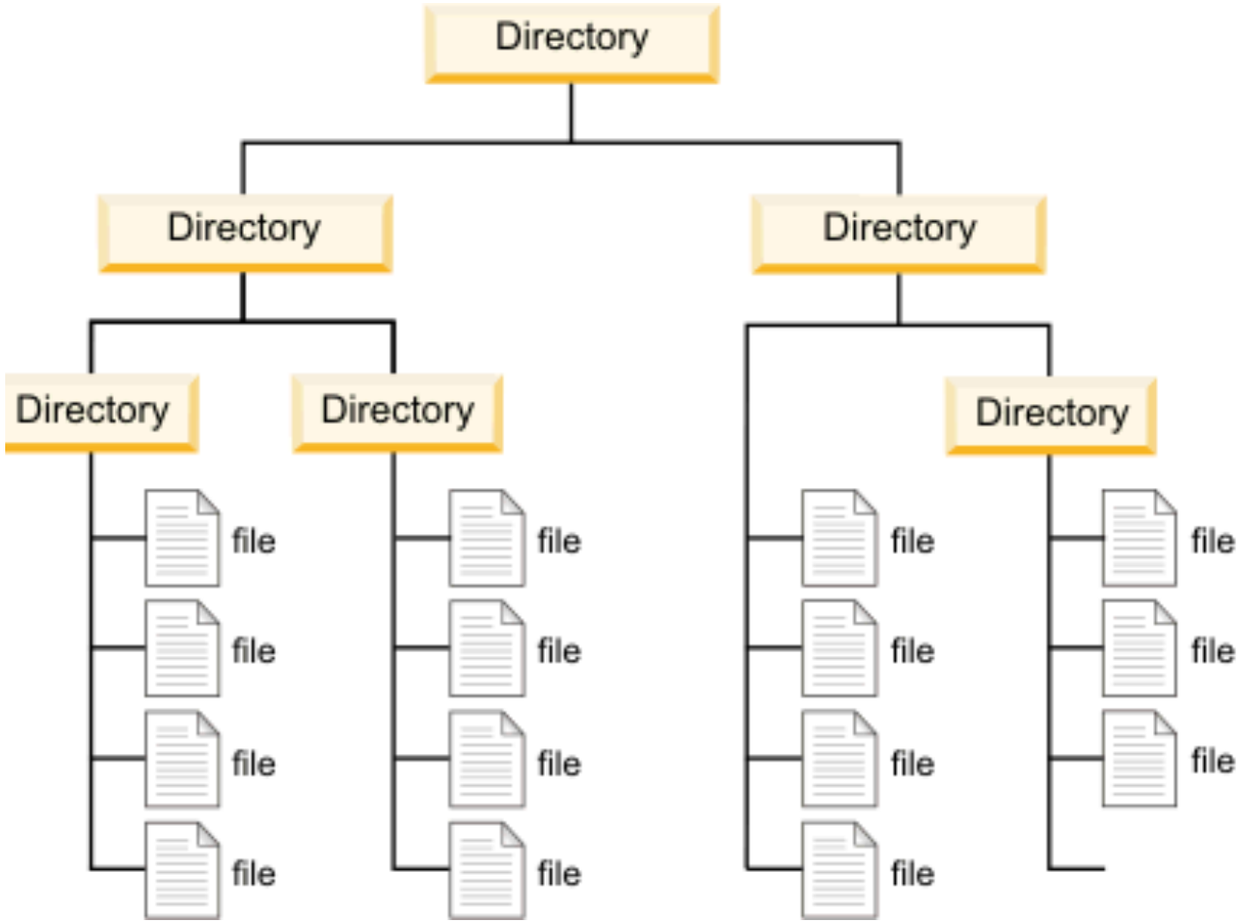
# Create your 111 Website




MICHAEL HENNESSY, UO/CIS

# Create your 111 Website

As MDN's Getting Started with the Web states, a website consists of many files: text content, code, stylesheets, media content, and so on.

When you're building a website, you need to organize these files on your local computer for fast, efficient retrieval.

You also need to make sure the files can connect to one another and get all your content looking exactly right, before you upload them to a server.

In this week's lab, you will set up the file structure for your 111 website on both your local computer and the uoregon.edu server.

# Week 1 Lab: Create your 111 Website


## BEFORE YOUR WEEK 1 LAB

1. Open the CIS 111 Schedule page in Chrome.
2. Complete steps 1-3 and 5-7 on the *Setup* menu. We will do step 4 in this lab.

## IN YOUR WEEK 1 LAB

At the start of the term your lab instructor will stop and ask the question, "Can all of you hear me, especially in the back of the room? Can you follow what I am saying?"

It's important that you are able to follow along as the instructor presents material. If you cannot hear or understand what is being said, please raise your hand to say that you are having difficulty following the presentation.

The lab instructor will appreciate your assistance.

Your part of the bargain is to avoid distractions (email, texts, sports, Facebook, etc.) and stay on task.

**Follow your lab instructor to complete these steps:**

1. Open Where should your website live on the server? in Chrome.

   Follow your lab instructor's directions to complete this step.

   You are now a Web Developer, so ***Memorize the Unix path to your home directory on the server:*** You already know your DuckID-- all you have to remember is the home number associated with your home directory.

2. Open Where should you website live on your computer? in Chrome.

## Take-Away points

Always maintain the identical folder structure on your computer and the web server, uoregon.edu.

For CIS 111, Information Architecture (IA) is defined as "how to organize folders and directories for efficient web page retrieval."

Best Practice: the 111 folder on your computer and the 111 directory on the server will have the same organization and structure.

Know the path to your home directory on the server.

# Hello, World in JavaScript

Your lab instructor will explain the four steps in the 111 WebDev Workflow (as defined in the glossary).

Then your lab instructor will get you started on this hands-on exercise to learn the **Four Steps of the 111 WebDev Workflow**.

## 1. EDIT WITH A CODE EDITOR

a. Download hello-world.html to the `Documents/111/p1/` folder on your computer. This is the example from Ch. 1 of *JavaScript: Absolute Beginner's Guide*, but without the html script element.

b. Open hello-world.html in a code editor (Atom or VS Code).

## 2. PREVIEW WITH CHROME

a. Preview hello-world.html in Chrome.

   a) Do this using your editor's Open in browser command.

   b) Do this using your editor's File Open .. command.

b. Add an h2 element with content, *Welcome to my First Web Page*.

c. Replace the *Hello World* text with this script element from Ch. 1:

```
<script>
   alert("Hello, World!");
   console.log("Hello, World!");
</script>
```

d. Preview the web page in Chrome.

   You should see a pop-up window displaying the Hello-World message. If you do not, ask your GE for assistance.

   Open the DevTools console to see the same message.

e. Add an h3 heading element following the script element, with the content, *Reasons I am Taking CIS 111*.

f. Add an ordered list following the h3 element. Add at least two list items, describing two reasons you are taking CIS 111 (adding more reasons is OK).

Preview the web page in Chrome. Make more edits, if necessary.

When the web page on your computer is perfect, and not before, you are done editing and can go to Step 3.

## 3. Upload with an SFTP Client

a. Start an SFTP client, select the SFTP protocol (not FTP), and connect to shell.uoregon.edu.

b. Upload hello-world.html to the server.

## 4. Test with Chrome

a. In Chrome, open the web page that is on the uoregon.edu server by opening this URL:

**http://pages.uoregon.edu/*yourDuckID*/111/**

Substitute your actual DuckID user name in the URL.

The hello-world.html web page hosted on the server should work exactly the same as the web page on your computer.

## Notes

- You will repeat these same four steps for each web page you create for 111. The fourth step is easy for beginners to forget. Do not forget this essential step!
- SFTP is used *only* for the Upload step: you drag and drop a file to transfer it from your computer to the server.
- Chrome is used for the Preview step and the Test step.

## Debugging JavaScript with Chrome DevTools

a. Modify the alert statement by spelling alert with a capital A, and then save the file:

```
Alert("Hello, World!");
```

b. Reload the web page:

```
Windows: control-shift-r
MacOS:   command-shift-r
```

Note the alert window does not appear, because the alert statement contains a syntax error.

c. Open Chrome's DevTools console:

```
MacOS:   option-command-j
Windows: control-shift-j (or the F12 key)
```

d. Note the error message, in red ink:

```
Uncaught ReferenceError: Alert is not
defined
```

Also note that Chrome provides the line number where the error occurred in the source file.

Correct the error, save the file, and reload the page. Check the console to see if the error message disappeared.

e. Repeat this entire exercise by introducing a new syntax error-- omit the closing quotes in the string:

```
alert("Hello, World!);
```

The error message you get this time will be much more cryptic.

The JavaScript engine was expecting a close-quote when it hit the right parenthesis.

Debugging is an essential skill that takes time and practice to master.

=> **ALWAYS REMEMBER**: When your JavaScript programs are not doing what you expect them to, open the DevTools Console and check for error messages.

# The 111 WebDev Workflow

This is the CIS 111 Web Development Workflow. Memorize these 4 steps.

1. **Edit**. Use a code editor (e.g., Atom or VS Code) to create a web page (.html file) on your computer.

2. **Preview**. Open the web page on your computer using Chrome. When the web page is perfect, and not before, you are done editing and can go to the next step.

3. **Upload**. Move all project files (.html, .css, image files, etc.) to the server using an SFTP client. This step is also known as *Deploying* the web page.

4. **Test**. Use Chrome to open your web page that is on the server. Do not use your SFTP app for this-- use Chrome.

   Type **http://pages.uoregon.edu/*yourDuckID*/111/** into Chrome to open the web page that is on the server.

   Make sure that the web page on the server is correct, because that is what will be graded.

**IMPORTANT**: Read Beware the Browser Cache (http://blogs.uoregon.edu/cis111schedule/beware-the-browser-cache/).

---

**Related Glossary Terms**

Drag related terms here

---

**Index** Find Term