Asynchronous Updates

Original web

- Client sends request; Server responds; Client displays
- For any update of page:
  - new request sent from client to server
  - server has to respond with complete page, HTML, styling etc
  - client has to render the page again from scratch
- Potential issues
  - Server load: lots of redundant data to be sent each time
  - server-rendering -> more work
  - slow updates: load full page, re-render

Asynchronous Updates

- Update only part of the page
    - Load extra data in the background after the main page has been loaded and rendered
- Quick response on main page: better user experience
- Request for update can ask for just minimal data to refresh part of a page
    - Example: show user a form to select animal
    - request data about animal alone from server - no need for HTML or other styling
    - refresh only one <div> in the page with text about the animal
- Originally seen as AJAX, now many variants

Core idea: refresh part of the document based on asynchronous (background) queries to server

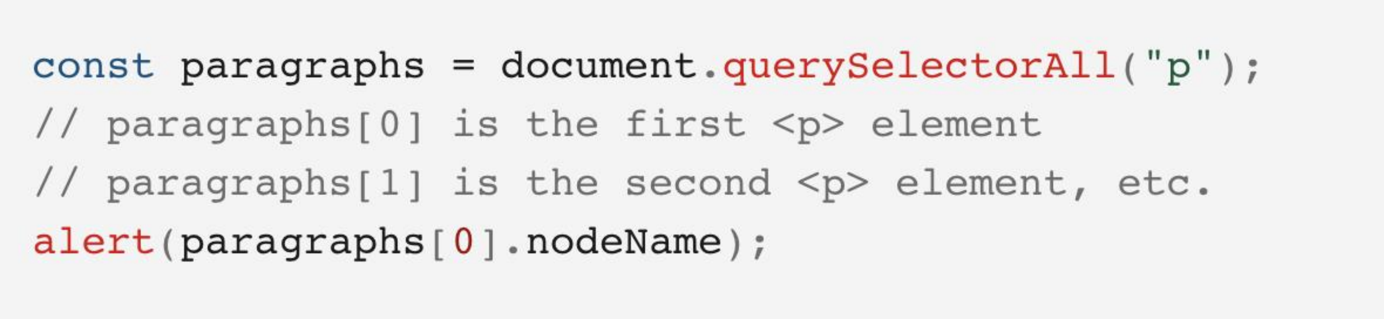
DOM

Document Object Model

- Programming interface for web documents
- What is a web-page?
  - HTML source? Rendered image?
- DOM is an abstract model (tree structure) of the document
- Object-oriented allows manipulation like known objects
- Tightly coupled with JavaScript in most cases
  - Can also be manipulated from other languages (Python has xml DOM interface for example)

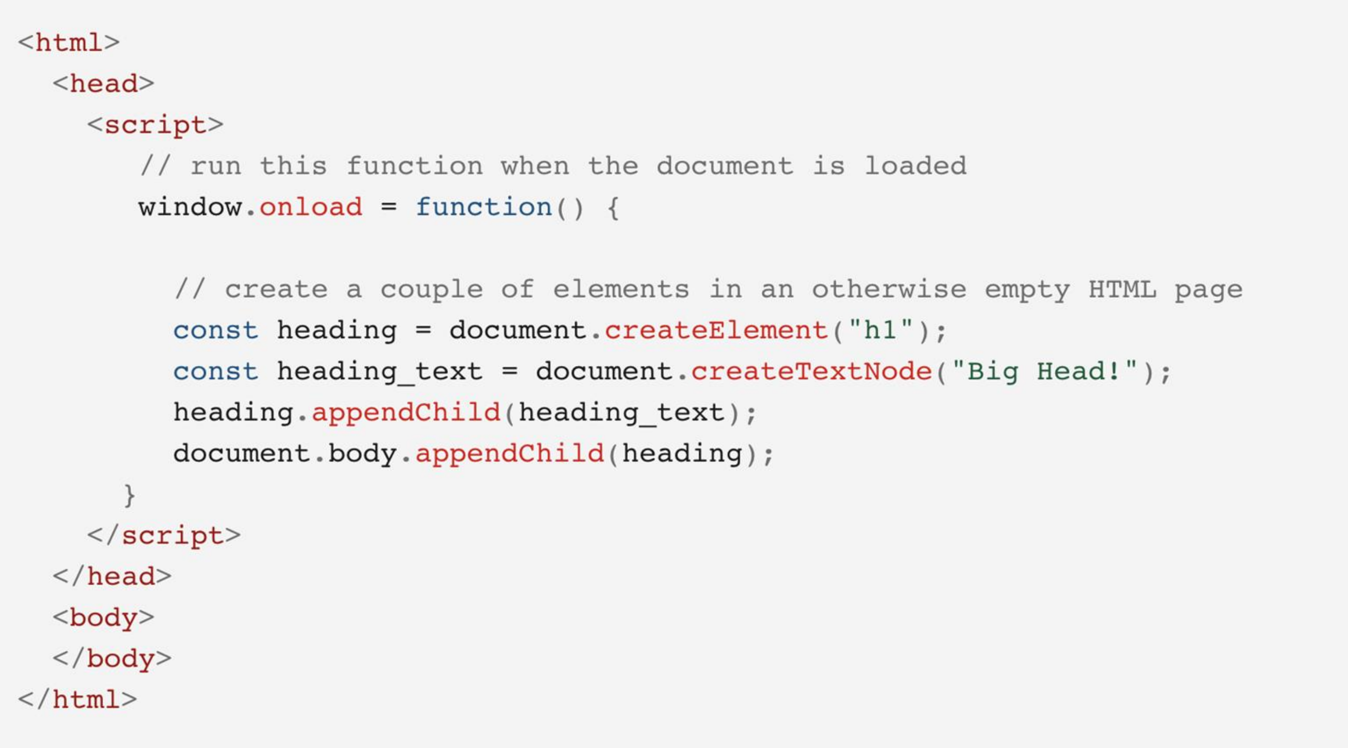
Example usage

https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction



```
const paragraphs = document.querySelectorAll("p");
// paragraphs[0] is the first <p> element
// paragraphs[1] is the second <p> element, etc.
alert(paragraphs[0].nodeName);
```

Manipulating the DOM



```
<html>
  <head>
    <script>
       // run this function when the document is loaded
       window.onload = function() {

         // create a couple of elements in an otherwise empty HTML page
         const heading = document.createElement("h1");
         const heading_text = document.createTextNode("Big Head!");
         heading.appendChild(heading_text);
         document.body.appendChild(heading);
      }
    </script>
  </head>
  <body>
  </body>
</html>
```

Component building

- DOM is manipulatable through programs
- Bring concepts of programming into DOM manipulation:
  - Objects
  - Composition of objects, inheritance
  - Loops, iterators, programmable placement
- Lot more flexibility in front-end
- Lot more *complexity* in front-end

Summary

- Asynchronous updates opened up front-end development
- Many new frameworks, technologies
- Beyond scope of this course
    - But essential knowledge for someone interested in app development