Tools

SWE 432, Fall 2016
Design and Implementation of Software for the Web
Today

• Before we can really make anything, there’s a bunch of technical stuff to get out of the way

• Tools make our lives so much easier

  • Actually, this is what a lot of software engineering research is - making better tools for making software!

• Today we’ll cover version control, and some really basic hello world stuff in HTML* and Javascript*

  *See next 2 lectures for a lot more detail
Version Control
We’ve always had some kind of version control

Carbon copies?
Git

• One of the latest generation of Version Control Systems (VCS)

• Came out of the Linux development community

• Fast

• Supports non-linear development

• Fully distributed (no need for “central” server)

• Able to handle large projects effectively
Linear vs Nonlinear Development

Linear history

Non-linear history
Distribution Model

Centralized Model

Examples:
CVS, Subversion, Perforce
Distribution Model

Centralized Model

Examples:
CVS, Subversion, Perforce

Distributed Model

Examples:
Git, Mercurial
No central point of failure
No latency when doing most operations
GitHub

• You don’t need a central server to use git

• But it sure makes it easier

• GitHub: Free open source hosting site
  • For students, free private repositories too!

• Set up a user account

• Create a repository

• Add collaborators
Git, High level

- **Workspace**: E.g. The files you see
- **Index**: Git's local record of changes
- **Local repository**: Git's local repository
- **Remote repository**: E.g. GitHub

**Commands**:
- **add**
- **commit**
- **commit -a**
- **push**
- **pull**
- **fetch**
- **checkout**
- **diff**
Typical Use Case

Create Project & Repo
Push to Github
Edit/commit
Edit/commit
Edit/commit
Pull/push

Clone from Github
Edit/commit/push
Edit/commit
Edit/commit
Pull/push

Github
Git Demo

https://try.github.io/levels/1/challenges/1

Download and install git:
https://git-scm.com/downloads

Also: Git Cheat Sheet
Chrome Developer Tools
+ HTML + JavaScript
Chrome Developer Tools

• A suite of tools built into Chrome (Safari and other browsers have similar tools) to help with debugging web apps

• Key features:
  • View the HTML elements that make up the current page
  • Access a JavaScript Console
  • View all supplemental files loaded by a page
  • View all outbound requests that a page made
  • Plus more (we’ll skip for now)

• Let’s all open Chrome together now!
Accessing Chrome Dev Tools
Inspecting HTML Elements
After highlighting an element in the browser, the “elements” section filters to highlight the code that generates that element.
Works both ways: Highlight some HTML in the Elements panel, see the corresponding part of the page be highlighted.
You can even change the HTML or CSS!
You can even change the HTML or CSS!
You can even change the HTML or CSS!
Using the JavaScript Console
Welcome to the Department of Computer Science. The department offers BS, MS, and PhD programs in Computer Science; MS programs in Software Engineering, Information Systems, and Information Security and Assurance; and a BS program in Applied Computer Science with various concentrations. The department also offers concentrations in the interdisciplinary MS in Data Analytics Engineering and the PhD in Information Technology offered by the Volgenau School of Engineering.

Faculty in the department have research interests in networking, architecture, parallel and distributed computing, performance evaluation, software engineering, multimedia, graphics and visualization, databases, software engineering, data mining, secure information systems, artificial intelligence, computer vision, and robotics.

The Department is part of the Volgenau School of Engineering at George Mason University.
Welcome to the Department of Computer Science. The department offers BS, MS, and PhD programs in Computer Science; MS programs in Software Engineering, Information Systems, and Information Security and Assurance; and a BS program in Applied Computer Science with various concentrations. The department also offers concentrations in the interdisciplinary MS in Data Analytics Engineering and the PhD in Information Technology offered by the Volgenau School of Engineering.

Faculty in the department have research interests in networking, architecture, parallel and distributed computing, performance evaluation, software engineering, multimedia, graphics and visualization, databases, software engineering, data mining, secure information systems, artificial intelligence, computer vision, and robotics.

The Department is part of the Volgenau School of Engineering at George Mason University.
Welcome to the Department of Computer Science. The department offers BS, MS, and PhD programs in Computer Science; MS programs in Software Engineering, Information Systems, and Information Security and Assurance; and a BS program in Applied Computer Science with various concentrations. The department also offers concentrations in the interdisciplinary MS in Data Analytics Engineering and the PhD in Information Technology offered by the Volgenau School of Engineering.

Faculty in the department have research interests in networking, architecture, parallel and distributed computing, performance evaluation, software engineering, multimedia, graphics and visualization, databases, software engineering, data mining, secure information systems, artificial intelligence, computer vision, and robotics.

The Department is part of the Volgenau School of Engineering at George Mason University.
Sources Panel
View and edit CSS and JavaScript files (we’ll talk about that later!)
Network Panel
View all resources loaded in a timeline
Let’s spend 5 minutes playing with Chrome Developer Tools
IDEs
Integrated Development Environments (IDEs)

- **Integrates** everything you need to **develop** into a single **environment**

- E.g. text editor (with complex highlighting, auto complete, etc), version control access, debugger, etc.

- You don’t have to use one. It sure makes things handy though!

- We recommend you use WebStorm for this class. You can get a free student license: [https://www.jetbrains.com/student/](https://www.jetbrains.com/student/)
Node.js
Node.js

• Node.js is not a web server

• Node.js is a runtime for JavaScript based on Chrome’s JavaScript runtime

• Uses an event-driven, non-blocking I/O model

• Your code runs single threaded

• Has a large ecosystem of packages (npm)
Non-Blocking I/O

Regular (blocking) I/O

App → Read → Network

App waits for response

Non-Blocking I/O

App → Read → Network

App can do other things
Event-Driven Programming

Procedural

Main

Calls some methods

Does stuff

Done?

Event-Driven

Wait for Events

Call Event Handler

Event-driven programming relies on non-blocking I/O

Non-blocking I/O is easy with events… data being returned is just an event!
Packages

- Easily re-use libraries written by others

- npm manages libraries that your app uses: it downloads them, keeps them up to date, and ensures that they are in the right spot

- Example packages:
  - express (web server)
  - gulp (build system)
Node.js Getting Started

• Download and install it: https://nodejs.org/en/
  • We recommend v4.5.0 LTS (LTS -> Long Term Support, designed to be super stable)

• Demo: Hello world server

• Demo will show:
  • Using package manager to get a package (express)
  • Running a simple node application
Demo: Hello World Server

1: Make a directory, myapp

2: Enter that directory, type `npm init` (accept all defaults)

3: Type `npm install express --save`

4: Create text file `app.js`:

```javascript
var express = require('express');
var app = express();
var port = process.env.port || 3000;
app.get('/', function (req, res) {
    res.send('Hello World!');
});

app.listen(port, function () {
    console.log('Example app listening on port ' + port);
});
```

5: Type `node app.js`

6: Point your browser to `http://localhost:3000`

---

- Creates a configuration file for your project
- Tells NPM that you want to use express, and to save that in your project config
- Let's not worry about JavaScript syntax until next Thursday!
- Runs your app
Building & Grunt
Build Systems & JavaScript

• You’ve probably used a build system before
  • Make, ant, maven, gradle, etc.

• Apps written in JavaScript get a build system too!
  • Need to package together modules that we use, possibly also compile
    the JavaScript into some intermediate language before running it

• Why use a build system?
  • Builds should be repeatable
  • Builds should be reproducible
  • Builds should be standard
What else does a build system do for JS?

• Transcompile
• Transform modules
• Minification
• Preprocess stylesheets
• Cache Busting
• Image Optimization
• Run Tests
Grunt: A JS Build System

• One of many build systems for JS

• Configuration is written in JS

• Grunt is installed as an npm module (see http://gruntjs.com/getting-started )
Example Grunt Configuration

module.exports = function(grunt) {

    // Project configuration.
    grunt.initConfig({
        pkg: grunt.file.readJSON('package.json'),
        uglify: {
            options: {
                banner: '/*! <%= pkg.name %> <%= grunt.template.today("yyyy-mm-dd") %> */
            },
            build: {
                src: 'src/<%= pkg.name %>.js',
                dest: 'build/<%= pkg.name %>.min.js'
            }
        }
    });

    // Load the plugin that provides the "uglify" task.
    grunt.loadNpmTasks('grunt-contrib-uglify');

    // Default task(s).
    grunt.registerTask('default', ['uglify']);
};
Deploying & Heroku
Deployment

• Our little hello world example was cool, but how do we share with our friends?

• We need somewhere to run our node scripts

• Option 1: Use a machine that you have access to

• Option 2: Use a deployment service
Heroku

Hosts personal projects for free
Uses git as an interface

Create Project & Repo
Push to Github
Edit/commit
Edit/commit
Edit/commit

Pull/push
Push

Magic deployment!

Github

Heroku
Let’s adapt our example to run on Heroku

(See https://devcenter.heroku.com/articles/getting-started-with-nodejs for reference)
Heroku Example

1: Create account, install Heroku on your machine

2: In our app directory, create file “Procfile” with following contents:
   ```
   web: node app.js
   ```
   Tells Heroku what to do when it gets your app

3: Make the app a git repository: `git init; git add app.js Procfile package.json; git commit -m “Initial Commit”`
   Because Heroku uses git

4: Type `heroku create` and follow instructions

5: `git push heroku master`
   Deploys your code

6: Visit your app at the site listed in the result of the push (e.g. https://salty-depths-97600.herokuapp.com)
Reminder: HW0 due Tuesday!
Exit-Ticket Activity

Go to socrative.com and select “Student Login”
Class: SWE4320001 (Prof LaToza) or SWE4320002 (Prof Bell)
ID is your @gmu.edu email

For question 3:
Write one project idea that you have