

# Information Technologies

**Anselm Spoerri** PhD (MIT)

SC&I @ Rutgers University

[aspoerri@rutgers.edu](mailto:aspoerri@rutgers.edu)

[anselm.spoerri@gmail.com](mailto:anselm.spoerri@gmail.com)

## Lecture 2 – Overview

### **HTTP – HyperText Transfer Protocol**

**HTML** URL Basics | HTML Basics | Elements & Tags: Block vs. Inline  
Big Picture | HTML5 Basics | Key Rules | Links, Images, Tables  
HTML5 New Capabilities | New Elements | New Semantics

### **Tools**

- **Text Editor** to use to create web pages → **Install Visual Studio Code**
- **FTP & Permissions** FTP Program to Use → **Install Filezilla Client**

### **Demos**

- Your Server Account → **Activate rutgers-sci.domains account**
- Create Simple Web Page
- Create Site Definition in Filezilla Client, Upload to Server , Set Permissions

### **Exercise 1** Demos

### **Lectures – Week 2** Content

<http://comminfo.rutgers.edu/~aspoerri/Teaching/InfoTech/Lectures.html#week2>

HTTP – HyperText Transfer Protocol

**Protocol to Exchange Information over the Web**

## **Internet ≠ Web**

- **Internet = Collection of Global Networks**
- Web: way to manage information exchange

**There are many other uses for the Internet**

- File transfer (**FTP**)
- Email (SMTP, POP, IMAP)

## Web Basics – URL

# URL - Uniform Resource Locator

"http://www.abc.com/aaa/bbb/ccc.html"

"**http://**" - hypertext transfer protocol - **scheme**

"**www.abc.com/**" - **server name**  
- domain name, owner, host

"**/aaa/bbb/ccc.html**" - **path** through folder hierarchy

## URL Basics

### – Absolute URL

- "http://www.abc.com/aaa/bbb/ccc.html"
- **"Complete street address"**
- Info located on external server

### – Relative URL

- "../..../xxx/yyy.htm"
  - "../" = up 1 level => up 3 levels,  
then subdir "xxx" to get to "yyy.htm"
- **"Direction to neighbor's house"**
- Anchor (same page), Internal (local)

## Default "Home" Page = index.html

- Keeps out prying eyes out of directories (also instructor :)

# Web Standards

## URL

- **Where** to find it

## HTTP

- **How** to get it

## HTML

- How to **write** and **interpret** the information
- **Simple Document Structure Language** for Web
- Advantages
  - **Adapts** easily to different display capabilities
  - Widely available display software (browsers)
- Disadvantages
  - Does **not** directly **control layout**

# HTML is made up of **elements**

- Elements are denoted in HTML by using **tags**
- For the most part, you will **enclose content** you are marking up in **between tags**
- Tags look like this: **<tag>Content</tag>**  
(read as: open tag, content, close tag)
- **Three major elements needed** for an HTML page
  - **<html>** - container for **all of our HTML code**
  - **<head>** - **put data** for browser and other machines
  - **<body>** - **put content** to show to the user

## HTML – Example

```
<html>  
  <head>  
    Machine readable code (metadata) goes here  
  </head>  
  <body>  
    User readable content goes here  
  </body>  
  
</html>
```

### Some text elements

**<p>, <h1>, <h2>, <h3>, <ul>, <li>**

paragraph, heading 1, heading 2, heading 3, unordered list, list item

### Other elements

**<img>, <a>, <strong>, <em>**

image, anchor, strong, emphasis



## HTML – Block-level vs. Inline Elements

### **Block-level elements (`<p>`, `<h1>`, `<h2>`, etc.)**

- take up their own space **vertically**.
- force elements after them to jump to next line.

### **Inline-level elements (`<a>`, `<img>`, `<strong>`, `<em>`, etc.)**

- do not take up their own vertical space
- can be placed **inside** of other elements.

Note: **cannot place a block-level element inside of inline-level element**

document will not validate properly if you do

# Basic HTML

## Add Headings and Paragraphs

- `<h1> </h1>` header level one
- `<h2> </h2>` header level two
- `<p> </p>` paragraph

## Adding emphasis to text

- `<b></b>`, `<i></i>`, `<strong></strong>`

## Add links to other pages

Use various kinds of lists

Add images

# HTML – Big Picture

## Page Design → Progressive Enhancement

- Create **HTML structure** → Add CSS → Add Responsive Layout → Add JavaScript

## HTML5 → Markup with Meaning

- HTML 4, XHTML: use tags to **structure the content**
- **HTML5**: use tags to **structure** and **describe meaning** of content
  - **New semantic elements**: <header>, <nav>, <main>, <article>, <section>, <figure>, <aside>, <footer>
  - Redefines existing tags or eliminates others
  - **More forgiving** than XHTML: **/> optional** and <b> and <i> are back
  - Still true and necessary
    - HTML elements must be **properly nested**
    - HTML elements **should have closing tag**

## HTML – Examples

HTML elements must be **properly nested**

```
<p><em>Content</p></em>
```

→ 

```
<p><em>Content</em></p>
```

HTML elements **should close**

```
<p>Content
```

→ 

```
<p>Content</p>
```

```
<br>
```

 and 

```
<br />
```

 acceptable

HTML elements **best be lowercase**

```
<P>
```

→ 

```
<p>
```

# HTML – Examples

## Semantic Meaning

**<em>** *emphasis*

`<p><em>Content</em></p>`

**<strong>** **important**

`<p><strong>Content</strong></p>`

## Visual Appearance

`<p><i>Content</i></p>`

`<p><b>Content</b></p>`

# Basic HTML5 – **Essential Elements** to Include

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

- **<meta charset=utf-8" />**
- **<title>My Page</title>**

**<body>**

- **<header> <nav>**
- **<main> <article> <section> <figure>**
- **<aside>**
- **<footer>**
- **Headers:** <h1> <h2> <h3> <h4> <h5>
- **Paragraphs:** <p>
- **Tables:** <table> <tr> <td> </td> </tr> </table>
- **Lists:** <ol>, <ul> (can be nested)
- Adding **semantic emphasis** to text : <em>, <strong>. <i>, <b>

Font specification: **use CSS**

# Hyperlinks

**<a>** hyperlink tag

Use Attributes and Values

**<a href="absolute or relative pathname"**

**target="\_self"**

[opens page in same browser window]

**target="\_blank"**

[opens page in new browser window]

**>**

Text of Hyperlink

**</a>**

**Create Anchor: <a name="anchor name">**

**Link to Anchor: <a href="#anchor name">**

# Images

**<img>** image tag

Use Attributes and Values (**src** and **alt** are required to be valid HTML)

```

```



## **Two methods** for creating **Web Page Layout**

- **Tables (this lecture)**
- **DIVs** and **CSS** (next lecture)

**Table**     `<table> <tr> <td> </td> </tr> </table>`

- **<table>**            table
- **<tr>**            row
- **<th>**            header
- **<td>**            cell
- **<td colspan="2">**     **<td rowspan="2">**

# Table Example

**<table>**

<b>&lt;tr&gt;&lt;td&gt; 1.1 &lt;/td&gt;&lt;td&gt; 1.2 &lt;/td&gt;&lt;td&gt; 1.3 &lt;/td&gt;&lt;/tr&gt;</b>
<b>&lt;tr&gt;&lt;td&gt; 2.1 &lt;/td&gt;&lt;td&gt; 2.2 &lt;/td&gt;&lt;td&gt; 2.3 &lt;/td&gt;&lt;/tr&gt;</b>
<b>&lt;tr&gt;&lt;td&gt; 3.1 &lt;/td&gt;&lt;td&gt; 3.2 &lt;/td&gt;&lt;td&gt; 3.3 &lt;/td&gt;&lt;/tr&gt;</b>

**</table>**

# HTML5 – Intro

## Why HTML5?

- Reduce the need for external plugins (like Flash)
- Better error handling
- More markup to replace scripting
- HTML5 should be **device independent**
- Based on HTML, CSS, **DOM**, and **JavaScript**

# HTML5 – New Capabilities

<http://www.w3schools.com/html5/default.asp>

**Video** specifies **standard way** to embed video (no plug-in)

**Audio** specifies **standard way** to embed audio (no plug-in)

**Drag and Drop** any element can be **draggable**  
what to drag | where to drop | do the drop

**Canvas** used to **draw graphics**, on the fly, on web page

**SVG** supported **Scalable Vector Graphics** to draw shapes

**Geolocation** can determine **user's position** with permission

**Web Storage** better **local storage** within browser than cookies

**Web Workers** **JavaScript runs in background** without affecting page performance

**Server-Sent Events** page gets **automatic updates** from server

# HTML5 – DOCTYPE | charset | lang | CSS & JavaScript links

- **HTML5 DOCTYPE: `<!DOCTYPE html>`**
  - not case sensitive | version dropped
  - all browsers recognize shortened DOCTYPE & render in strict mode and deprecated elements will not work
- **Specify Character Set: `<meta charset="UTF-8" />`**
- **Specify Language: `<html lang="en">`**

`<!DOCTYPE html> <html lang="en"> <head><meta charset="utf-8" />`

- **Script and Link Declarations: type attribute optional**
  - `<link rel="stylesheet" href="styles.css" />`
  - `<script src="scripts.js"></script>`

# HTML5 – New Elements

## New Media Elements

<audio>	sound content
<video>	video or movie
<source>	multiple media resources for <video> and <audio>
<embed>	container for external application or interactive content (a plug-in)
<track>	text tracks for <video> and <audio>

## New Form Elements

- New form controls, like calendar, date, time, email, url, search

## New Semantic / Structural Elements

- **header, nav, aside, section, article** and **footer**
- Focus on your content and consider semantics of each element
- Use **div** if you need containing element strictly for style purposes
- Some older browsers treat new elements like inline elements

header, footer, nav, article, aside, section { display: block; }

# HTML5 – New Semantic / Structural Elements

- **header** element
  - Used to contain headline(s) for a page and/or section.
  - Can contain logos and navigational aids.
- **nav** element
  - Contains **major navigation** links.
  - Often contained by header.
- **aside** element
  - Contains related information, such as **sidebar** or pull-quotes.
- **section** element
  - Contains content that is **related** or grouped thematically.
  - Only if its content has own **self-contained outline** (h1,... h6).
  - Do not use simply for styling purposes – use divs and spans instead.
- **article** element
  - **Stand-alone** content such as a blog entry.
- **footer** element
  - Contains information **about** a page and/or section.

Text Editor for Creating Web Pages

Windows / Mac: **Visual Studio Code**

<https://code.visualstudio.com>

Note: **Atom** is used in demos, yet steps **same in Visual Studio Code**

**Download & Install** before proceeding :)

Please **View**

in Videos tab in Week 2 on Lectures page.



# Visual Studio Code Editor

## Snippets

- html ... Tab
- Loren ... Tab
- **Code completion**: press TAB or Enter will insert selected suggested code

## Workspace

- Folders from different locations on HD

**Status Bar** : View > Appearance > Status Bar

**Command Palette** : View > Command Palette or CTRL + SHIFT + P

**Enable AutoSave** : File > Auto Save toggle

**Word Wrap** : View > Word Wrap or ALT + Z

## Indentation

- Select code, Right-Click and Select "Format Document"
- Command Palette : Type "Open User Settings" and select.  
In Search settings box, input "indent" to search for settings related to indentation.  
Select "full" in Editor: Auto Indent section.

## Extensions :

- Extensions : **Atom One Light** | **Live Preview** ... Right Click File select "Live Preview"
- Extensions : **@enabled**

Active YOUR rutgers-sci.domains account

<https://rutgers-sci.domains/>

Click on "**Get Started**"

Log in with **YOUR NetID and Password**

You should receive "**New Account Information**" email

If not received, check SPAM folder

Email contains: **Your FTP username** and **Your FTP password**

(which are **both NOT the same** as your NetID and password)

Your Server URL = <http://YourNetID.rutgers-sci.domains/>

Example: NetID = acstst11

FTP username = acststru

Server URL = <http://acstst11.rutgers-sci.domains/>

## FTP – File Transfer Program and Key Ideas

### **Move Files between Machines**

- **Upload** (put) moves from client to server
- **Download** (get) moves files from server to client

Both **visual** and **command line** interfaces available

Want to send package to

**John Smith in USA**

**What is missing?**

**Want to Transfer File to Server – FTP**

Need to know

- **Server Address**
- **Username & Password**

# Directory and File Permissions

## Types of Permissions

- **Read** : for being able to read the file/directory
- **Write** : for being able to write in the file/directory
- **Execute** : for being able to access the file/directory

## Who are Permissions set for?

- **Owner** : you
- **Group** : group you belong to (e.g., LIS department, etc)
- **Others** : the rest of the world

## Directory and File Permissions (cont.)

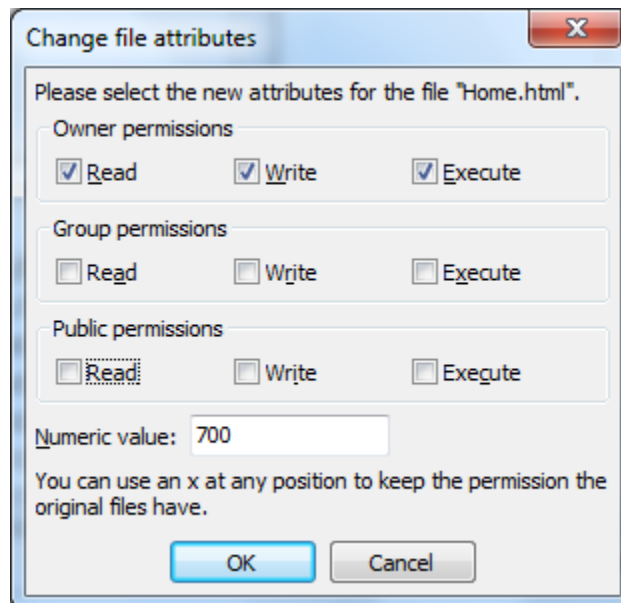
Want a **File** to **Save/Upload** on your server account but **others can not see/access it** then you should have the following

**Owner - Read(Yes) Write(Yes) Execute(Yes)**

**Group - Read(No) Write(No) Execute(No)**

**Others - Read(No) Write(No) Execute(No)**

**700**



## Directory and File Permissions (cont.)

Want people to **Access** and **View your files**

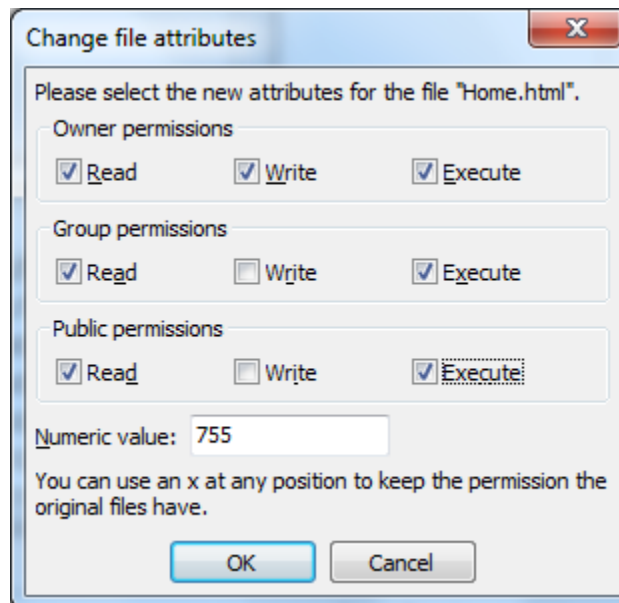
(such as your web page which will be stored in directory "public\_html") you should have the following set up

**Owner - Read(Yes) Write(Yes) Execute(Yes)**

**Group - Read(Yes) Write(No) Execute(Yes)**

**Others - Read(Yes) Write(No) Execute(Yes)**

**755**



# Difference between FTP and HTTP

## **FTP – File Transfer Protocol**

- Protocol used to **upload files** from a workstation to a FTP server or download files from a FTP server to a workstation.
- FTP is a **two-way system** as files are transferred back and forth between server and workstation.
- When *ftp* appears in a URL it means that the user is connecting to a file server and not a Web server and that some form of file transfer is going to take place.

## **HTTP –Hyper Text Transfer Protocol**

- Protocol used to **transfer files** from **Web server** to **Browser** to view a Web page. Unlike FTP, where entire files are transferred from one device to another and copied into memory, HTTP only transfers the contents of a web page into a browser for viewing.
- HTTP is a **one-way system** as files are transported only from the server onto the workstation's browser.
- When *http* appears in a URL it means that the user is connecting to a Web server and not a file server. The files are transferred but not downloaded, therefore not copied into the memory of the receiving device.

# FTP – File Transfer Protocol

Windows / Mac: **Filezilla**

<http://filezilla-project.org/>

## **To Do**

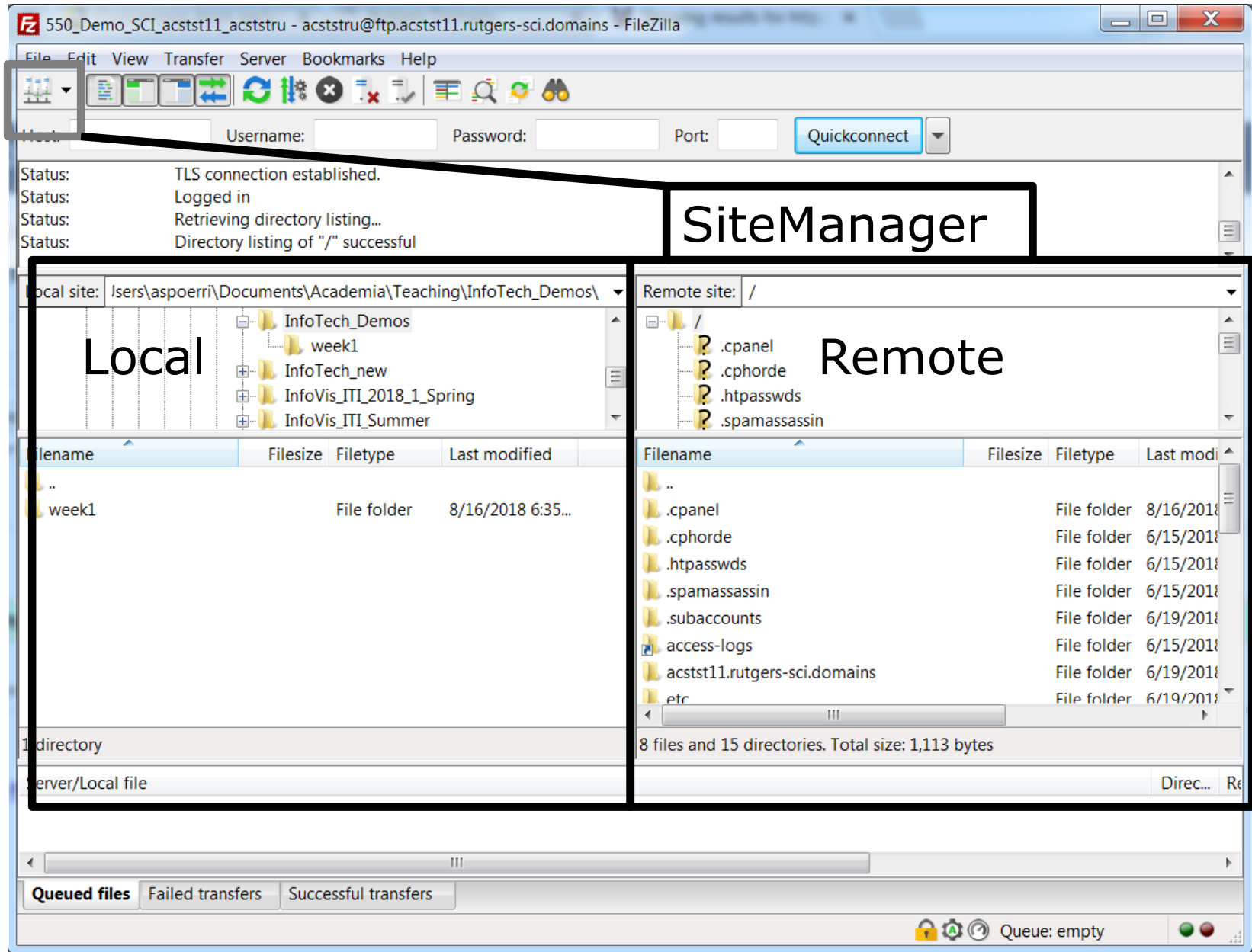
- **Install Filezilla FTP Client** on your computer

## **File Transfer Protocol : Demo**

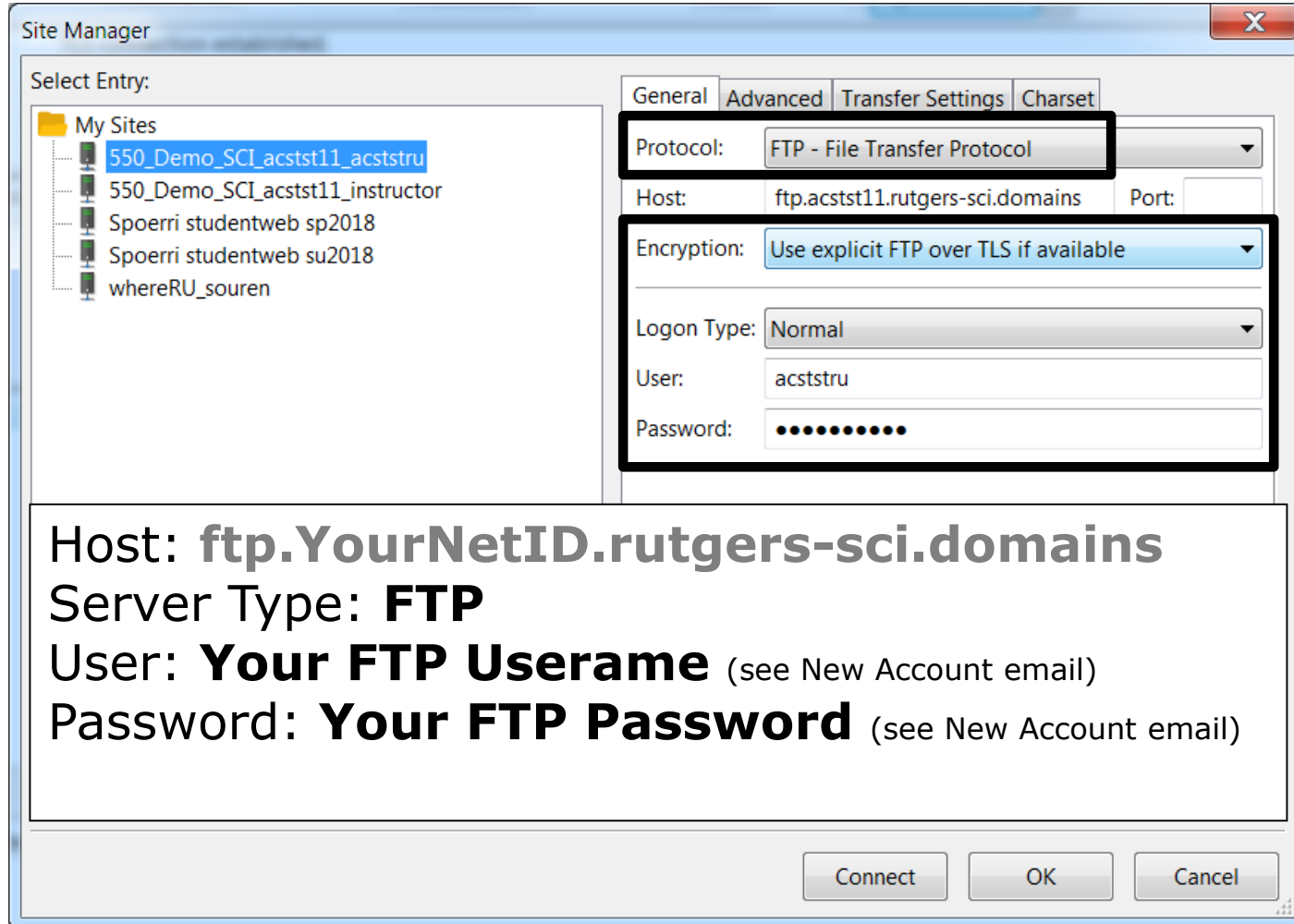
- Connect from local computer to server  
“ftp.**YourNetID**.rutgers-sci.domains”



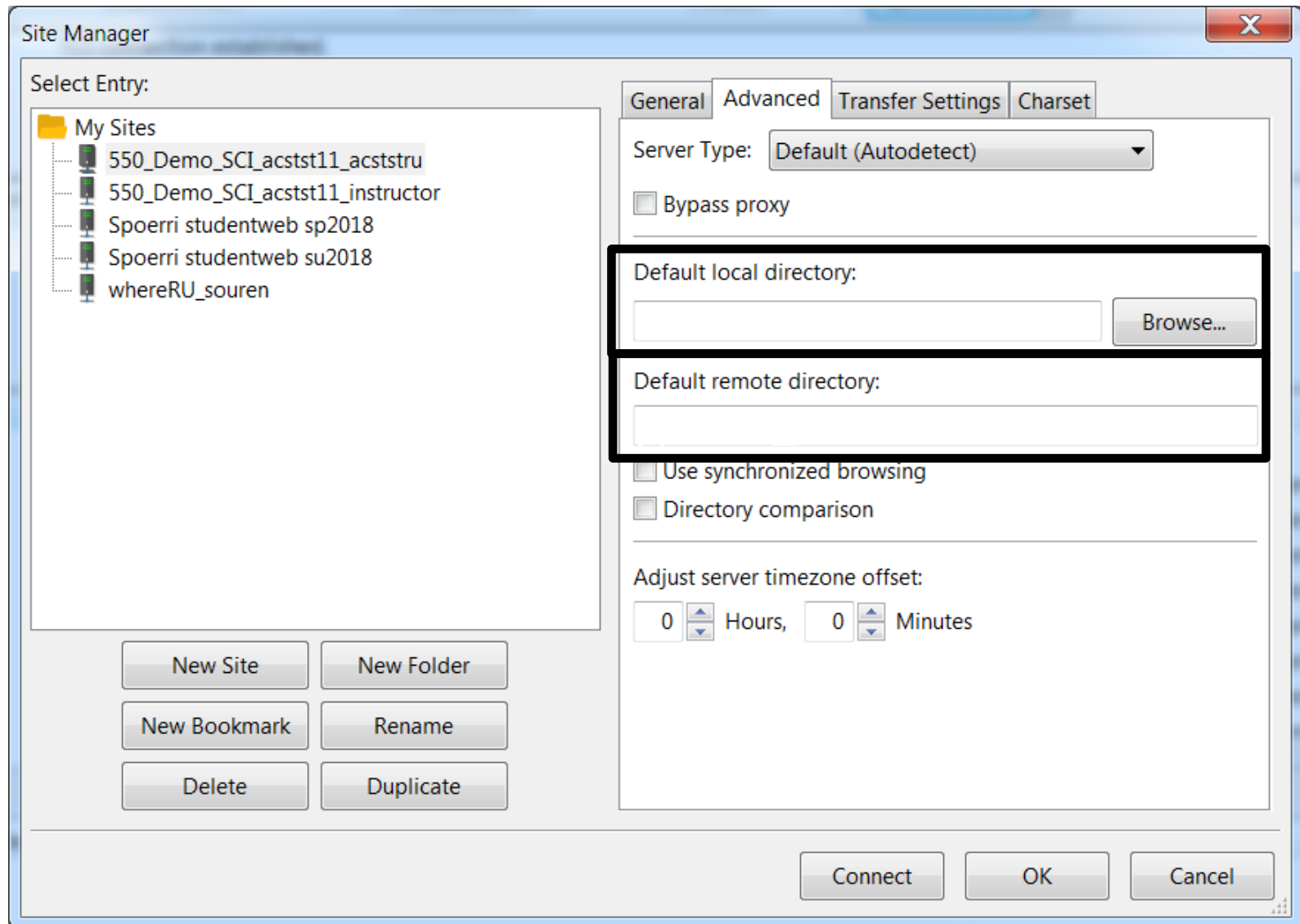
# FTP – File Transfer Protocol : Filezilla Main Window



# FTP – File Transfer Protocol : **Filezilla Site Manager**



# FTP – File Transfer Protocol : **Filezilla Site Manager**



## Demo – View Source

### **Use Web Browser to find page:**

<http://comminfo.rutgers.edu/~aspoerri/Teaching/InfoTech/Lectures/Lec2/Steps/notes.html>

### **Chrome: Right-Click** on Page and select **“View page source”**

- Opens window with source code

### **Compare HTML source with Web page**

- Observe how each effect is achieved

## Demo - Steps

### **Create Directory called "550"**

Go in that directory

### **Open New File with Visual Studio Code**

#### **Save as**

- **Save as type – All files** (otherwise saves it as .txt by default)
- File name - **test.html**

### **Practice HTML by editing test.html page**

## Tips

**Save Early, Save Often, just Save!**

**Reload Browser to see changes**

**File naming**

- **No spaces** in filenames
- Punctuation matters
- Use **lowercase** filenames

## **Upload** your work to **Your Server Account**

### **Content** needs to be in **public\_html directory**

- If you place “test.html” directly in public\_html directory you can see it

*http://**yourNetID**.rutgers-sci.domains/test.html*

- If you place an entire directory “550” in public\_html

*http://**yourNetID**.rutgers-sci.domains/550*

you will see the contents of directory

- If the 550 directory contains the file “test.html” and you want to see just that

*http://**yourNetID**.rutgers-sci.domains/550/test.html*

## URL & Server – Key Ideas

Want to see public web pages of specific user on Server

- Need to know
  - **yourNetID**
  - rutgers-sci.domains
- Where are public web pages physically stored on server?
  - **public\_html**
- What is the URL so that you can access it via Browser?
  - **http://yourNetID.rutgers-sci.domains/**
- **Why is public\_html is not part of URL?**



# Exercise 1

## Exercise 1: HTML – Create Information Resource

- Create at least **3 web pages**.
- Use a **table** to structure the page so that it has:
  - **3 columns**: 50 | 500 | 100 pixels wide
  - **4 rows**: 50 | 50 | 500 | 50 pixels high
- Each page has one **h1** tag and at least one **h2** tag  
**Sans-serif font** is used at least one word is **bolded** and another *italicized*
- Simple **navigation structure** using relative hyperlinks.
- Include at least one **absolute hyperlink** per page.
- Edit **images** found on flickr.com using free online tool.
  - Embed **200x200 cropped image** or **thumbnail**
- Create **screencast**, upload to YouTube and embed on a page
- Create **online survey** and insert link on a page

## Exercise 1 Demo – Step 1

Create folder = “**ex1**” inside your “550” folder

### Open New File with Atom

- **Save as type – All files** (otherwise saves it as .txt by default)
- File name - **ex1\_page1.html**
- **HTML5 doctype, encoding** Add **title** and essential elements (make sure to have closing tags)
- Font specification (e.g. “you are here” indicator) → Need to use **CSS**  
In <style> element inside <head> element:
  - `body { font: 100%/1.4 Verdana, Arial, Helvetica, sans-serif; color: #000099;}`
  - `td { width: 50px; height: 50px;}`
  - `.wide1 {width: 500px;}`
  - `.wide2 { width: 100px;}`
  - `.high1 { height: 500px;}`
  - `.here { color: #FF0000;}`

### Upload File to Server, Set Permissions, Test in Browser

## Exercise 1 Demo – Step 2

### **Create Table**

- **One row** with **three cells** and height = 50 and widths as indicated in Ex1
- Add following text "1.1", "1.2", "1.3" to respective cells

**Apply CSS classes** so that 2<sup>nd</sup> and 3<sup>rd</sup> column have width 500px and 100px, respectively

### **Save Page**

### **Upload File to Server and Test in Browser**

## Exercise 1 Demo – Step 3

### **Copy 1<sup>st</sup> Row code**

**Create 2<sup>nd</sup> Row** by pasting copied code and **customize**

- Add following text "2.1", "2.2", "2.3" to respective cells

**Create 3<sup>rd</sup> Row** by pasting copied code and **customize**

- Height = 500 (apply CSS class)
- Add following text "3.1", "3.2", "3.3" to respective cells

**Create 4<sup>th</sup> Row** by pasting copied code and **customize**

- Add following text "4.1", "4.2", "4.3" to respective cells

### **Save Page**

### **Upload File to Server and Test in Browser**

## Exercise 1 Demo – Step 4

Enter Following Content into 3<sup>rd</sup> Row and 2<sup>nd</sup> Column:

Heading 1 tag "Heading1"

Paragraph tag "Intro paragraph"

Heading 2 tag "Heading2"

Paragraph "Follow-up paragraph"

**Save Page**

**Upload File to Server and Test in Browser**

## Exercise 1 Demo – Step 5

**Find Image** and **Save to Hard Disk**

**Insert Image Detail or Thumbnail** before “follow-up” paragraph

**Specify Height, Width, Alternate Text, Left Alignment**

Make sure there is **visual gap between image and text**

Save Page

Upload File to Server and Test in Browser

## Exercise 1 Demo – Step 6

Enter Following Content into `<nav>` element:

“Page1 | Page2”

**Attach relative hyperlink to “Page1”** so that it links to page “ex1\_page1.html”

**Attach relative hyperlink to “Page2”** so that it links to page “ex1\_page2.html”

Save Page

Upload File to Server and Test in Browser

**Create “ex1\_page2.html” page** using **Save As**

Customize page

Upload File to Server, **Set Permissions** and Test in Browser

## Exercise 1 Demo – Step 7

**Create “you are here” indicator** in Navigation Structure

Select “PageN” on “ex1\_pageN” and remove hyperlink

Apply .here CSS class to “PageN” using inline <span> tag

Save Page

Upload File to Server and Test in Browser

You will need to **Validate Your Pages** using

<http://validator.w3.org/>

[instructor will let you know which errors you can ignore]