



# *Web Programming:* **HTML, CSS, JS, PHP**

Liangqin Ren

EECS 348: Software Engineering

Intro to HTML, CSS, JavaScript



1

# What is HTML



- HTML (Hyper Text Markup Language), is a language used to build Web pages
- Using HTML, you can build a Web page with text, graphics, sound, and video
- The essence of HTML programming is tags
  - A tag is a keyword enclosed by angle brackets
  - There are opening and closing tags
  - The affected text is between the two tags
  - Tag commands can be in lower or uppercase
  - Tags can be nested

# A Real example of HTML



```
<html>

<head>
    <h1>EECS 348 Lab Seven</h1>
</head>

<body>
Hello Everyone! This is the index of our Lab 7<br>

Please follow these links to visit other practices. Try to make it more beautiful 😊
<br>
    <a href="practice1.html">Practice1</a> <br>
    <a href="practice2.html">Practice2</a> <br>
    <a href="practice3.html">Practice3</a> <br>
    <a href="practice4.html">Practice4</a> <br>
</body>

</html>
```

← → ⌂ ⌄ 🔒 people.eecs.ku.edu/~l367r860/

## EECS 348 Lab Seven

Hello Everyone! This is the index of our Lab 7

Please follow these links to visit other practices. Try to make it more beautiful 😊

[Practice1](#)  
[Practice2](#)  
[Practice3](#)  
[Practice4](#)

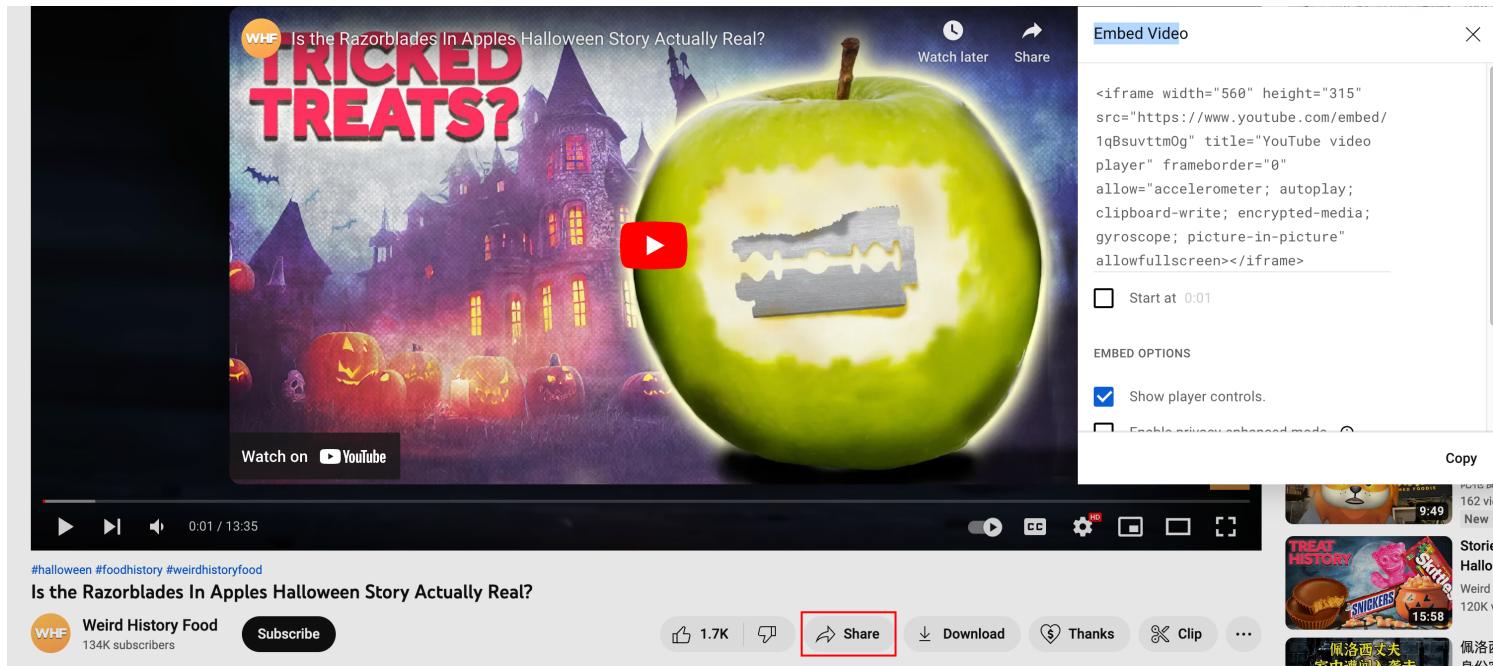
# Insert images



- Type `<IMG SRC = "image.png">`, where `image.png` indicates the location of the image file, it could be a local picture or a website picture URI
- The `WIDTH=n` and `HEIGHT=n` attributes can be used to adjust the size of an image
- The attribute `BORDER=n` can be used to add a border `n` pixels thick around the image

# Insert videos

- How to insert complex things like a video?
  - Some websites offer the codes for you, just use it!



- A link lets you move from one page to another, play movies and sound, send email, download files, and more....
- A link has three parts: a **destination**, a **label**, and a **target**
- To create a link type  
`<A HREF="page.html"> label </A>`
- The label is the text that will appear underlined or highlighted on the page

# What is JavaScript?



- A lightweight programming language ("scripting language")
  - used to make web pages interactive
  - insert dynamic text into HTML (ex: user name)
  - **react to events** (ex: page load user click)
  - get information about a user's computer (ex: browser type)
  - perform calculations on user's computer (ex: form validation)
- A web standard (but not supported identically by all browsers)
- **NOT** related to Java other than by name and some syntactic similarities

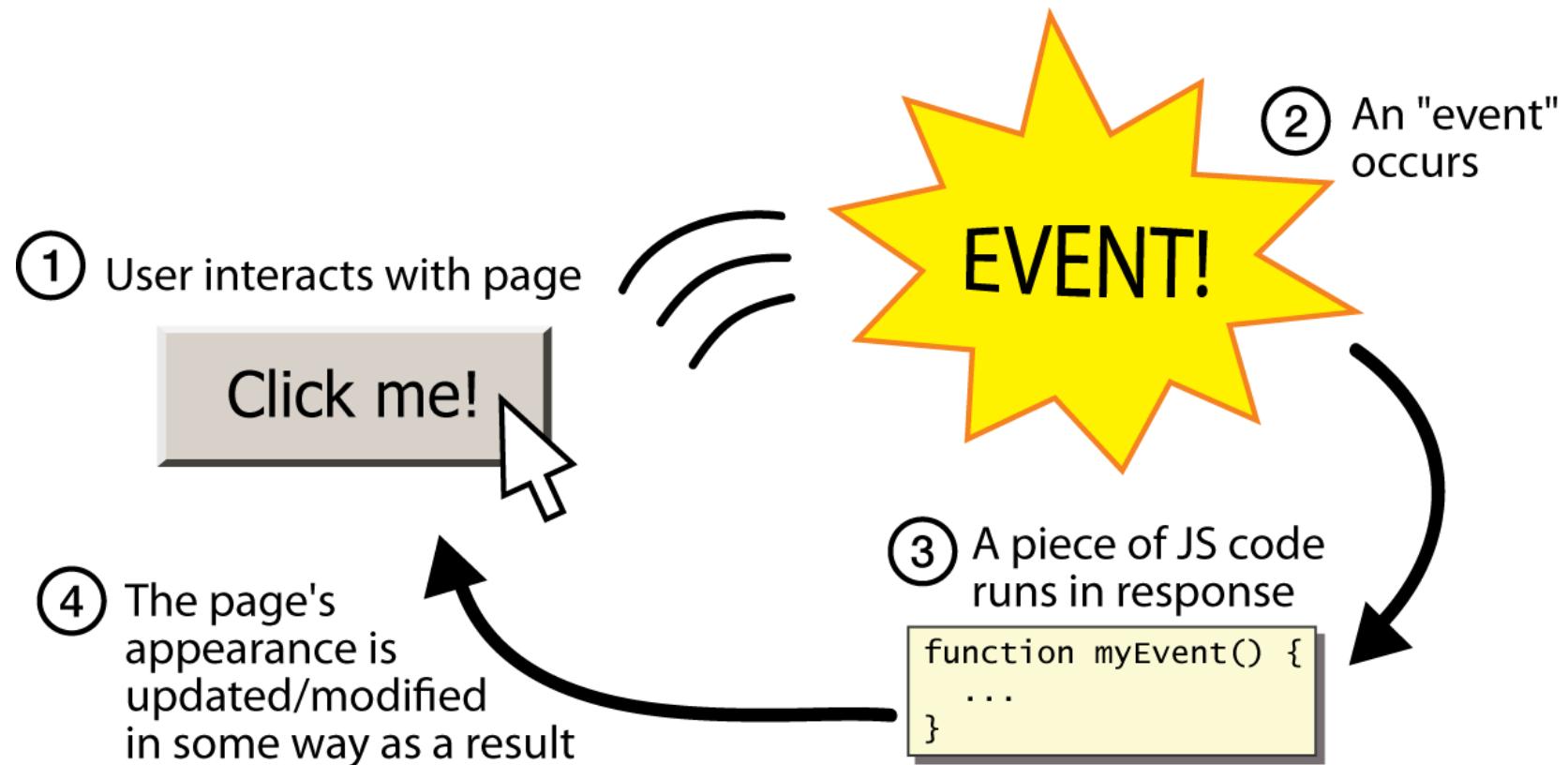
# Linking to a JavaScript file: script



```
<script src="filename" type="text/javascript"></script>
```

- Script tag should be placed in HTML page's head
- Script code is stored in a separate .js file
- JS code can be placed directly in the HTML file's body or head (like CSS)
  - but this is bad style (should separate content, presentation, and behavior)

# Event-driven programming



# Get user input



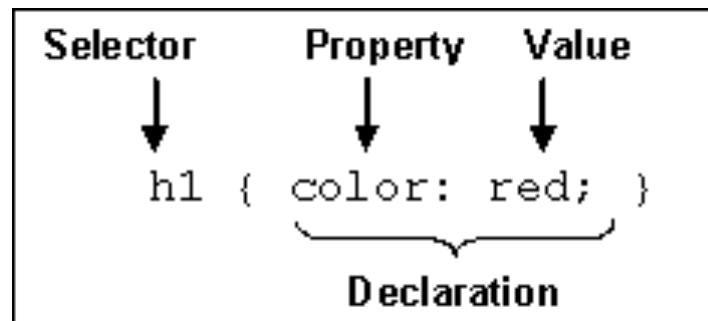
- HTML <input type="text" id="input" value="">
- Id is very important, JS use id to identify the variable to read or to write in HTML
- JS var input =  
document.getElementById("input").value;
- JS document.getElementById("input").value = input;

# What is CSS?

- Cascading Style Sheet
  - Stylesheet Language
    - \* Standards-based set of properties and attributes to define styles
  - To describe the presentation a document written in a ‘markup language’ like HTML or XML
    - \* Markup encoding: <p>My paragraph here.</p>
    - \* Defines the style of how things in <p> tags appear.
    - \* Font, color, size, margins, etc.
  - Cascading
    - \* Rules to determine how to apply markup that contains other markup



- 3 Elements to a CSS Statement
  - Selector
    - \* What HTML sections does it affect?
  - Property
    - \* What attribute of that HTML section will be affected?
  - Value
    - \* What change will be made to that attribute?



# Why CSS



- Separate Content from form
  - Specify the style once for every instance of that class.
    - \* Example: Specify the font once for all text on the HTML page that you've identified as a “header”
  - The stylesheet can be a separate file which all HTML pages on your entire site can link to.
    - \* Only have to specify the style once for your ENTIRE SITE
  - Can change the style for your entire site by editing only ONE FILE.

# Why CSS?



- Separate Content from Form
  - Content is the text and images, marked up to define regions of specific types
  - Form defines the “style” for the content

The old way:

```
<font size="14px">  
My First Header  
</font>  
<font size="12px" color="red" face="Verdana">  
My information 1 goes here.  
</font>  
<font size="14px">  
My Second Header  
</font>  
<font size="12px" color="red" face="Verdana">  
Different information goes here.  
</font>
```

# Why CSS?



- Separate content from form
  - Content

```
<div id="paragraph">  
    Paragraph to be altered.  
</div>
```

- Form or Style

```
<style>  
    #paragraph {  
        border-style: solid;  
        border-width: 10;  
        background-color: burlywood;  
        color: black;  
        font-size: 18pt;  
        background-image: url("");  
    }  
</style>
```

# Three CSS definition locations



- Inline: the “style” attribute

```
<p style="font-color:red;font-size:10px;">Content</p>
```

- Note, the selector for inline CSS is the tag that contains the style attribute

- Internal: the <style> markup tag

```
<html><head><style>
p { background-color: Red;
    font-family: serif;
    font-color: White; }
</style></head><body>
<p>Content</p>
</body></html>
```

- External: the .css stylesheet file

```
<link rel="stylesheet" type="text/css" href="mystylesheet.css" />
```

# CSS syntax: selectors



- There are many kinds of selectors and many ways to reference them:
  - Type, Class, ID, Pseudo, etc.
- HTML Type Tag – selected with the tag type

```
p { font-size: 10px;  
    font-color: White; }
```

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

- The Class Attribute – precede the class with a period

```
.myinfo { font-size: 10px;  
    font-color: White; }
```

```
<p class="myinfo">Content</p>  
<div class="myinfo">Other content</div>
```

- A widely-used open-source scripting language
  - Free to download ([php.net](http://php.net))
- Stands for Hypertext Preprocessor
- PHP scripts are executed on the server side
  - Not on a local machine (unless PHP is installed, and you are running a local webserver)



- Have a default .php file extension
- May contain text, HTML, JavaScript, and PHP code
- PHP code is executed on the server, and the result is returned to the browser as plain HTML

# Why PHP?



- PHP runs on different platforms (Windows, Linux, Unix, Mac OS X, etc.)
- PHP is compatible with almost all servers (Apache, IIS, etc.)
- PHP has support for a wide range of databases
- PHP is relatively easy to learn and runs efficiently on the server side
  - Lots of built-in functionality; familiar syntax
- PHP is well-documented:
  - Type `php.net/functionName` in browser address bar to get docs for any function

# Basic syntax



- A PHP script starts with <?php and ends with ?>

```
<?php  
// PHP code goes here  
?>
```

- The default file extension for PHP files is ".php".
- A PHP file normally contains HTML tags, and some PHP scripting code
- Each code line in PHP must end with a semicolon
  - The semicolon is a separator and is used to distinguish one set of instructions from another
- Two statements to output: echo and print

- Variable names start with \$ followed by the name
- A variable name must begin with a letter or the underscore character
- A variable name can only contain alphanumeric characters and underscores (A-z, 0-9, and \_ ) and no spaces
- Variable names are case sensitive (\$y and \$Y are two different variables)

# For loop (similar to Java)



```
for (initialization; condition; update) {  
    statements;  
}
```

## Example

```
for ($i = 0; $i < 10; $i++) {  
    print "$i squared is " . $i * $i . ".\n";  
}
```

# In the future

- Use the existing framework rather than writing HTML files by yourself.
- Markdown + GitHub + Hugo/Hexo/Jekyll + (Netlify)

