

INTRODUCTION TO jQuery

DOM Interaction, simplified.

GLOSSARY OF TERMS

- **DOM: Document Object Model**
 - Browser's representation of HTML
 - All elements in the tree
- **Client-side**
 - As opposed to server-side
 - Javascript executed in the browser
- **Selectors**

FRONT-END APPLICATIONS
=
USER INTERFACES

MANIPULATING THE DOM BASED ON EVENTS AND INTERACTIVITY

JAKE WARY









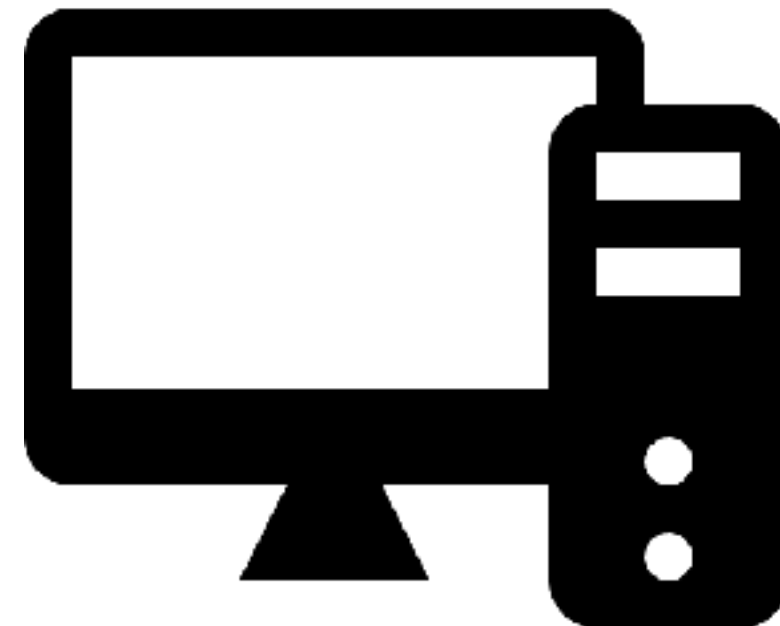
GET /

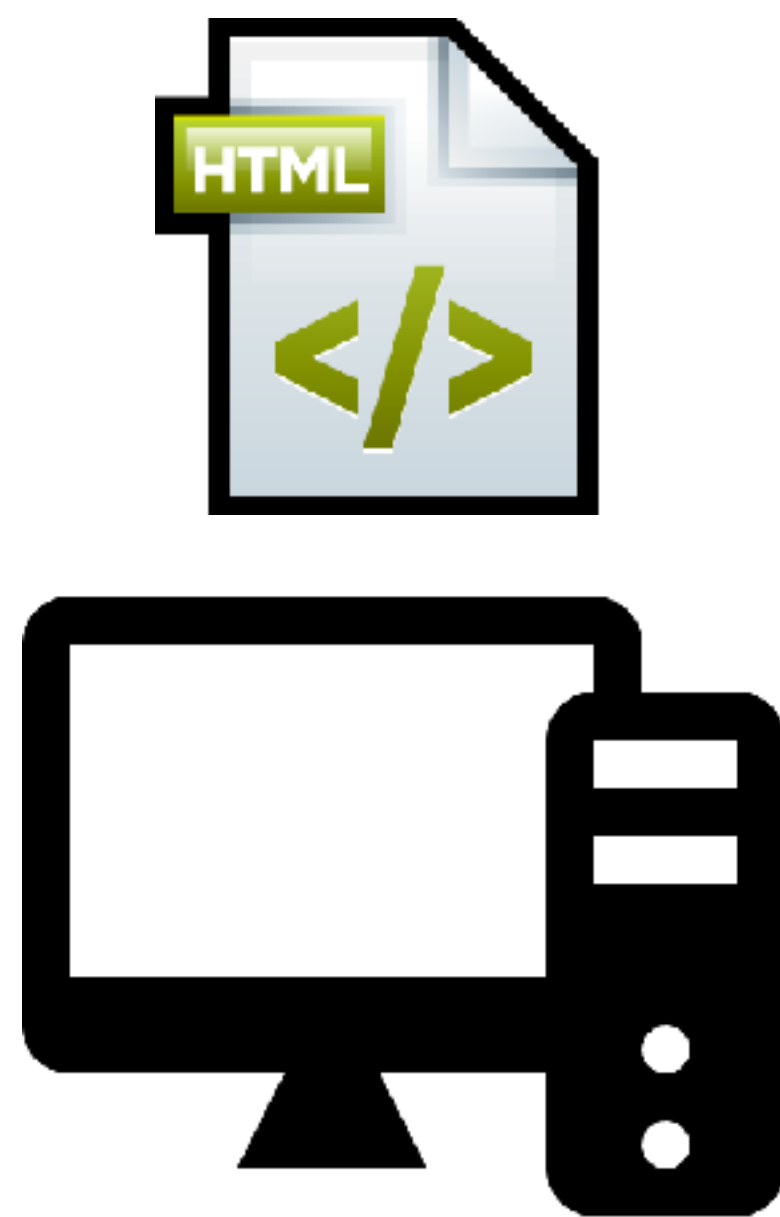

```
app.get('/', function (req, res) {  
  res.sendFile(__dirname + '/index.html');  
});
```

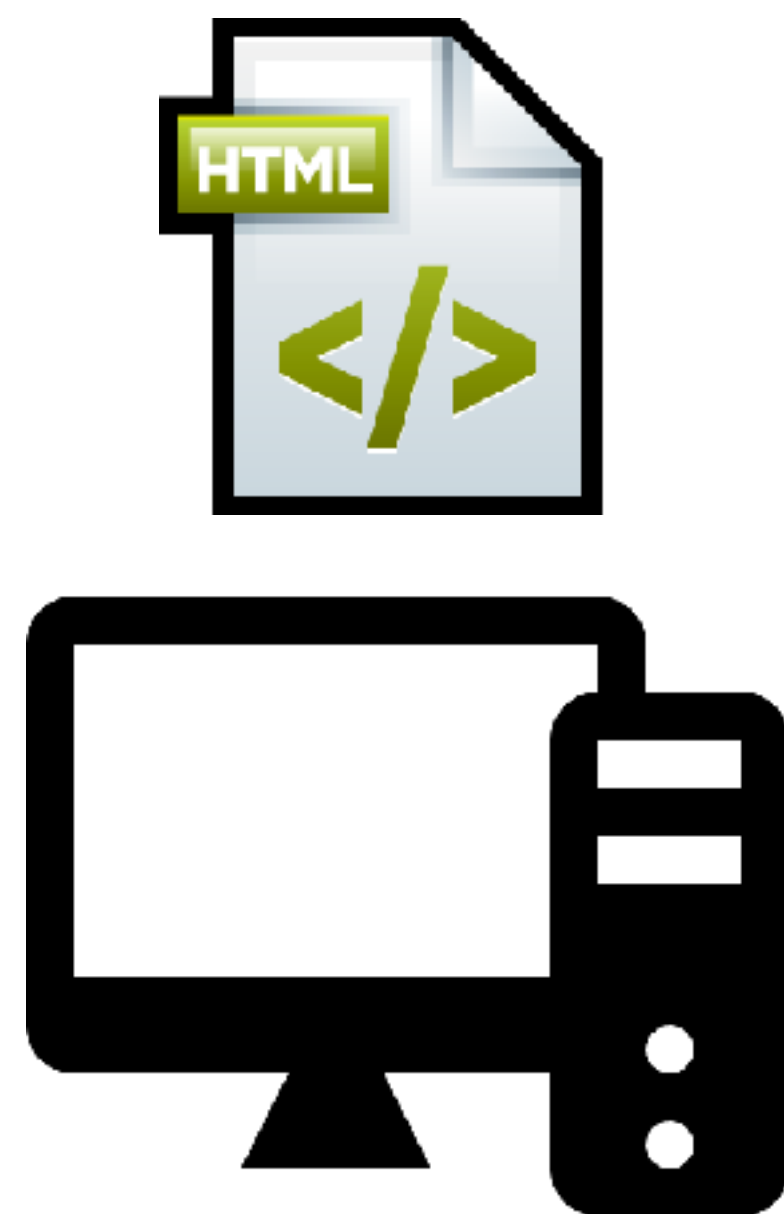


GET /

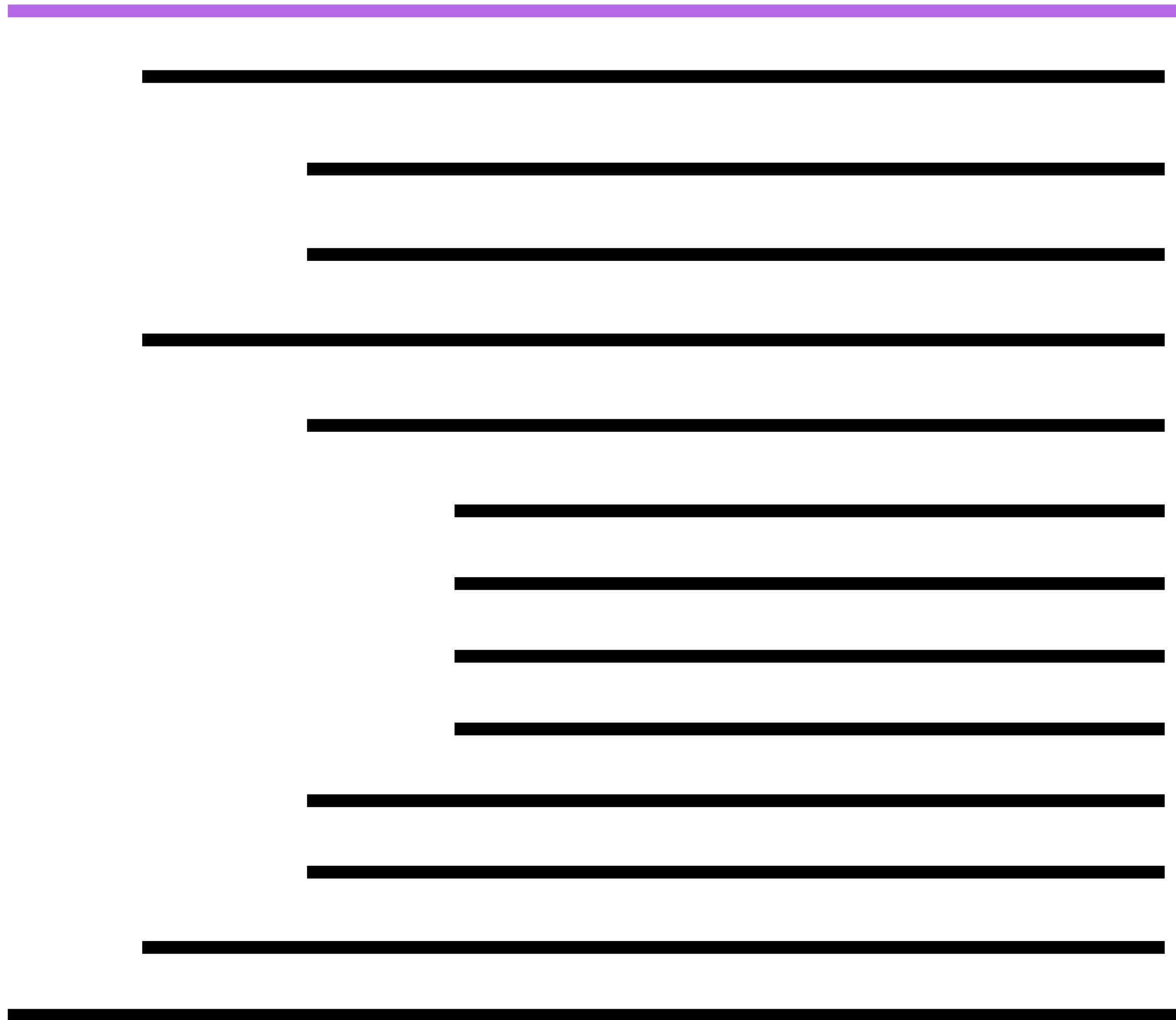


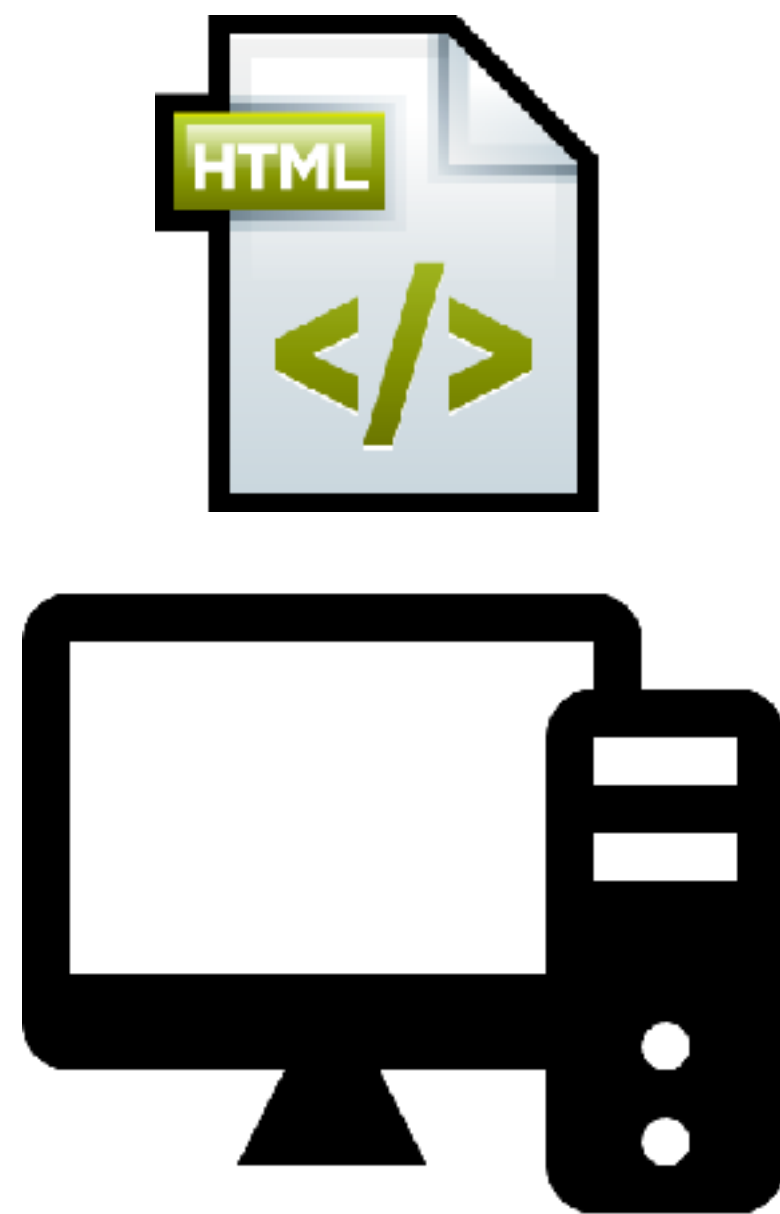




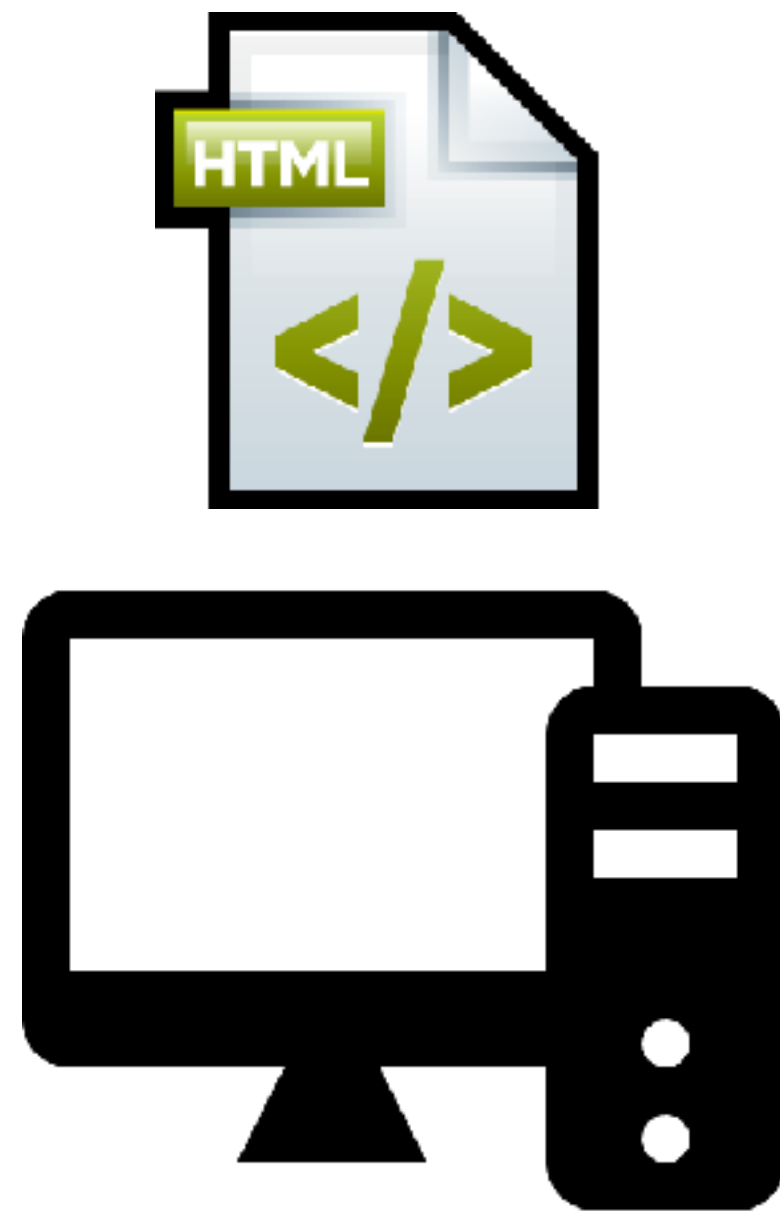


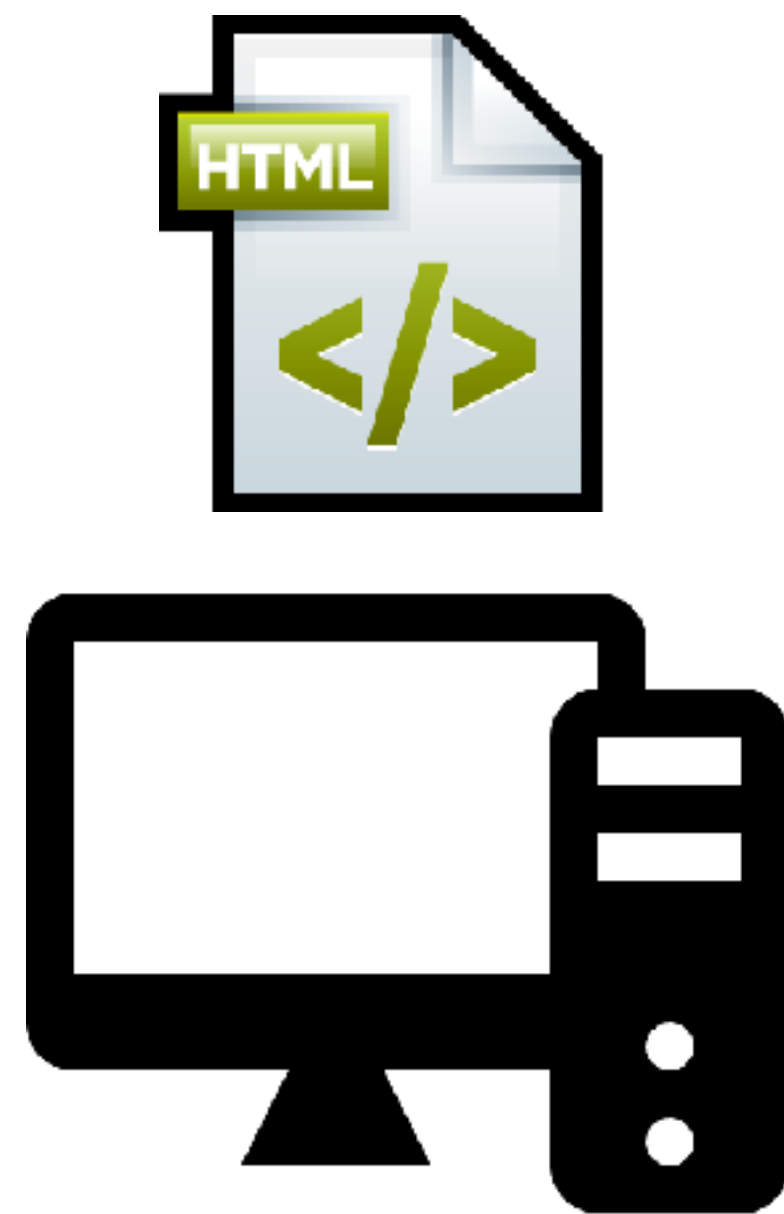
A diagram consisting of 14 horizontal black bars of equal height but varying lengths. The bars are arranged in a descending staircase pattern from top to bottom. The first bar at the top is the longest, extending across most of the frame. Each subsequent bar is shorter than the one above it, starting at a horizontal offset from the left. The bars are evenly spaced vertically. The final bar at the bottom is the shortest, starting at a significant horizontal offset from the left.



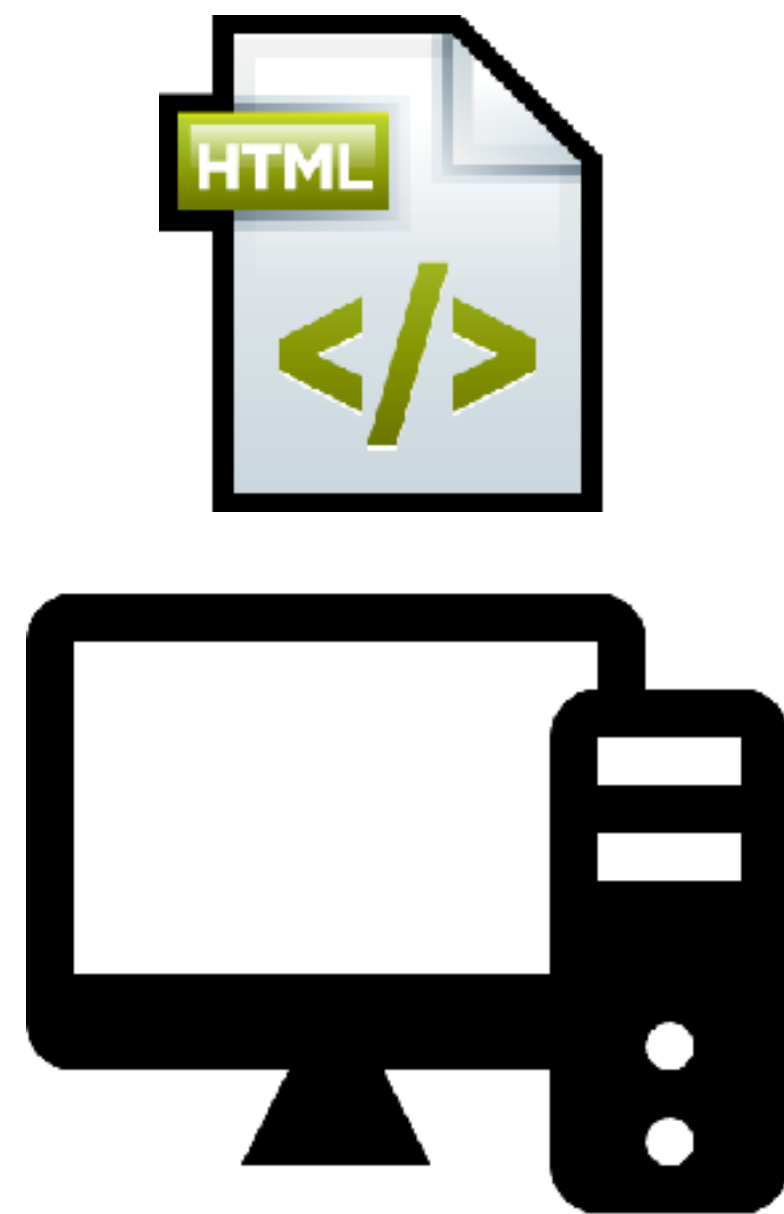


```
<link rel="stylesheet" href="style.css" />
```

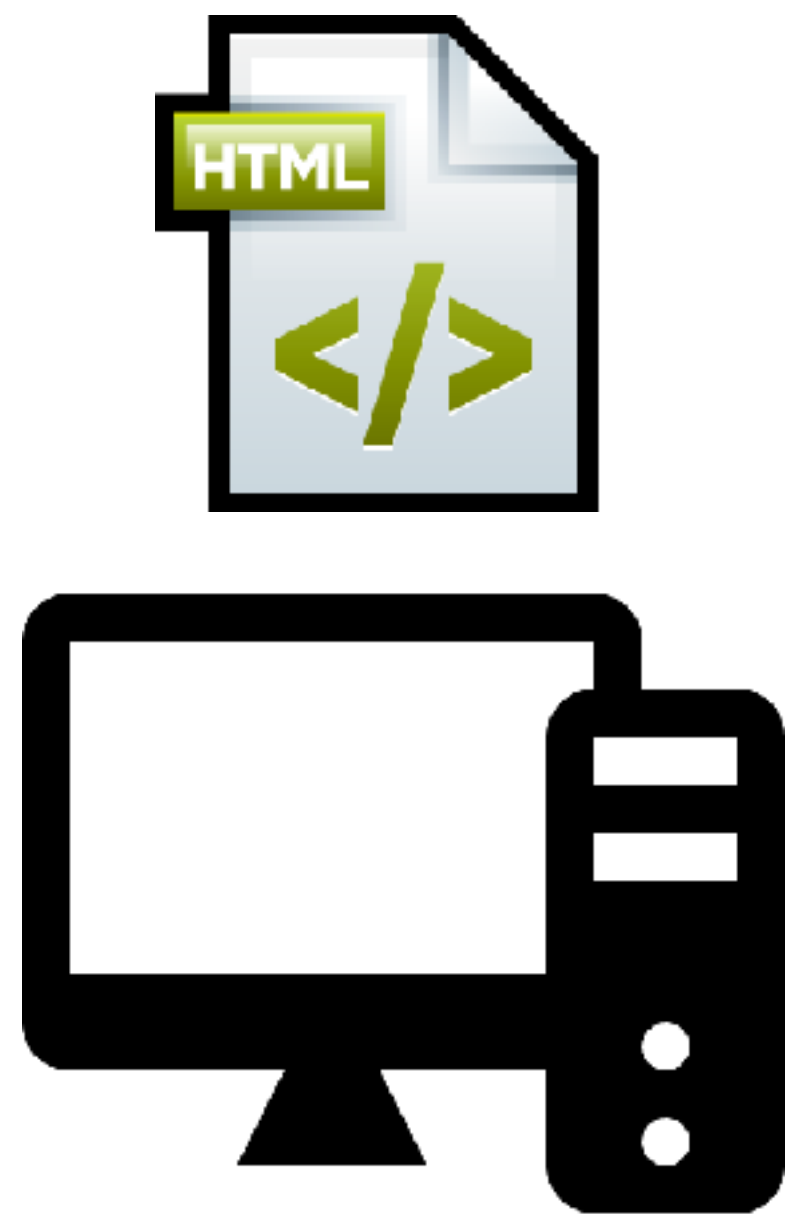




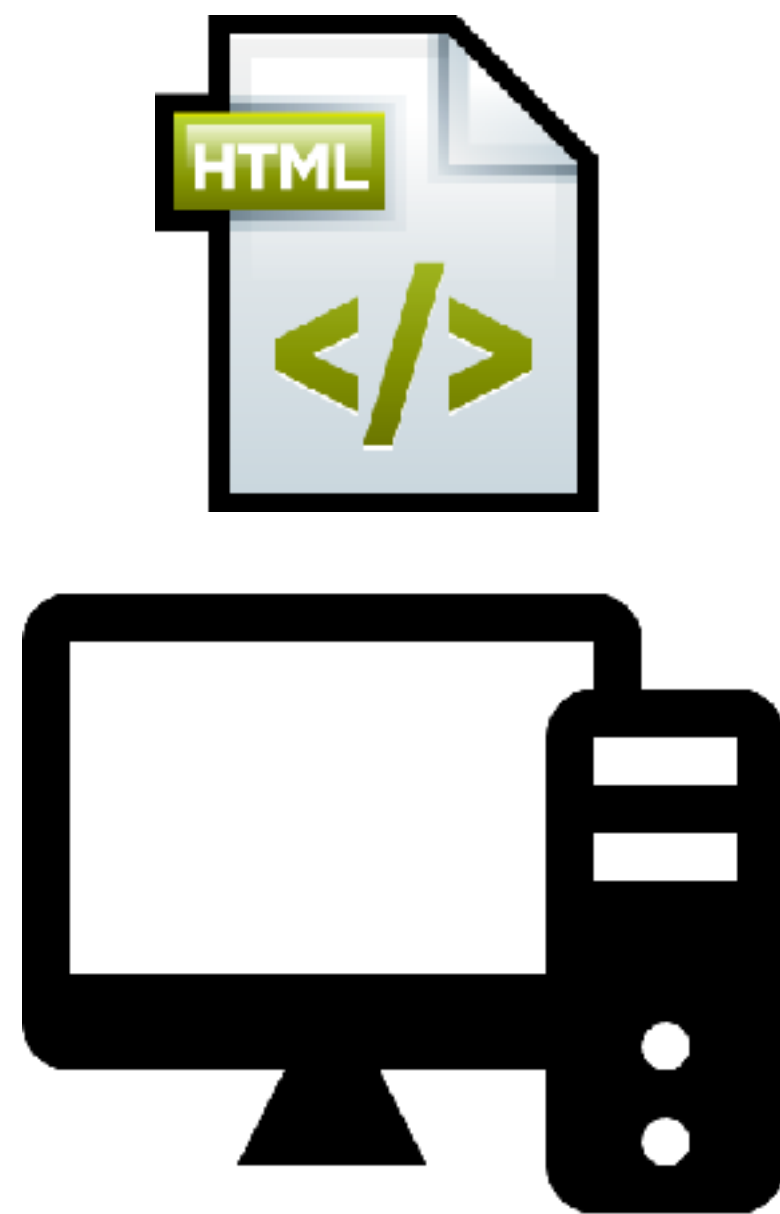
GET /style.css



A diagram consisting of 14 horizontal black bars of equal height but varying lengths. The bars are arranged in a descending staircase pattern from top to bottom. The first bar at the top is the longest, extending across most of the frame. Each subsequent bar is shorter than the one above it, starting at a horizontal offset from the left edge. The bars are evenly spaced vertically. The final bar at the bottom is the shortest, starting at a significant horizontal offset from the left edge.



```
<script src="scripts/app.js"></script>
```

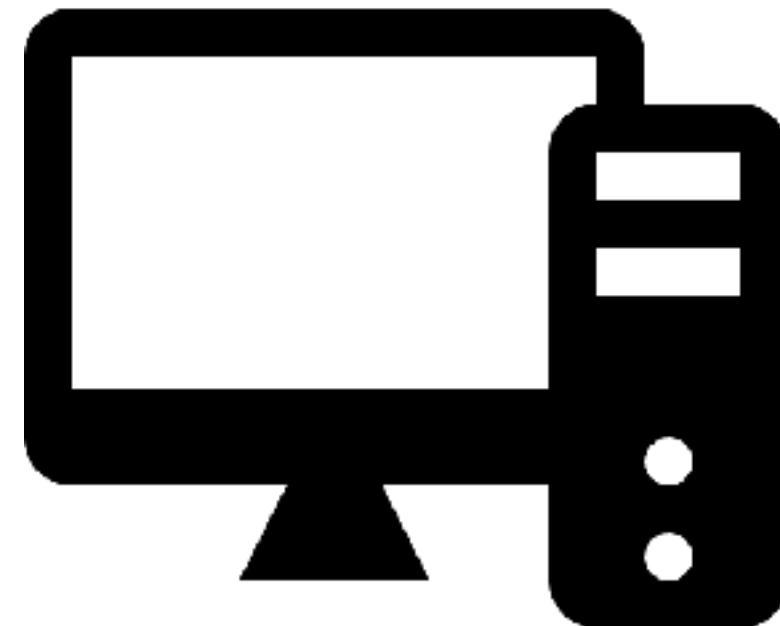


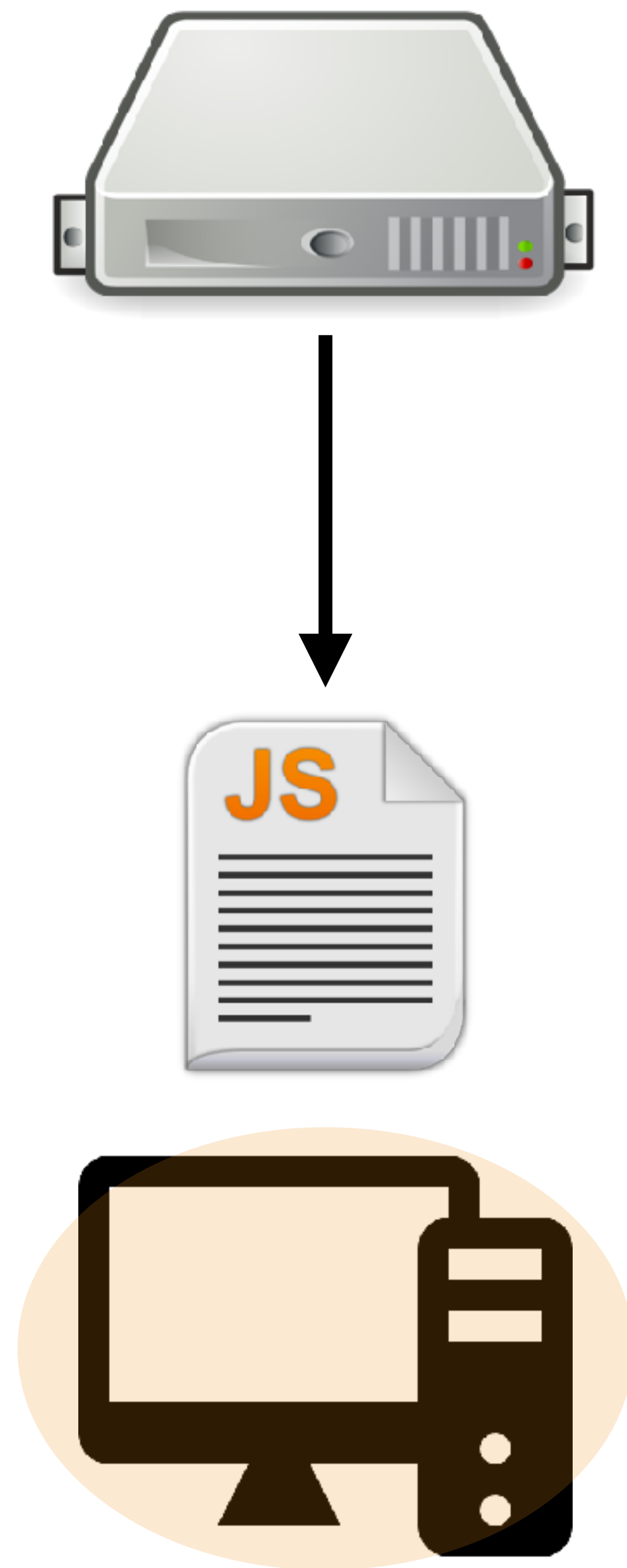
GET /scripts/app.js

```
app.use(express.static(__dirname + '/public'));
```



GET /scripts/app.js

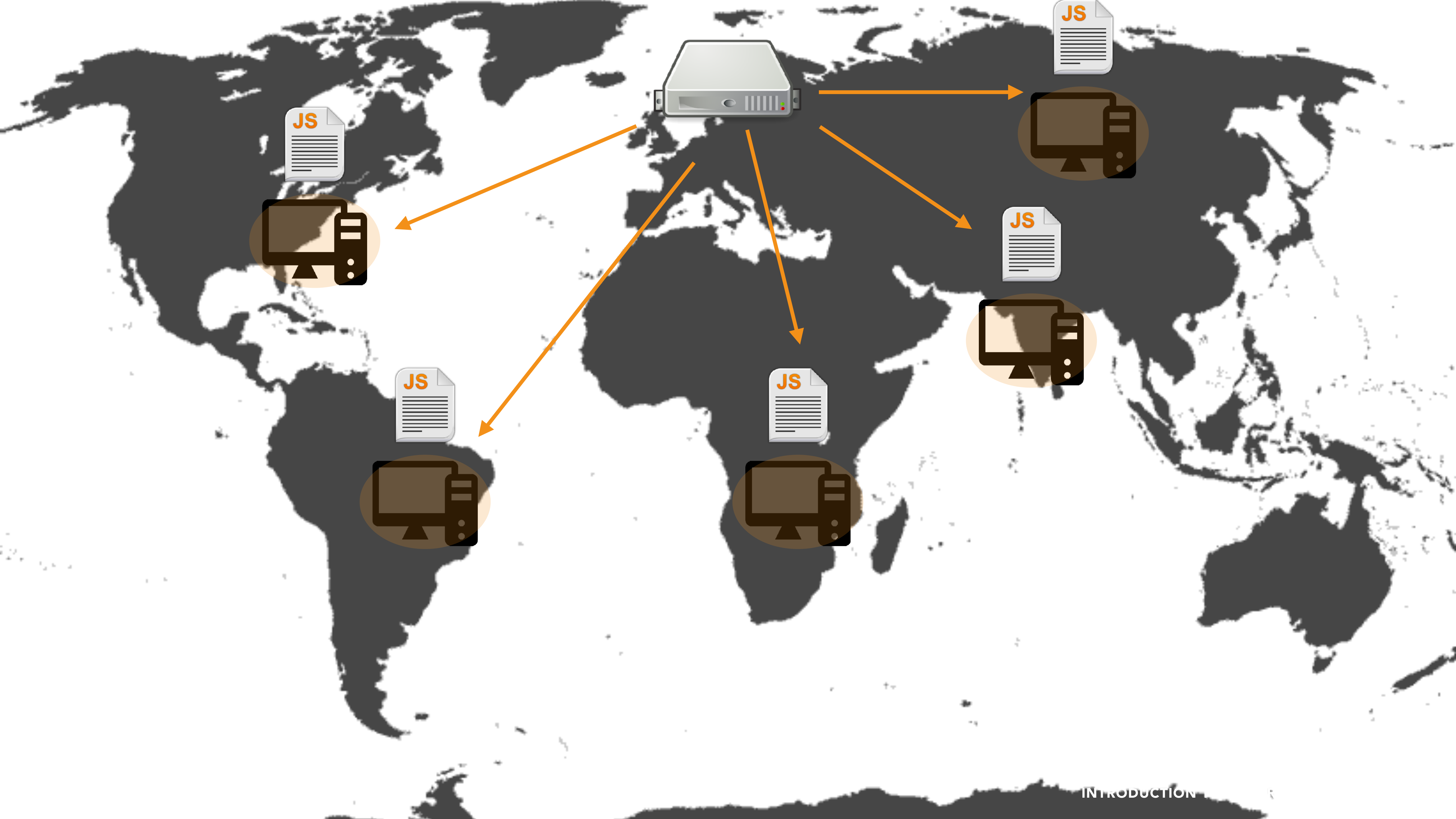




FRONT-END JAVASCRIPT IS A FILE SERVED TO A BROWSER

**IT DOES NOT SHARE ANY CODE
OR FUNCTIONALITY WITH THE
SERVER ITSELF**





CROSS-BROWSER CONCERNS

Does what the competitors do, just slightly different.







USING JAVASCRIPT TO GET CLASSES OF AN ELEMENT

- **Modern browsers:** `getComputedStyle()`
- **Some older browsers:** `currentStyle()`
- **Other browsers:** `runtimeStyle()`
- **Other browsers:** Get class attribute as a string, parse it yourself



```
var classes;  
  
if (DOMElement.getComputedStyle) {  
    classes = DOMElement.getComputedStyle();  
} else if (DOMElement.currentStyle()) {  
    classes = DOMElement.currentStyle();  
} else if (DOMElement.runtimeStyle) {  
    classes = DOMElement.runtimeStyle();  
} else {  
    // ...  
}
```

THINGS BROWSERS DO DIFFERENTLY

- Like, everything.
- How elements get selected.
- How to set styles on elements.
- Available APIs.
- Like, everything.



```
$( 'h1' ).addClass( 'my-class' );
```

jQuery PROVIDES ABSTRACTION

JQUERY IS A JAVASCRIPT MILESTONE

- Released in 2006.
- Nearly single-handedly changed the industry's attitude towards Javascript and the web.
- Still used and useful today.
- So popular that some people know jQuery without any Javascript fundamentals.




```
$ ( ' h1 ' )
```

```
$ ( ' #my-heading ' )
```

```
$ ( ' .control-buttons ' )
```

```
$ ( ' h1 span ' )
```

**GOOGLE WHAT YOU
WANT TO DO**

```
$( '.btn' ).on( 'click', function () {  
    // All that my heart desires.  
});
```

```
$( '.btn' ).on( 'click', function () {  
    // All that my heart desires.  
});
```

```
$( '.btn' ).on( 'click', function () {  
    // All that my heart desires.  
});
```



CLICK
MOUSEDOWN
MOUSEUP
HOVER
CHANGE
KEYPRESS



CLICK
MOUSEDOWN
MOUSEUP
HOVER
CHANGE
KEYPRESS

```
<div class="author-profile">
  <h1 class="author-name">William Gibson</h1>
  <h2 class="genre">Science Fiction</h2>
  <ul class="titles">
    <li class="top-pick">Neuromancer</li>
    <li>Pattern Recognition</li>
    <li>Zero History</li>
  </ul>
</div>
```



```
$( '.author-profile' ).children();
```

```
<div class="author-profile">  
  <h1 class="author-name">William Gibson</h1>  
  <h2 class="genre">Science Fiction</h2>  
  <ul class="titles">  
    <li class="top-pick">Neuromancer</li>  
    <li>Pattern Recognition</li>  
    <li>Zero History</li>  
  </ul>  
</div>
```

```
$( '.author-profile' ).children();
```

```
<div class="author-profile">  
  <h1 class="author-name">William Gibson</h1>  
  <h2 class="genre">Science Fiction</h2>  
  <ul class="titles">  
    <li class="top-pick">Neuromancer</li>  
    <li>Pattern Recognition</li>  
    <li>Zero History</li>  
  </ul>  
</div>
```

```
$( '.author-profile' ).children( '.author-name' );
```

```
<div class="author-profile">  
  <h1 class="author-name">William Gibson</h1>  
  <h2 class="genre">Science Fiction</h2>  
  <ul class="titles">  
    <li class="top-pick">Neuromancer</li>  
    <li>Pattern Recognition</li>  
    <li>Zero History</li>  
  </ul>  
</div>
```



```
$( '.genre' ).parent( );
```

```
<div class="author-profile">  
  <h1 class="author-name">William Gibson</h1>  
  <h2 class="genre">Science Fiction</h2>  
  <ul class="titles">  
    <li class="top-pick">Neuromancer</li>  
    <li>Pattern Recognition</li>  
    <li>Zero History</li>  
  </ul>  
</div>
```

```
$( '.genre' ).parent( );
```

```
<div class="author-profile">  
  <h1 class="author-name">William Gibson</h1>  
  <h2 class="genre">Science Fiction</h2>  
  <ul class="titles">  
    <li class="top-pick">Neuromancer</li>  
    <li>Pattern Recognition</li>  
    <li>Zero History</li>  
  </ul>  
</div>
```

```
$( '.genre' ).siblings( );
```

```
<div class="author-profile">  
  <h1 class="author-name">William Gibson</h1>  
  <h2 class="genre">Science Fiction</h2>  
  <ul class="titles">  
    <li class="top-pick">Neuromancer</li>  
    <li>Pattern Recognition</li>  
    <li>Zero History</li>  
  </ul>  
</div>
```



```
$( '.genre' ).siblings( );
```

```
<div class="author-profile">  
  <h1 class="author-name">William Gibson</h1>  
  <h2 class="genre">Science Fiction</h2>  
  <ul class="titles">  
    <li class="top-pick">Neuromancer</li>  
    <li>Pattern Recognition</li>  
    <li>Zero History</li>  
  </ul>  
</div>
```

```
$( '.titles li' );
```

```
<div class="author-profile">  
  <h1 class="author-name">William Gibson</h1>  
  <h2 class="genre">Science Fiction</h2>  
  <ul class="titles">  
    <li class="top-pick">Neuromancer</li>  
    <li>Pattern Recognition</li>  
    <li>Zero History</li>  
  </ul>  
</div>
```



```
$( '.titles li' ).css({ color: 'red' });
```

```
<div class="author-profile">
  <h1 class="author-name">William Gibson</h1>
  <h2 class="genre">Science Fiction</h2>
  <ul class="titles">
    <li class="top-pick">Neuromancer</li>
    <li>Pattern Recognition</li>
    <li>Zero History</li>
  </ul>
</div>
```

IMPLICIT LOOPING

If a jQuery object represents multiple elements, actions and traversal will be taken for each element.

```
var $titles = $( '.titles li' );  
$titles.css( { width: 200 } );
```

```
var $titles = $( '.titles li' );  
  
$titles.text();
```

```
var $titles = $('.titles li');  
  
$titles.children();
```

EXPLICIT LOOPING

```
var $titles = $('.titles li');  
  
$titles.each(function () {});
```

```
<div id="jukebox">
```

```
  <button>Duran Duran – Hungry Like the Wolf</button>
```

```
  <button>Toto – Africa</button>
```

```
  <button>A Flock of Seagulls – I Ran</button>
```

```
</div>
```



```
<div id="jukebox">  
  <button>Duran Duran – Hungry Like the Wolf</button>  
  <button>Toto – Africa</button>  
  <button>A Flock of Seagulls – I Ran</button>  
</div>
```

```
var $jukeboxChoices = $('#jukebox').find('button');  
$jukeboxChoices.on('click', function () {  
  
});
```

```
<div id="jukebox">  
  <button>Duran Duran – Hungry Like the Wolf</button>  
  <button>Toto – Africa</button>  
  <button>A Flock of Seagulls – I Ran</button>  
</div>
```

```
var $jukeboxChoices = $( '#jukebox' ).find( 'button' );  
  
$jukeboxChoices.on( 'click', function () {  
  var $buttonPressed = $(this);  
});
```



JQREAM - JQUERY RULES EVERYTHING AROUND ME

DOLLA DOLLA BILL Y'ALL \$\$



\$

FIND || CREATE || CAST

FIND

```
$(' .titles li ');
```

CREATE

```
$( '<h1 id="new-heading">Hi everyone!</h1>' );
```


CAST

```
$(this); // this being a DOM element.
```

```
$(document.getElementById( 'jukebox' ));
```

```
$( $( '#jukebox' ) );
```



```
<div id="jukebox">
```

```
  <button>Duran Duran – Hungry Like the Wolf</button>
```

```
  <button>Toto – Africa</button>
```

```
  <button>A Flock of Seagulls – I Ran</button>
```

```
</div>
```

```
<div id="jukebox">  
  <button>Duran Duran – Hungry Like the Wolf</button>  
  <button>Toto – Africa</button>  
  <button>A Flock of Seagulls – I Ran</button>  
</div>
```

```
var $jukeboxChoices = $( '#jukebox' ).find( 'button' );  
$jukeboxChoices.on( 'click', function () {  
  
});
```

```
<div id="jukebox">
```

```
<button>Duran Duran – Hungry Like the Wolf</button>
```

```
<button>Toto – Africa</button>
```

```
<button>A Flock of Seagulls – I Ran</button>
```

```
<button>Pearl Jam – Jeremy</button>
```

```
<button>Destiny's Child – Say My Name</button>
```

```
</div>
```

```
$( '#jukebox' ).on( 'click', 'button', function () {  
  
});
```

RECAP

- **Client-side Javascript library.**

RECAP

- **Client-side Javascript library.**
- **Good for DOM selection, event handling, managing element properties.**

RECAP

- **Client-side Javascript library.**
- **Good for DOM selection, event handling, managing element properties.**
- **jQuery still has a large slot in the industry and is still a better solution to some problems than Angular/React/what have you.**