Web Technologies Workshop

Why is this useful?

Web Technology

Web technology is a core component of modern software. Programs communicate over networks, and services render results in web browsers for a wide variety of domains, including Bioinformatics. How many professors and laboratories have web sites serving their applications and databases? How many repositories do we as both data consumers and producers interact with as central parts of our daily work? How many people use Entrez (http://www.ncbi.nlm.nih.gov/gquery/), EBI (http://www.ebi.ac.uk/services), UniProt (http://www.uniprot.org/), KEGG (http://www.genome.jp/kegg/), PFAM (http://pfam.xfam.org/), and other bioinformatics databases as a web service?

JavaScript

Within the last ten years, JavaScript has become a powerful and popular fully featured programming language, in addition to being the only programming language available for general consumption online. It defines logic for both automating and interacting with a webpage. With these, we can build accessible representations of data for consumption by both technical and non-technical users. With HTML5 Graphics standards, web technology has become a major competitor in data visualization.

JavaScript is an asynchronous, event driven language, supporting both object-oriented, imperative programming as well as functional programming. It is an active domain of programming language research, with multiple implementations, such as V8 (WebKit and Node.js), SpiderMonkey (Firefox), and Chakra (Internet Explorer). Given its monopoly on the web, it's valuable to be familiar with it. Given some of its design quirks, idioms and popular libraries are difficult for beginners to grasp, it is useful to have a guide through the language.

Topics

- **JavaScript for general computing**
  - **What kind of language is JavaScript?**
    - **The difference between the client and the server**
  - **What kind of paradigms does it support?**
    - **Imperative**
      - Scripting
      - Prototypal Object Oriented
    - **Functional**

        - Functions are First Class Citizens
        - Callback Chains
    - **Basic building blocks**
        - **Numerics**
        - **Strings**
        - **Arrays, Objects, and JSON**
        - **"Classes" and Prototypal Inheritance (Time permitting)**
- **JavaScript Document Object Model APIs and jQuery**
    - **Searching and manipulating the DOM**
    - **Building structures programmatically (jQuery)**
        - Build a table dynamically vs statically
- **Asynchronous, Event-driven programming and Ajax**
    - **Event-based flow of control**
        - **Loading data from the server**
        - **Reacting to user actions**
- **HTTP and more Ajax**
    - **More about asynchronously sending and receiving information.**
    - **Posting data**
    - **Getting data**
    - **Cross-Origin Policy, JSONP, and proxy-ing requests**
- **Tools for bioinformatics programmers**
    - **BioJS**
        - **Sequence View and other select entries from http://registry.biojs.net/ (http://registry.biojs.net/)**
    - **General purpose libraries I use (besides jQuery)**
        - **D3**
        - **Highcharts**
        - **LoDash**
        - **AngularJS (Time permitting)**
- **Web Application Architecture**

    - **Something other than CGI**

      CGI is a bare-bones, raw socket-like method of writing web applications. There are more abstracted, modern methods for building them in just about every language in active use for rapid development

    - **"What does this page do?"**

      -

    - **Routing and Web Service endpoints**

      -

    - **REST architecture**

      -

- ## Report Building
    - Tools by language
        - **Python**
            - **Pweave**
            - **IPython Notebooks**
            - **Templating engines**
        - **R**
            - **knitr**
            - **Sweave**

Many of these topics will be related, but I've ordered the topics in a waterfall so that the topics flow into each other logically.

I'd be available to answer questions during the final project month as well. I worked with four or five groups answering questions and helping with technical problems last year.