*

CSCI-UA-4-005

Intro to Web Design + Computer Principles

Review Unix, Introduce HTML

Professor Emily Zhao
M/W 12:30PM – 1:45PM

*

Agenda

Review Assignment #1

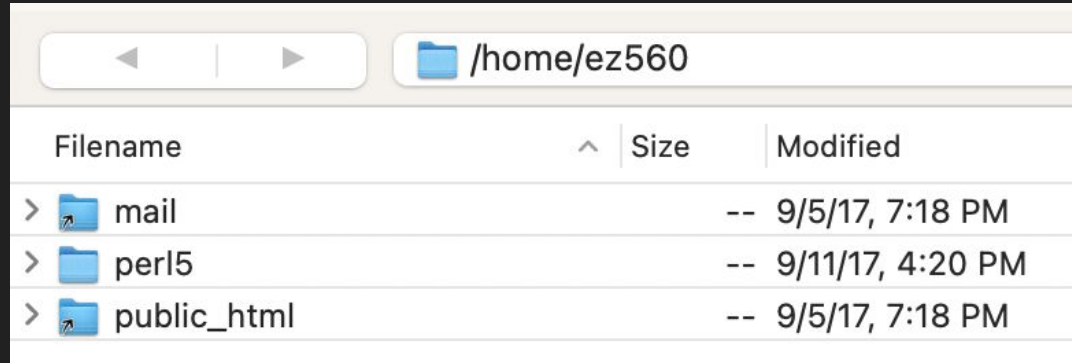— i6 accounts

— Accessing servers

— Setting permissions

Introduce HTML

— Code editors

— History of HTML

— HTML Basics

i6 Services

`http://i6.cims.nyu.edu/~NETID`

— `i6.cims.nyu.edu` is a server that provides a Linux environment for students to develop and host their websites
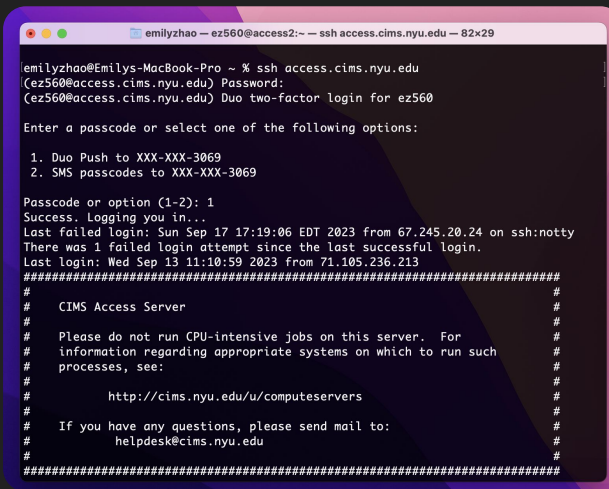— each account is granted a home directory and a web directory:

SSH (Secure Shell)

Allows users to securely log into remote systems and execute commands on those systems

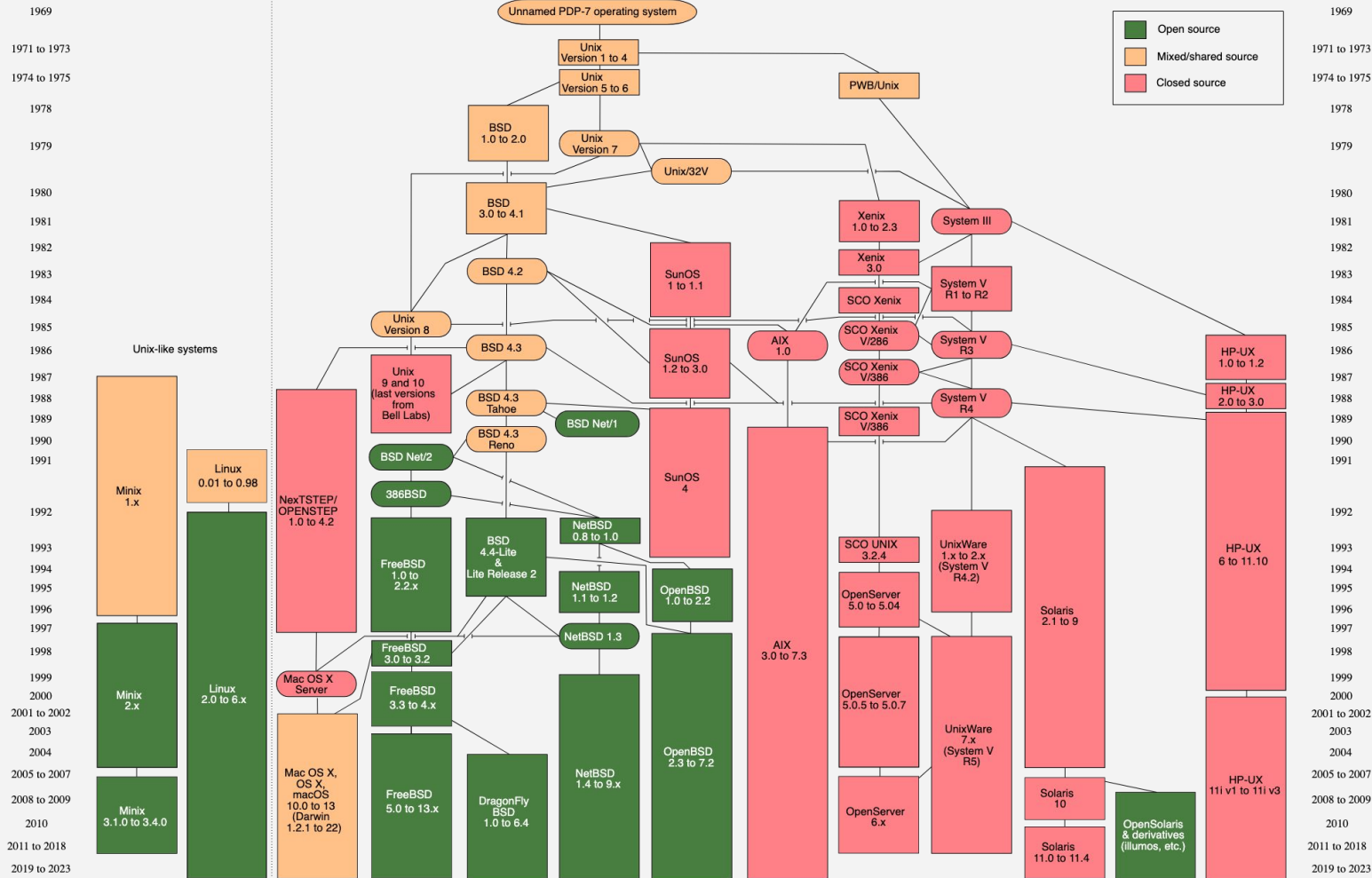— Login, change password, change file permissions


```
emilyzhao — ez560@access2:~ — ssh access.cims.nyu.edu — 82×29

emilyzhao@Emilys-MacBook-Pro ~ % ssh access.cims.nyu.edu
(ez560@access.cims.nyu.edu) Password:
(ez560@access.cims.nyu.edu) Duo two-factor login for ez560

Enter a passcode or select one of the following options:

 1. Duo Push to XXX-XXX-3069
 2. SMS passcodes to XXX-XXX-3069

Passcode or option (1-2): 1
Success. Logging you in...
Last failed login: Sun Sep 17 17:19:06 EDT 2023 from 67.245.20.24 on ssh:notty
There was 1 failed login attempt since the last successful login.
Last login: Wed Sep 13 11:10:59 2023 from 71.105.236.213
####################################################################
#                                                                  #
#      CIMS Access Server                                          #
#                                                                  #
#      Please do not run CPU-intensive jobs on this server.  For   #
#      information regarding appropriate systems on which to run such #
#      processes, see:                                             #
#                                                                  #
#              http://cims.nyu.edu/u/computeservers               #
#                                                                  #
#      If you have any questions, please send mail to:            #
#              helpdesk@cims.nyu.edu                               #
#                                                                  #
####################################################################
```

Common Unix Commands

```
% ls            list directory files

% pwd           show current directory

% cd            change directory
% cd ~          go to home directory
% cd ..         go to parent directory

% touch         create, change, modify timestamp of file

% mkdir         create directory
```
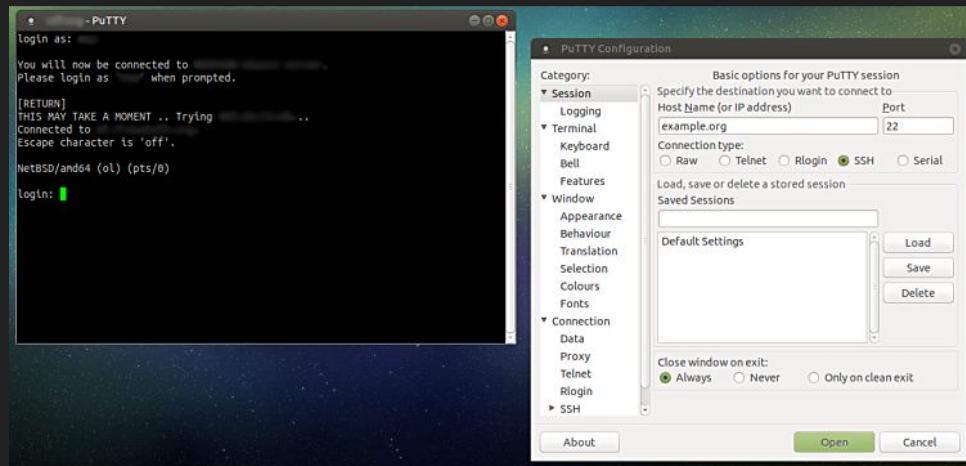
SFTP (SSH File Transfer Protocol)

A file transfer protocol that operates over an SSH connection

— Used solely for transferring files between client and server
— File management capabilities such as uploading, downloading, renaming, and deleting files
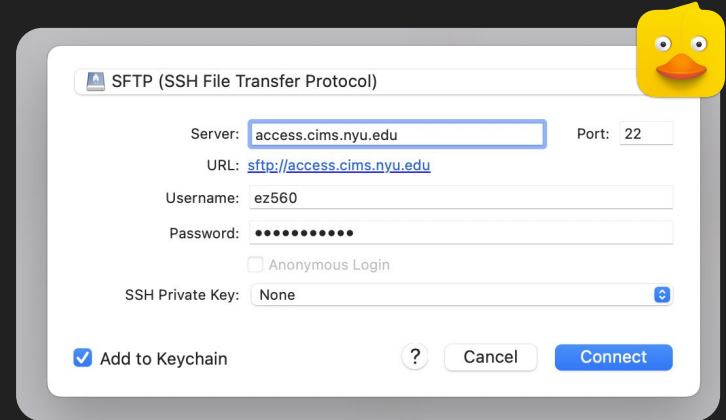
puTTY

an open-source terminal emulator and SSH client for Windows

chmod

Every file and directory has nine permissions associated with it

The Unix chmod command sets permissions of files and directories

Files and directories have three types of permissions (or none):

— r (read)
— w (write)
— x (execute)
— - (no permission)

The above permissions occur for each of the following classes or users:

— u (user/owner)
— g (group)
— o (other/world)

Standard Web Permissions

Permissions			Unix Commands
U	**G**	**W**	
rwx	rwx	rwx	% chmod 777 filename
rwx	rwx	r-x	% chmod 775 filename
rwx	r-x	r-x	% chmod 755 filename
rw-	rw-	r--	% chmod 664 filename
rw-	r--	r--	% chmod 644 filename

Standard **file** permission: 644

— owner can read and write file
— group can read file
— others can read file

Standard **directory** permission: 755

— owner can read, write + execute file
— group can read and execute file
— others can read and execute file

Assignment #1

1. Log into i6 account (`ssh`)
2. Change password (`ssh`)
3. Create HTML landing page (`text editor`)
4. Upload file to server (`SFTP`)
5. Change permissions of file/directory (`ssh`)

HTML

```html
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Your name goes here</title>
  </head>
  <body>
    <h1>Introduction to Web Design</h1>
    <p>Your self introduction goes here</p>
    <ul>
      <li>HTML</li>
      <li>CSS</li>
      <li>Raster Graphics</li>
      <li>Vector Graphics</li>
      <li>Page Layout</li>
      <li>Responsive Design</li>
      <li>JavaScript</li>
      <li>Final Project</li>
    </ul>
  </body>
</html>
```
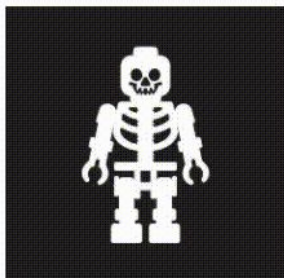
HTML
Add and orders elements on a webpage. Like the *skeleton* of a webpage.

CSS
Handles the styling of your website. For example, fonts, colors, sizing, etc.

JavaScript
Adds action and allows user interactions. For example, buttons and text input fields.

| HTML | CSS | JavaScript |
| structure | presentation/appearance | dynamism/action |

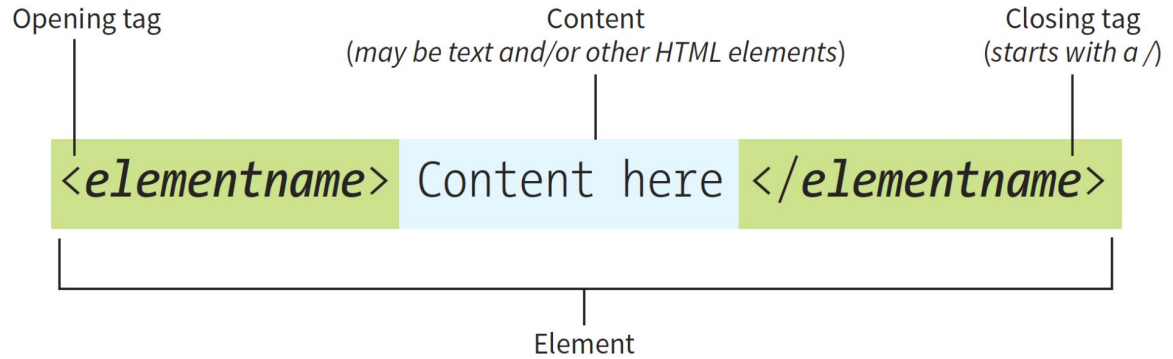|	**HTML**	**CSS**	**Javascript**
Language	**HTML**	**CSS**	**Javascript**
Purpose	Structure, Objects, Things	Looks, Style	Actions
Syntax	<p> <h1> 	P {color: red;}	var x = 5;
Grammar	nouns	adjectives	verbs
Building	Walls, structure	Paint, curtains	Electrical, Plumbing, AC

Hypertext Markup Language (HTML)

Hypertext: links that connect web pages to another

Markup: uses tags to define text structure and formatting

Language: what the computer system understands and
              uses to interpret commands

**HTML
Element**

Opening tag

Content
(*may be text and/or other HTML elements*)

Closing tag
(*starts with a /*)

`<elementname>` `Content here` `</elementname>`

Element

Example: `<h1>`Black Goose Bistro`</h1>`

FIGURE 4-6. The parts of an HTML container element.

```html
<h1> Why Coding Your Own Website Rocks </h1>
<p> There are many reasons why coding your
own website is better than using a website
builder </p>
<ul>
    <li> Low cost </li>
    <li> Rewarding </li>
    <li> Flexibility </li>
</ul>
<h2> Low Cost </h2>
<p> It's much cheaper than Squarespace!</p>
```

Why Coding Your Own Website Rocks

There are many reasons why coding your
own website is better than using a website builder.

- Low Cost
- Rewarding
- Flexibility

Low Cost

It's much cheaper than Squarespace!

Common HTML tags

`<h1>` to `<h6>`	Headings that define the hierarchy of section titles
`<p>` … `</p>`	Defines paragraphs of text
` `	A line break element to create new line
`<hr>`	"Horizontal rule", used to insert a visual break in content
`<ul>` … `</ul>`	Defines an unordered (bulleted list)
`<li>` … `</li>`	List element
`<blockquote>` … `</blockquote>`	Defines a block of text that is a quotation

```html
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Your name goes here</title>
  </head>
  <body>
    <h1>Introduction to Web Design</h1>
    <p>Your self introduction goes here</p>
    <ul>
      <li>HTML</li>
      <li>CSS</li>
      <li>Raster Graphics</li>
      <li>Vector Graphics</li>
      <li>Page Layout</li>
      <li>Responsive Design</li>
      <li>JavaScript</li>
      <li>Final Project</li>
    </ul>
  </body>
</html>
```

Common HTML tags

`<!DOCTYPE>`	Declares the document type
`<html> … </html>`	Container for all other HTML elements (except <!DOCTYPE>)
`<head> … </head>`	Contains metadata about the document, such as page title and links to stylesheets or scripts
`<title> … </title>`	Sets the title of web page, which is displayed in browser's title bar or tab
`<meta> … </meta>`	Contains metadata information about the document, including character encoding and viewport settings

HTML Boilerplate

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>HTML 5 Boilerplate</title>
  </head>
  <body>
  </body>
</html>
```

Can you nest HTML elements with each other?

Yes, you can!
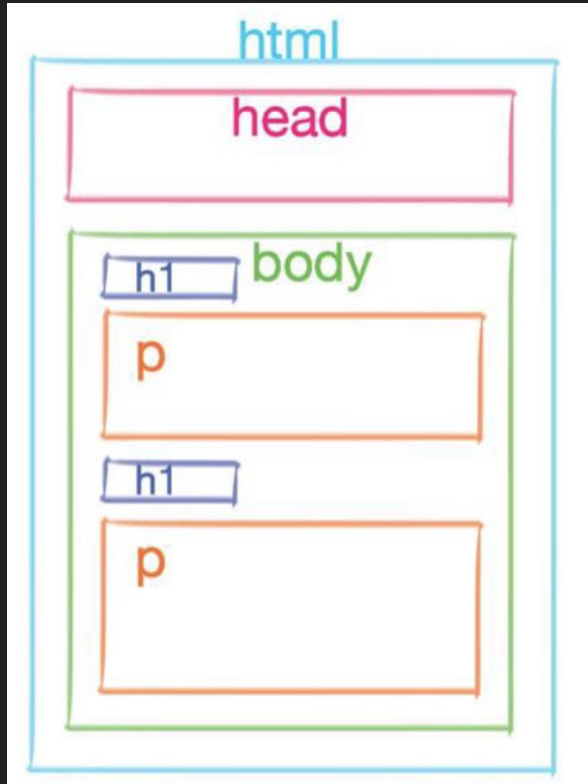

```html
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Your name goes here</title>
  </head>
  <body>
    <h1>Introduction to Web Design</h1>
    <p>Your self introduction goes here</p>
    <ul>
      <li>HTML</li>
      <li>CSS</li>
      <li>Raster Graphics</li>
      <li>Vector Graphics</li>
      <li>Page Layout</li>
      <li>Responsive Design</li>
      <li>JavaScript</li>
      <li>Final Project</li>
    </ul>
  </body>
</html>
```
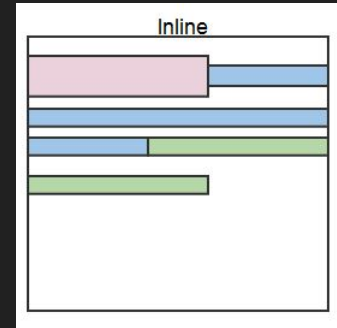
Block Element

— Starts on a new line
— Takes up the full width available (stretches out to the left and right as far as it can).
— Examples: `<h1>, <p>, <ul>, <blockquote>`
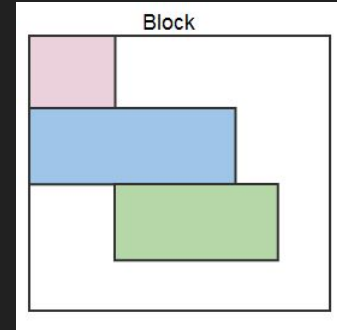
Inline element

— Doesn't create line breaks
— Occupies only necessary width
— Examples: `<strong>, <em>, <img>, <br>`

Rules for nesting

— Block elements can contain
  — ✅ Other block-level elements
  — ✅ Inline elements
— Inline elements can contain
  — ✅ Other inline elements
  — ❌ Block-level element
— ❌ Self-closing elements (i.e. `<br>`, `<img>`) can't contain other elements
— ✨ Special rules: `<ul>` should only contain `<li>` elements
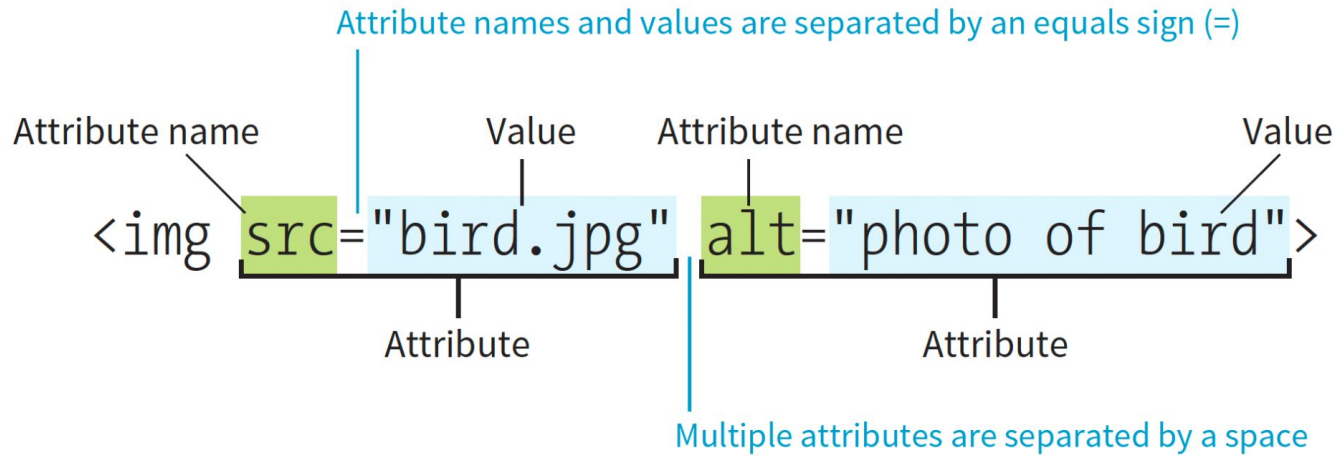
Basic HTML Live Demo

`<img>` tag

FIGURE 4-13. An **img** element with two attributes.

Absolute urls

— the full URL, including protocol (http/https), the optional subdomain (e.g.www), domain (example.com), and path.
— i.e. https://www.example.com/about/team/

Relative urls

— specifies the location of a resource relative to the current document's location
— does not include the protocol or domain name, focusing only on the path to the resource
— i.e. images/pic.jpg

Homework:

— Due tonight:

Assignment #1 (Unix)

— Due Wednesday:

Learning Web Design

Chapter 6: Adding Links

Chapter 7: Adding Images

— Due next Monday:

Assignment 2: (HTML) [posted after class]