SWEN1005

MOBILE WEB PROGRAMMING

Session Ten

MOBILE INFORMATION ARCHITECTURE

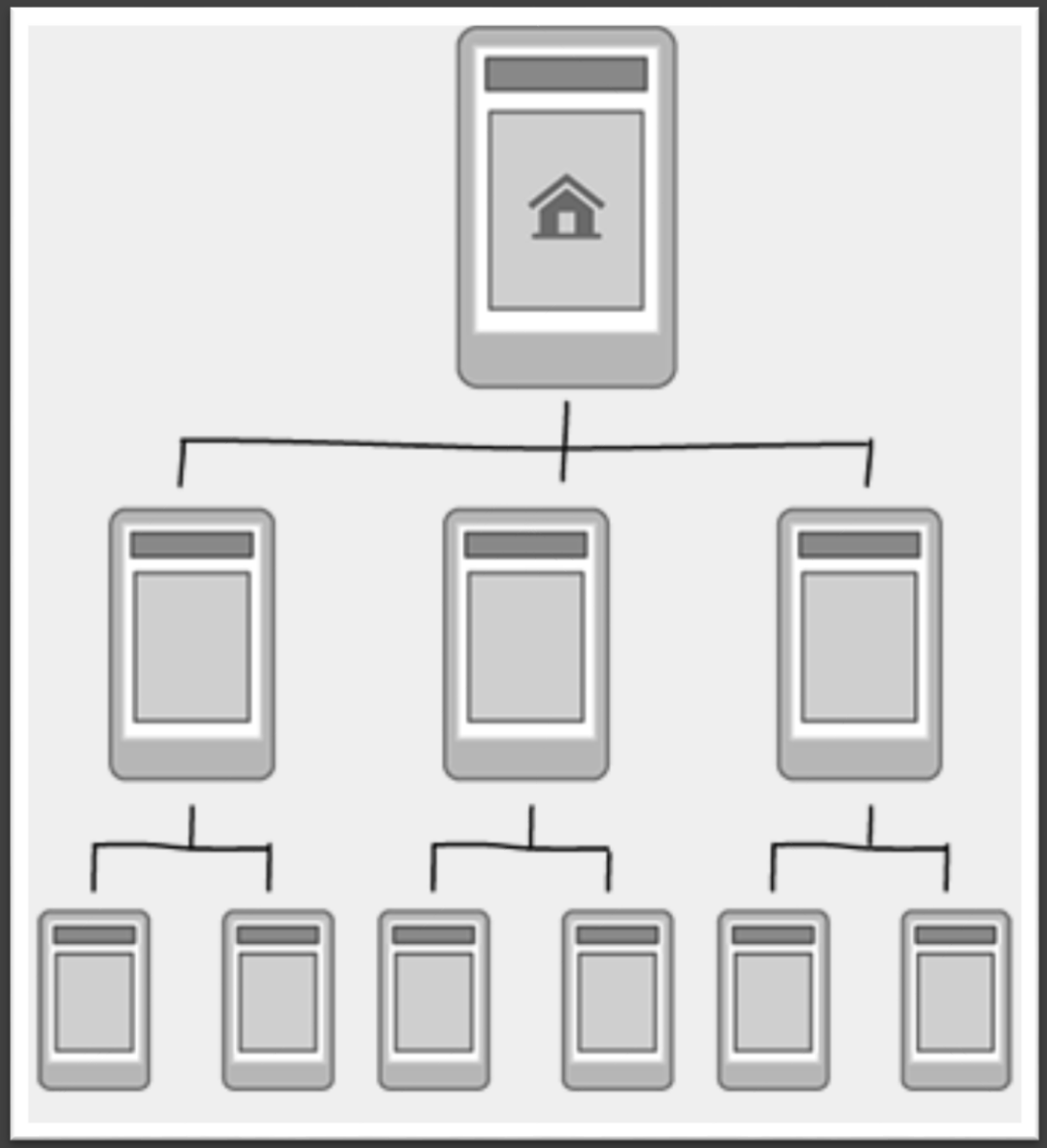
Mobile Information Architecture

- Mobile devices have their own set of Information Architecture patterns, too. While the structure of a responsive site may follow more “standard” patterns, native apps, for example, often employ navigational structures that are tab-based.
- Some of the most popular patterns:
  - Hierarchy
  - Hub & spoke
  - Nested doll
  - Tabbed view

Hierarchy

- The hierarchy pattern is a standard site structure with an index page and a series of sub pages.
- If you are designing a responsive site you may be restricted to this, however introducing additional patterns could allow you to tailor the experience for mobile.

Hierarchy



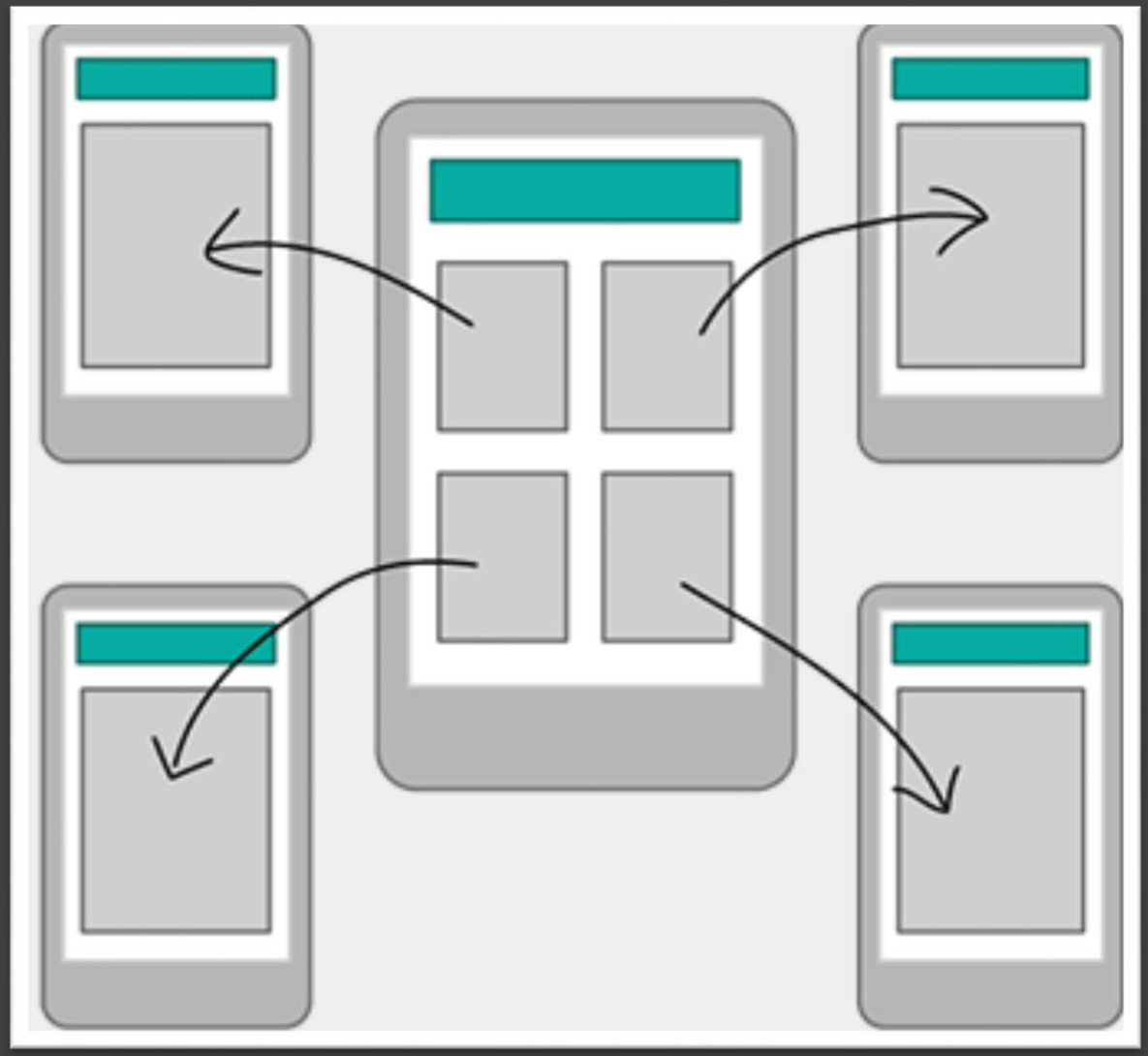
Hierarchy

- The Mobile First approach focuses on the important stuff first:
  - Features and user journeys
  - Helps create great user experiences
- **Pro**
  - Organizes complicated site structures that need to follow a desktop site's structure.
- **Con**
  - Navigation. Multi-faceted navigation structures can present a problem to people using small screens.

Hub & spoke

- A hub and spoke pattern gives you a central index from which users will navigate out. It's the default pattern on Apple's iPhone.
- Users can't navigate between spokes but must return to the hub, instead.
- Typically used on desktops where a workflow is restricted due to technical restrictions.
- This is becoming more prevalent within the mobile landscape due to users being focused on one task, as well as the form factor of the device, making a global navigation more difficult to use.

Hub & spoke



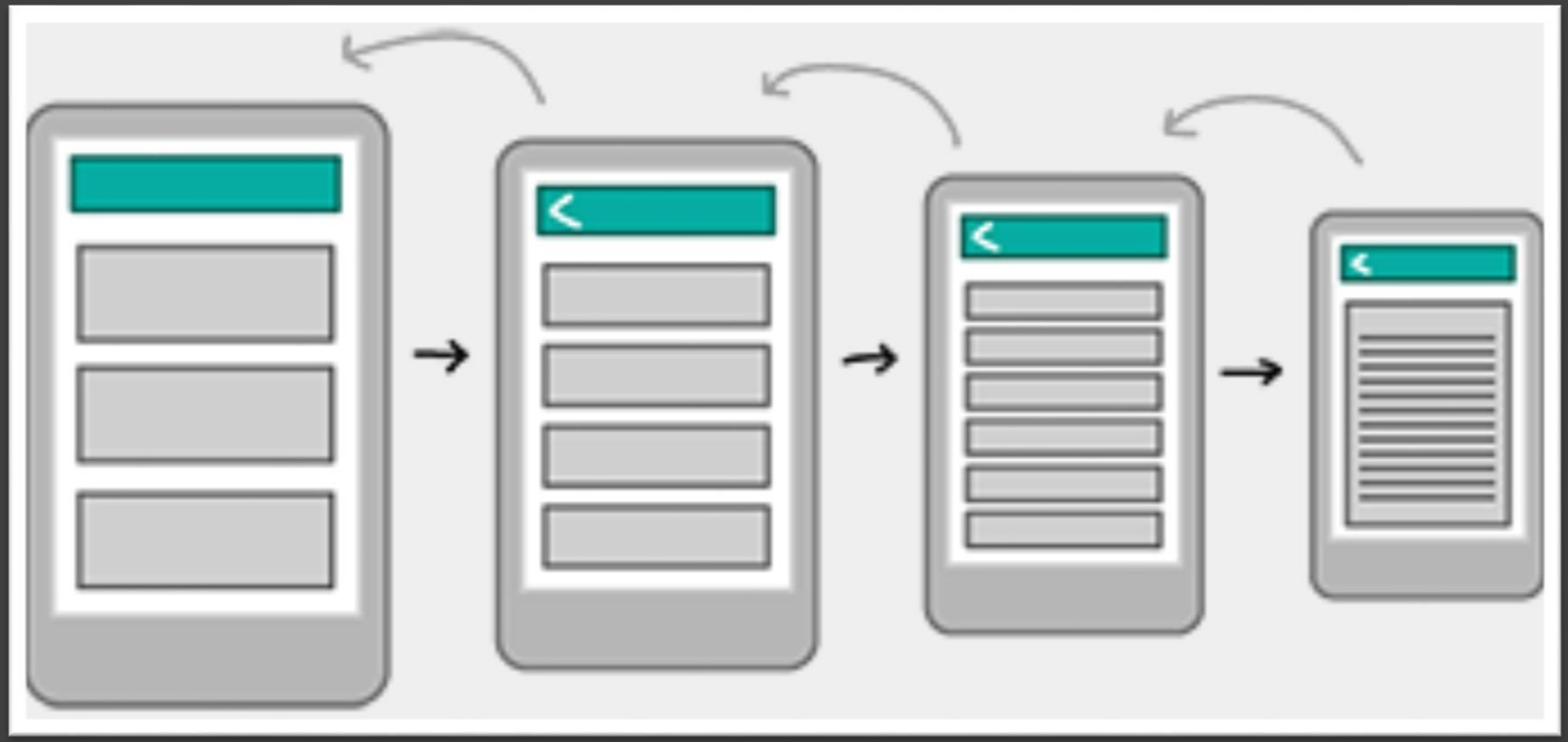
Hub & spoke

- **Pro**
  - Multi-functional tools, each with a distinct internal navigation and purpose.
- **Con**
  - Users cannot multi-task

Nested doll

- The nested doll pattern leads users in a linear fashion to more detailed content.
- When users are in difficult conditions this is a quick and easy method of navigation.
- It also gives the user a strong sense of where they are in the structure of the content due to the perception of moving forward and then back.

Nested doll



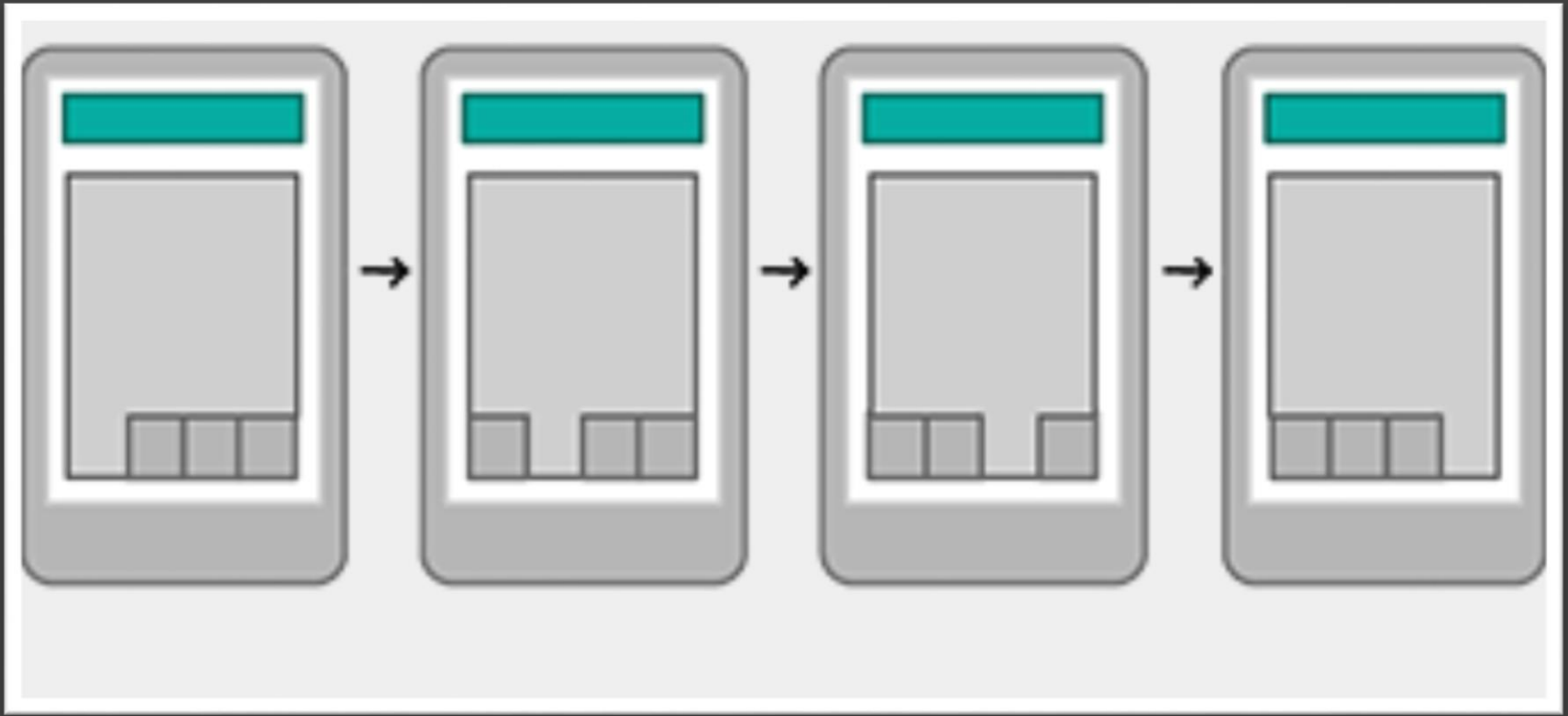
Nested doll

- **Pros**
  - Suitable for apps or sites with singular or closely related topics.
  - Can also be used as a sub section pattern inside other parent patterns, such as the standard hierarchy pattern or hub and spoke.
- **Cons**
  - Users won’t be able to quickly switch between sections
  - Might be unsuitable because it can act as a barrier to exploring content instead of enabling exploration

Tabbed view

- This is a pattern that regular app users will be familiar with.
- It’s a collection of sections tied together by a toolbar menu.
- This allows the user to quickly scan and understand the complete functionality of the app when it’s first opened.

Tabbed view



Tabbed view

- **Pros**
  - Good for tools based apps with a similar theme
  - Facilitates multi-tasking
- **Cons**
  - Complexity in implementation
  - This pattern is best suited to very simple content structures.

Mobile information architecture

- Most desktop oriented sites will be hybrids in this regard.
- However when considering mobile first design it will be necessary to consider the most appropriate architecture for the information you are serving your visitors

Session 10

JAVASCRIPT FRAMEWORKS - JQUERY

What is jQuery

- One important thing to know is that jQuery is just a **JavaScript library**.
- All the power of jQuery is accessed via JavaScript, so having a strong grasp of JavaScript is essential for understanding, structuring, and debugging your code.
- While working with jQuery regularly can, over time, improve your proficiency with JavaScript, it can be hard to get started writing jQuery without a working knowledge of JavaScript's built-in constructs and syntax.

… at least this is what I gathered from jQuery.com

What is jQuery

- **jQuery** is a fast, small, and feature-rich JavaScript library.
- It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers.
- jQuery was created by John Resig and released 01/06
- The most current release is jQuery 3.3.1
  - Jan. 20, 2018

jQuery
write less, do more.

jQuery.com goes on to say…

So what is it?

- It is a library of JavaScript functions
- It is small (~32kb)
- It’s fast
- there are Tons of features
- It’s Open Source

So what is it?

- Provides HTML document traversal and manipulation
- Can add animation
- Uses Ajax
- It’s extensible
- It’s very compatible – Cross Browser
- Provides event handling

Are there other frameworks?

- Tons…
  - Sproutcore - http://sproutcore.com/
  - Cappuccino - http://cappuccino.org/
  - JavaScriptMVC - http://javascriptmvc.com/
  - Asana Luna - http://asana.com/luna/
  - Backbone.js - http://documentcloud.github.com/backbone/
  - Qooxdoo - http://qooxdoo.org/

Who uses jQuery

- Google
- Microsoft
- Amazon
- IBM
- Twitter
- Dell

So how do we begin…

- We have to mindful of two things
- Download the file to your computer
  - https://jquery.com/download/
- Link to a remotely hosted version from a reliable and established website (as at April 2018)
  - Google CDN: https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js
  - Microsoft CDN: https://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.3.1.min.js
  - jQuery CDN: jQuery - https://code.jquery.com/jquery/

Download the file to your computer

- https://jquery.com/download/
  - The main benefit is that if you are offline, the file will still be there on your computer when you are developing your code

Link to a remotely hosted version

- These externally linked files are typically the most up to date.
- However, it requires that you are online, and also takes a few extra milliseconds which can, in theory, slow down your site a little bit.
- Still, once your page goes ‘live’, you should use the version linking to an external file

Advantage of using a link

- Many users already have downloaded jQuery from Google or Microsoft when visiting another site. As a result, it will be loaded from cache when they visit your site, which leads to faster loading time.
- Also, most CDN's will make sure that once a user requests a file from it, it will be served from the server closest to them, which also leads to faster loading time.

So are we ready to code?

```
<html>

<head>

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js"></script>

<script>

$(document).ready(function(){

    $("button").click(function(){

        $("ul li:first-child").hide();

    });

});

</script>
```

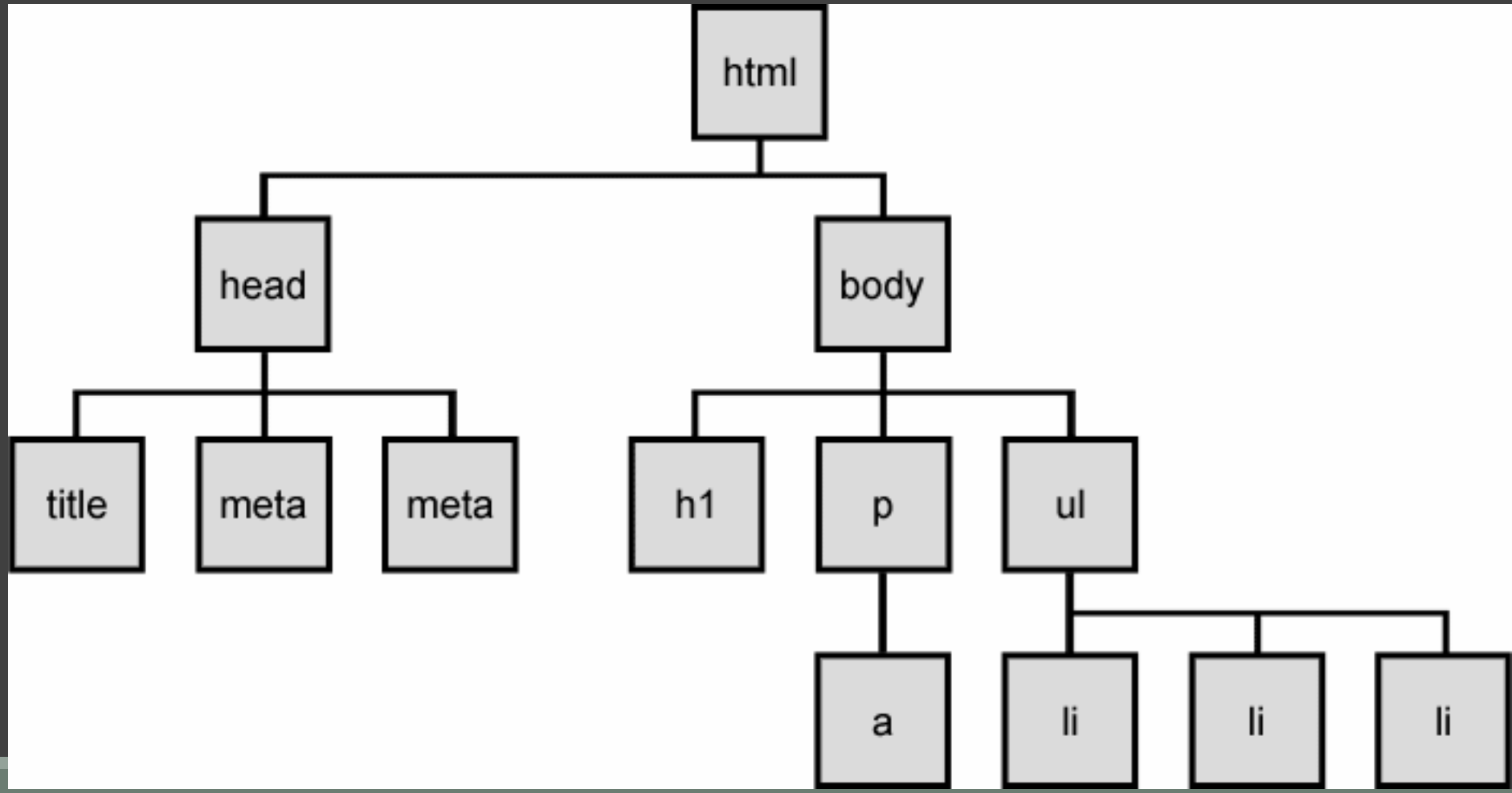
Document traversal and manipulation

- The Document Object Model (DOM) is a cross-platform and language-independent convention for representing and interacting with objects in HTML documents.
- Aspects of the DOM (such as its "Elements") may be addressed and manipulated within the syntax of the programming language in use.” Wikipedia

How does jQuery interact with the DOM?

- **Identification:** how do I obtain a reference to the node that I want.
- **Traversal:** how do I move around the DOM tree.
- **Node Manipulation:** how do I get or set aspects of a DOM node.
- **Tree Manipulation:** how do I change the structure of the page.

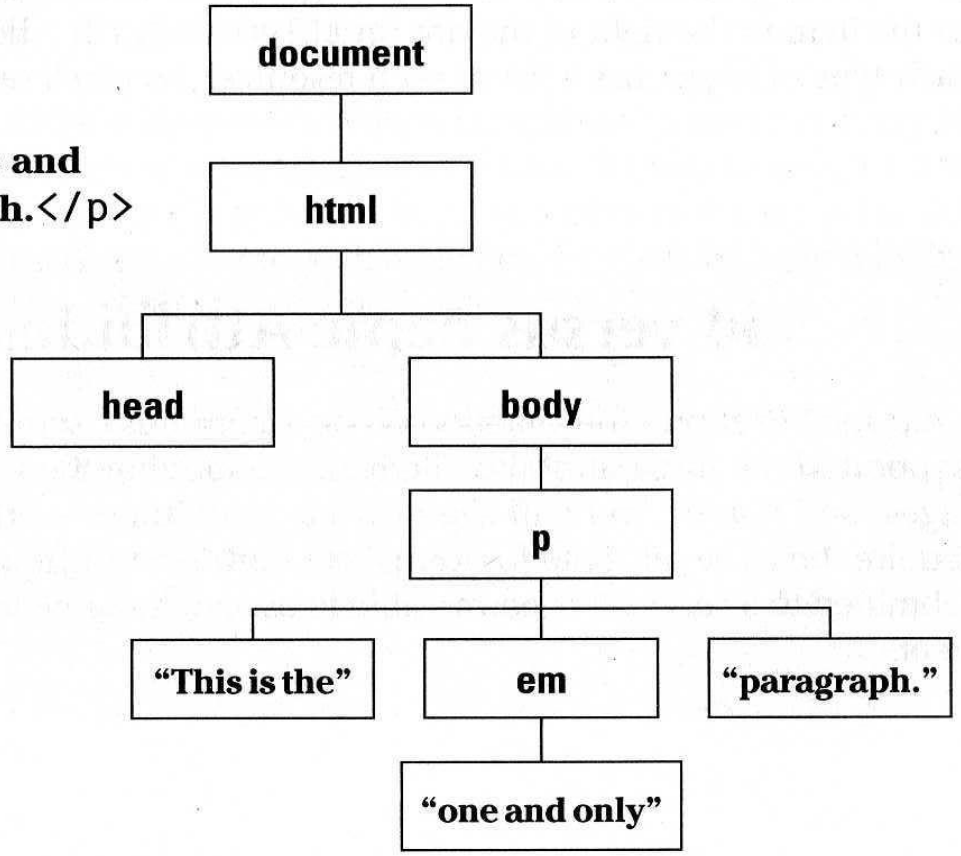
The DOM tree



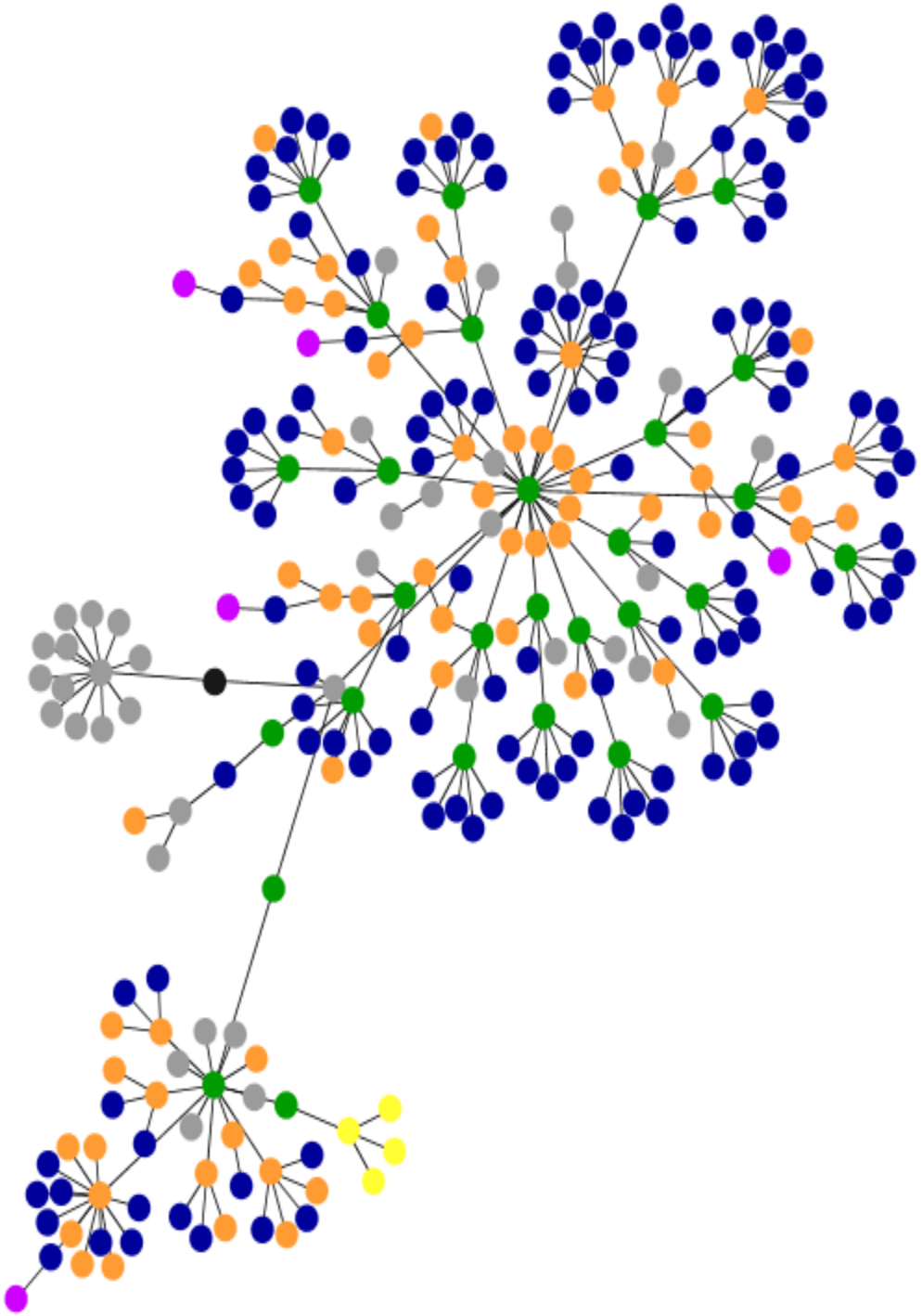
An example of…

A simple HTML document node tree.

```
<html>
  <head></head>
  <body>
      <p>This is the <em>one and
        only</em>  paragraph.</p>
  </body>
</html>
```



…but the DOM tree is not always simple



A few things to note about the DOM tree

- The DOM is everything you write in your html documents, images, css, all your tags, everything
- There are a million different ways to accomplish things within it and many different doctypes and uppercase and lowercase are allowed attributes that sometimes need quotes, and other times don't
- jQuery is coded around all those inconsistencies
- jQuery can modify the DOM, but it cant do so until the DOM is ready

Important

- The DOM is loaded top to bottom
- The DOM is "ready" when everything on the page has loaded.
- For this reason you should include your scripts at the bottom of the page for best performance

This is why we start here…

```
<html>

<head>

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js"></script>

<script>

$(document).ready(function(){

    $("button").click(function(){

        $("ul li:first-child").hide();

    });

});

</script>
```

Starting the document ready event

- This is to prevent any jQuery code from running before the document is finished loading (is ready).
- It is good practice to wait for the document to be fully loaded and ready before working with it. This also allows you to have your JavaScript code before the body of your document, in the head section.
- Here are some examples of actions that can fail if methods are run before the document is fully loaded:
  - Trying to hide an element that is not created yet
  - Trying to get the size of an image that is not loaded yet

Document read event

- There is a shorter but less explicit version:
  - $(function(){

```
// jQuery methods go here...

});
```

jQuery Syntax

The jQuery syntax is tailor-made for **selecting** HTML elements and performing some **action** on the element(s).

Basic syntax is: **$(*selector*).*action*()**

- A $ sign to define/access jQuery
- A (*selector*) to "query (or find)" HTML elements
- A jQuery *action*() to be performed on the element(s)

jQuery Syntax – Examples

- $(this).hide() - hides the current element.
- $("p").hide() - hides all <p> elements.
- $(".test").hide() - hides all elements with class="test".
- $("#test").hide() - hides the element with id="test".

jQuery Selectors

- jQuery selectors allow you to select and manipulate HTML element(s).
- jQuery selectors are used to "find" (or select) HTML elements based on their name, id, classes, types, attributes, values of attributes and much more. It's based on the existing CSS Selectors, and in addition, it has some own custom selectors.
- All selectors in jQuery start with the dollar sign and parentheses: $().

Which selector was used in this case?

```
$(document).ready(function(){
    $("button").click(function(){
        $("p").hide();
    });
});
```

Which selector was used in this case?

```
$(document).ready(function(){
    $("button").click(function(){
        $("#test").hide();
    });
});
```

Which selector was used in this case?

```
$(document).ready(function(){
    $("button").click(function(){
        $(".test").hide();
    });
});
```

Some more selectors

$(this)	Selects the current HTML element
$("p.intro")	Selects all <p> elements with class="intro"
$("p:first")	Selects the first <p> element
$("ul li:first")	Selects the first element of the first
$("ul li:first-child")	Selects the first element of every
$("[href]")	Selects all elements with an href attribute
$("a[target='_blank']")	Selects all <a> elements with a target attribute value equal to "_blank"

Some more selectors

$("a[target!='_blank']")	Selects all <a> elements with a target attribute value NOT equal to "_blank"
$(":button")	Selects all <button> elements and <input> elements of type="button"
$("tr:even")	Selects all even <tr> elements

Click here to try and see more selectors

Functions In a Separate File

- If your website contains a lot of pages, and you want your jQuery functions to be easy to maintain, you can put your jQuery functions in a separate .js file.
- Sometimes it is preferable to place them in a separate file, like this (use the src attribute to refer to the .js file):
  - <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js"></script>
    <script src="my_jquery_functions.js"></script>

jQuery Event Methods

- All the different visitor's actions that a web page can respond to are called events.
- An event represents the precise moment when something happens.

Examples:

- moving a mouse over an element
- selecting a radio button
- clicking on an element

- The term **"fires/fired"** is often used with events. Example: "The keypress event is fired, the moment you press a key".

Some DOM events

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

jQuery Syntax For Event Methods

- $("p").click(function(){
  // action goes here!!
})

click()

- The click() method attaches an event handler function to an HTML element.
- The function is executed when the user clicks on the HTML element.
- The following example says: When a click event fires on a <p> element; hide the current <p> element:

Example

```
$("p").click(function(){
    $(this).hide();
});
```

dblclick()

- The dblclick() method attaches an event handler function to an HTML element.
- The function is executed when the user double-clicks on the HTML element:

Example

- ```
  $("p").dblclick(function(){
      $(this).hide();
  });
  ```
  ```

mouseenter()

- The mouseenter() method attaches an event handler function to an HTML element.
- The function is executed when the mouse pointer enters the HTML element:

Example

```
$("#p1").mouseenter(function(){
    alert("You entered p1!");
});
```

mouseleave()

- The mouseleave() method attaches an event handler function to an HTML element.
- The function is executed when the mouse pointer leaves the HTML element:

Example

```
$("#p1").mouseleave(function(){
    alert("Bye! You now leave p1!");
});
```

mousedown()

- The mousedown() method attaches an event handler function to an HTML element.
- The function is executed, when the left, middle or right mouse button is pressed down, while the mouse is over the HTML element:

Example

- ```
  $("#p1").mousedown(function(){
      alert("Mouse down over p1!");
  });
  ```
  ```

hover()

- The hover() method takes two functions and is a combination of the mouseenter() and mouseleave() methods.
- The first function is executed when the mouse enters the HTML element, and the second function is executed when the mouse leaves the HTML element:

Example

```
$("#p1").hover(function(){
    alert("You entered p1!");
},
function(){
    alert("Bye! You now leave p1!");
});
```

focus()

- The focus() method attaches an event handler function to an HTML form field.
- The function is executed when the form field gets focus:

Example

```
$("input").focus(function(){
    $(this).css("background-color", "#cccccc");
});
```

blur()

- The blur() method attaches an event handler function to an HTML form field.
- The function is executed when the form field loses focus:

Example

- ```
  $("input").blur(function(){
      $(this).css("background-color", "#ffffff");
  });
  ```
  ```

The on() Method

- The on() method attaches one or more event handlers for the selected elements.
- Attach a click event to a <p> element:

Example

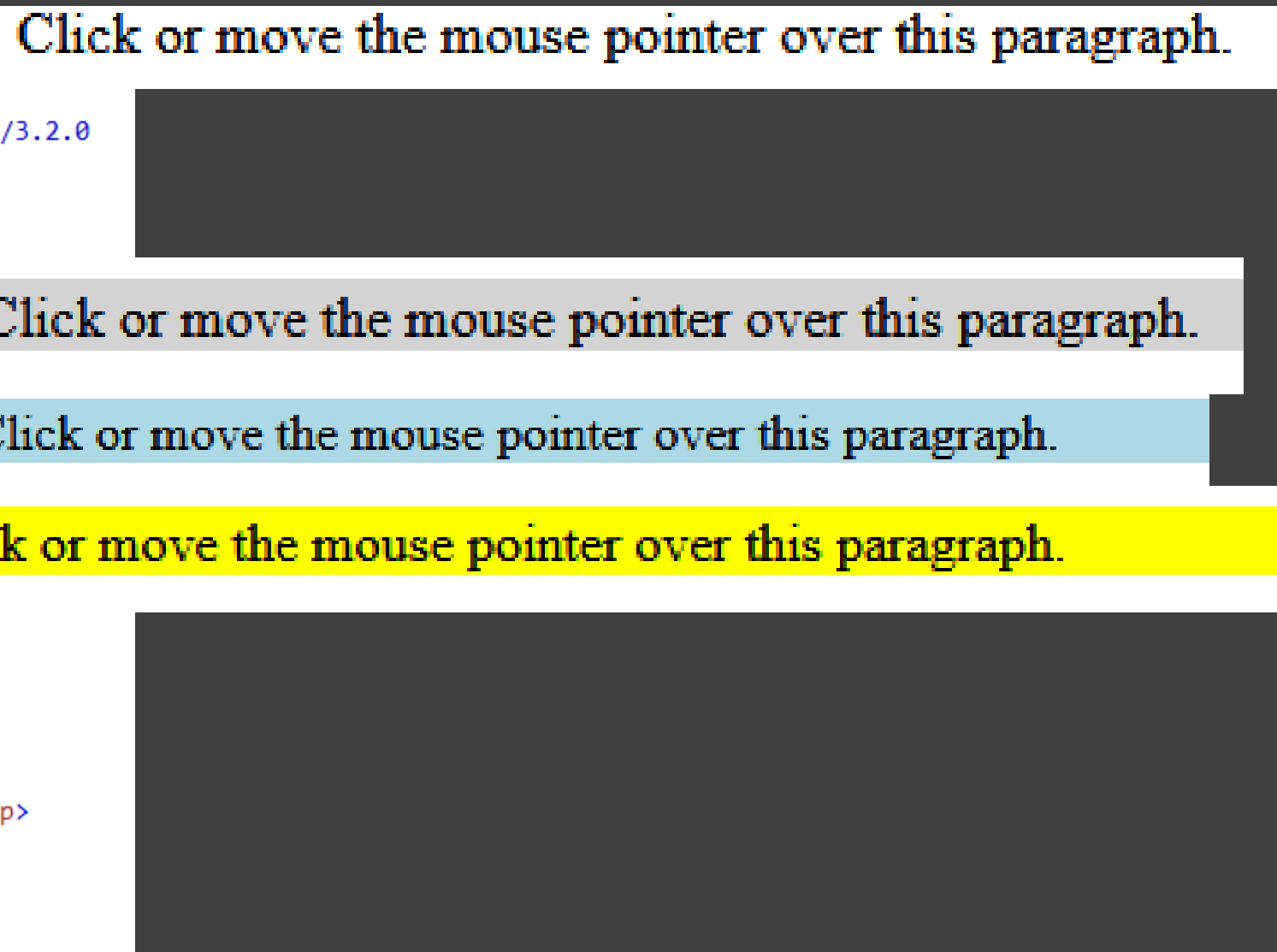
```
$("p").on("click", function(){
    $(this).hide();
});
```

Multiple event handlers

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.0
/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("p").on({
        mouseenter: function(){
            $(this).css("background-color", "lightgray'
        },
        mouseleave: function(){
            $(this).css("background-color", "lightblue
        },
        click: function(){
            $(this).css("background-color", "yello
        }
    });
});
</script>
</head>
<body>

<p>Click or move the mouse pointer over this paragraph.</p>

</body>
</html>
```



jQuery Effects

- jQuery Hide/Show
- jQuery Fade
- jQuery Slide
- jQuery Animate
- jQuery stop()
- jQuery Callback
- jQuery Chaining

jQuery HTML

- jQuery Get
- jQuery Set
- jQuery Add
- jQuery Remove
- jQuery CSS Classes
- jQuery css()
- jQuery Dimensions

Quick Review

- **Mobile Information Architecture**
  Hub and Spoke
  - Hierarchy
  - Nested Doll
  - Tabbed View
- **JavaScript Frameworks**
  - jQuery
    - Syntax
    - Selectors
    - Events

Thank you

THE END