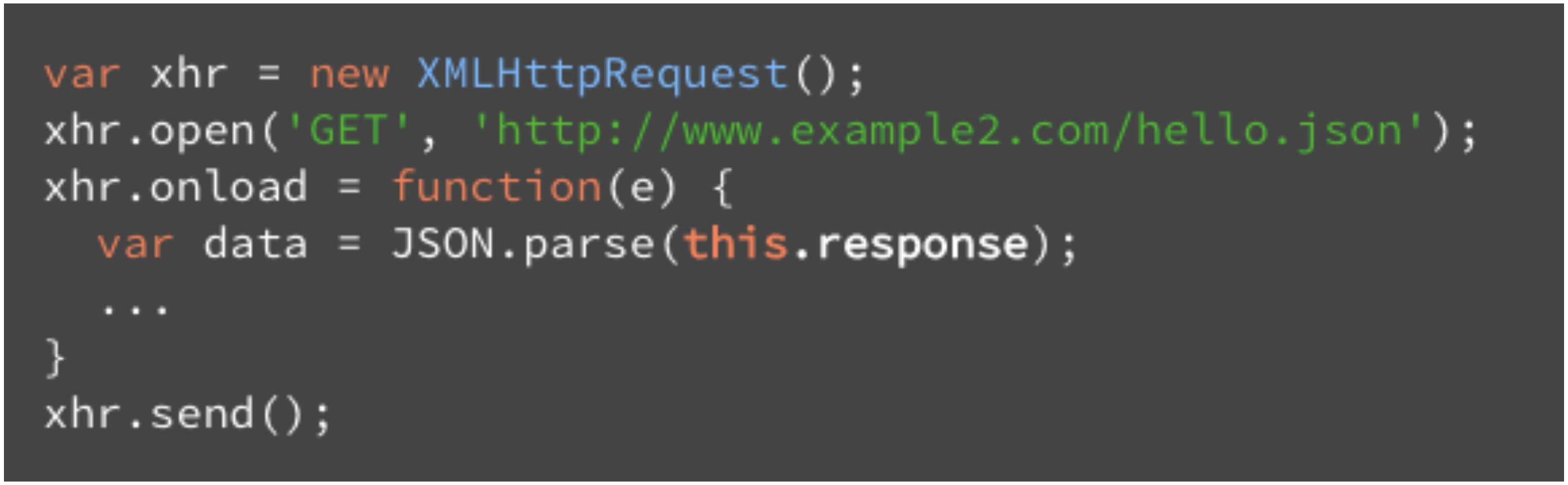
```
<script type=“text/javascript” src=“code.js”></script>
```

# jQuery

- Cross-browser compatibility
- Easier DOM manipulation
- Powerful event binding
- Chaining
- AJAX

# AJAX

- An architecture style ?
- XMLHttpRequest
- DOM manipulation



```
var xhr = new XMLHttpRequest();
xhr.open('GET', 'http://www.example2.com/hello.json');
xhr.onload = function(e) {
  var data = JSON.parse(this.response);
  ...
}
xhr.send();
```

# and REST!

```
$.getJSON("http://api.flickr.com/services/feeds/photos_public.gne?jsoncallback=?",
{
    tags: "mount rainier",
    tagmode: "any",
    format: "json"
},
function (data) {
  $.each(data.items, function (i, item) {
        $("<img/>").attr("src", item.media.m).appendTo("#images");
        if (i == 3) return false;
    });
});
```