CSS – Ch 5

Normal flow: how browser displays block levewl and inline elements from left to right top to bottom.

Block-level elements: <p>,<div>,etc Inline: <em>,<a>,<img>,<span>

Replaced inline: context/appearance defined by external source such as <img> and form elements.

Nonreplaced inline: content defined within document.

display:inline; display:block;

POSITION: absolute – element removed from normal flow and pos is relative to nearest positioned ancestor.

fixed – element in fixed pos in window even when document scrolls

relative – elements move relative to where it would be in normal flow

static – element pos according to normal flow. this is default

A moved element via absolute is positioned relative to its nearest positioned ancestor container

z-index:#; allows moving items in front or behind other elements

FLOAT: floated elements move left/right within its container

left – left-hand edge of element cannot be adjacent to another element

right – same but for right

both – both

none – element can be adjacent to other elements

Stop elements from flowing around floated element using clear:left/right/both;

overflow:auto; fixes issues with float containers with float objects

display:auto; default. display:none; essentially removes element from page. visibility:hidden; hides item but the space from that element exists still.

use :hover for hovering over item

Fixed Layout: Based around ideal width based on typical monitor resolution.

Liquid Layout: Widths set with percentages. disadvantage: can be difficult to create and line length may become too long or too short.

Hybrid Layout: Combines pixel and percent measurement.

Responsive design: page responds to changes in browser size. Four components to make this work: Liquid layout, scaling images to viewport size, setting viewports using <meta> tag, customizing css for different viewports.

<meta name=”viewport” content=”width=device-width”/>

CSS Media Query: apply style rules based on medium that is displaying the file.

@media only screen and (max-width:480px){css rules go here and are only used if device matches conditions}

media:defines as media query, only: only use if both conditions are true, screen:device has to be screen

Progressive Enhancement: Designing css rules first for phones, then tablets, then desktops, etc. Design adapted to progressively more advanced devices.

MEDIA QUERIES: width/height – width/height of viewport

device-width/height – width/height of device

orientation – portrait or landscape

color – number of bits per color

CSS Preprocessor – tools that sue variables, inheritence, etc.

JAVASCRIPT – Ch 6

Javascript is dynamically typed, object oriented, and runes inside browser.

Objects are prototype-based instead of class based

Advantages of client-side scripting:

Processing offloaded from server

Browser responds more rapidly to user events

JS can interract with html in ways the server cannot.

Disadvantages:

Client may not have JS enabled

Differences between different browsers and OSs cause problems

JS heavy web apps can be hard to debug/maintain

JS introduced by netscape in navigator browser in 1996

JS is implementation of ECMAScript

JS is more importent with development of AJAX (Asynchronnous Javascript And XML)

Layers: Presentation (focused on UI), Business (model entities such as customers/sales), Data (handle data source interaction)

Alternate Layers:

Javascript(Framework, e.g. jQuery)

Javascript (Asynchronous)

Javascript (Validation)

Javascript (Presentation)

CSS

HTML

Users without Javascript:

Web Crawler: Client running for search engine to download your site so it can be featured in their search results.

Browser Plug-in: Software that works with browser that may interfere with JS.

Text-based browsers

Visually Impaired Client: These people use different browsing software to read contents of page out loud.

<noscript> tag only displays items between it for users without ability to load JS. Often used to prompt users to enable JS.

Graceful Degradation:

Develop site for abilities of current browsers. Users with old browsers get alternate site pages that lack JS used on main site.

Progressive Enhancement:

Create site using CSS, JS, and HTML features that are supported by all browsers of a certain age or newer.

Inline JS allowed but sucks <a href=”JavaScript:OpenWindow();” etc>

Embedded JS is just a JS block wrapped in <script></script>

External JS file called with <script type=”text/JavaScript” src=”filename.js”></script>

JS is type sensitive. Variables declared in loop are accessable outside of loop so scope of vars in blocks is not supported.

‘===’ operator tests for equality and type equivalence.

null and undefined two different states for vars

semicolons arent required but are allowed and encouraged

declare variables with var name = blahblah;

‘!==’ not equal in value or type

functions dont need a return type set since everything is dynamic

call function with functionName(var1,var2,etc);

alert(); makes browser popup with code being passed in.

console.log(); used to output to console

try{nonexistantfunction(“hello”);}catch(err){alert(“exception” + err);} for error handling

throw “new error”; can be used to throw custom errors within try/catch block

Create new object: var someObject = new ObjectName(p1,p2,...,pn);

To use properties: alert(someObejct.property);

To use methods: someObject.doSomething();

Arrays:

var arr = new Array();, var arr = new Array(“one”, “two”);, var arr = [“one”, “two”];

arr.push(“new item”);

arr.pop(); removes item from end of array.

arr.concat(), slice(), join(), reverse(), shift(), sort() exist as well

Math class exists as well: Math.PI, Math.sqrt(4), etc...

String concatenation: var str = str1 + “more words”;

var date = new Date();

DOM – Document Object Model

Platform and language-neutral interface allowing programs and scripts to dynamically access and update content, structure, and style of documents.

Each element in HTML cdocument is called a node.

Node Properties: attributes, childNodes, fitstChild, lastChild, nextSibling, nodeName, nodeType, nodeValue, parentNode, previousSibling

var a = document.doctype.name;

var b = document.inputEncoding;

DOM Methods: createAttribute(), createElement(), createTextNode(), getElementById(id), getElementsByTagName(name)

getElementsByName returns a nodeList of elements.

Element Node Properties: className, id, innerHTML, style, tagName

document.write() method is used to create output to html page from JS. using innerHTML property allows us to write at a specific spot in html page

var latest = document.getElementById(“latestComment”);

var oldMsg = latest.innerHTML;

latest.innerHTML = oldMsg + “<p>Update div with JS</p>”;

Better way of implementing this:

var latest = document.getElementById(“latestComment”);

vat oldMessage = latest.innerHTML;

var newMessage = oldMesage + “<p>new</p>”;

latest.removeChild(latest.firstChild);

latest.appendChild(document.createNextNode(newMessage));

Updating element style

var commentTag = document.getElementById(“specificTag”);

commentTag.style.backgroundColor = “#FFFF00”;

Dom properties by tag type

a href

a, input, textarea, form name

img, input, iframe,script src

input, textarea, submit value

Event Listeners

var greeting = document.getElementById(“element”);

greeting.addEventListener(‘click’, alert(“Clicked”));

greeting.addEventListener(‘mouseOut’, alert(“gone”));

Can call functions from event listeners

element.onClick = function() { function crap goes here };

function someHandler(e){} function where DOM event object is passed in

Event Objects:

Bubbles – If an even’t bubbles property is set to true, then there must be an event handler in place to handle the event or it will bubble up to itrs parent and trigger and event handler there.

Cancelable – Cancelable property is also a bool value that indicates whether or not the event can be cancelled.

preventDefault – Cancelable default action for an event can be stopped using preventDefault() method

function submitButtonClicked(e){ if(e.cancelable){e.preventDefault();}}

Types of Events: mouse, keyboard, form, frame

Mouse Events: onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmouseout, onmousemove

Keyboard Events: onkeydown, onkeypress, onkeyup

Form Events: onblur (element lost focus), onchange (some field vlaue changed), onfocus, onreset, onselect, onsubmit

Frame Events: onabort, onerror, onload, onresize, onscroll, onunload

JQUERY/AJAX – Ch 15

Object Literal – List of key value pairs with colons betwene key and value with comma separating key-value pairs

Known as Plain Objects in jQuery

var oneDie = {color : “FF0000”, faces : [1,2,3,4,5,6]};

oneDie.color=”0000FF”; changes color

Instantiate Class in jQuery: var obj = new ClassObject(pass in variables here);

Can nest functions in functions to emulate class with methods

Prototype:

function Die(col){this.color=col; this.faces=[1,,2,3,4,5,6];}

Die.prototype.randomRoll = function(){var randNum = Math.Floor((Math.random() \* this.faces.length) + 1);

return faces[randNum-1];};

Library/Framework is software you can use in your software.

CDN – Content Delivery Network

Including jQuery: <script src=<http://code.jquery.com/jquery-1.9.1.min.js>></script>

ERRORS – Ch 12

Types: Expected (bad input, db connect, etc.), Warnings (warnings but not halt execution), Fatal (page terminates)

Check input for number

$id = $\_GET[‘id’];

if(!empty($id) && is\_numeric($id)){}

Error is a type of problem that generates nonfatal warning.

Exception refers to objects that are of type Exception and which are used in conjunction with try/catch for runtime errors.

PHP Error Reporting Flags: error\_reporting, display\_errors, log\_errors

Set error reporting type: error\_reporting(E\_ALL/E\_ERROR/E\_WARNING/(blank));

display\_error setting specifies if error messages are displayed in browser: ini\_set(‘display\_errors’,’0’);

same for log\_errors

Regular Expression: set of special characters that define a pattern. Two types of characters

Literal: Character to match in target

MetaCharacter: special symbol that acts as command to regex parser

. [ ] \ ( ) ^ $ | \* ? { } +

To escape any of these, preface it with \

|  |  |
| --- | --- |
| **Expression** | **Description** |
| **^ … $** | If used at the very start and end of the regular expression, it means that the entire string (and not just a substring) must match the rest of the regular expression contained between the ^ and the $ symbols. |
| **\t** | Matches a tab character. |
| **\n** | Matches a new line character. |
| **.** | Matches any character other than \n. |
| **[qwerty]** | Matches any single character of the set contained within the brackets. |
| **[^qwerty]** | Matches any single character not contained within the brackets. |
| **[a-z]** | Matches any single character within range of characters. |
| **\w** | Matches any word character. Equivalent to [a-zA-Z0-9]. |
| **\W** | Matches any nonword character. |
| **\s** | Matches any white-space character. |
| **\S** | Matches any nonwhite-space character. |
| **\d** | Matches any digit. |
| **\D** | Matches any nondigit. |
| **\*** | Indicates zero or more matches. |
| **+** | Indicates one or more matches. |
| **?** | Indicates zero or one match. |
| **{n}** | Indicates exactly n matches. |
| **{n,}** | Indicates n or more matches. |
| **{n,m}** | Indicates at least n but no more than m matches. |
| **|** | Matches any one of the terms separated by the | character. Equivalent to Boolean OR. |
| **()** | Groups a subexpression. Grouping can make a regular expression easier to understand. |

regex to mathc phone number without area code ^\d{3}-\d{4}$

The ^ and$ indicate beginning and end of string.

\d indicates digit and {#} indicates number of digits

Phone number not allowing first digit to be 0 or 1: ^[2-9]\d{2}-\d{4}$

Phone number allowing single space, period, or dash between numbers: ^[2-9]\d{2}[-\s\.]\d{4}$

Allow multiple spaces but only a single dash or period: ^[2-9]\d{2}[-\s\.]\s\*\d{4}$

Allow area code to be surrounded by brackets, separated by spaces, a dash, or a period

^\(?\s\*\d{3}\s\*[\)-\.]?\s\*[2-9]\d{2}\s\*[-\.]\s\*\d{4}$

Zero or one ‘(‘ = \(?

To make the area code optional, surround the first block with ( )?

|  |  |
| --- | --- |
| **Regular Expression** | **Description** |
| **^\S{0,8}$** | Matches 0 to 8 nonspace characters. |
| **^[a-zA-Z]\w{8,16}$** | Simple password expression. The password must be at least 8 characters but no more than 16 characters long. |
| **^\d{5}(-\d{4})?$** | American zip code. |
| **^((0[1-9])|(1[0-2]))\/(\d{4})$** | Month and years in format mm/yyyy. |
| **^(.+)@([^\.].\*)\.([a-z]{2,})$** | Email validation based on current standard naming rules. |
| **^((http|https)://)?([\w-] +\.)+[\w]+(/[\w- ./?]\*)?$** | URL validation. After either http:// or https://, it matches word characters or hyphens, followed by a period followed by either a forward slash, word characters, or a period. |
| **^4\d{3}[\s\-]d{4}[\s\-] d{4}[\s\-]d{4}$** | Visa credit card number |
| **^5[1-5]\d{2}[\s\-]d{4}[\s\-] d{4}[\s\-]d{4}$** | Mastercard credit card number |

Types of Input Validation: Required Info, Correct Data Type, Correct Format, Comparison, Range Check, Custom

CAPTCHA – Completely Automated Public Turing test to tell Computers and Humans Apart

Always perform server side validation as client side is not to be trusted

Validate on server side using PHP

STATES – Ch 13

Cookies: Session cookie is deleted at end of browsing session. Persistent cookies have expiry date set.

In php: setcookie($name, $value, $expiryTime);

$\_COOKIE[]

Serialization: Process of taking complicated object and reducing to string representation for storage or transmission.

In PHP: serialize(); and unserialize();

Session State: session\_start();

$\_SESSION[]

WebStorage is a JS only API introduced in HTML5.4 to replace cookies. Two Types; localStorage, sessionStorage

Caching in PHP: $memcache = new Memcache; $memcache->set();

Information Security: Holistic practice of protecing information from unauthorized users.

Information Assurance: ensures data is not lost when issues arise.

CIA Triad:

Confidentiality: Maintaining pricavy for data being stored/transmitted

Integrity: ensuring data is accurate and correct

Availability: making information available when needed to authorized people

Risk Assessment: actors, impacts, threats, vulnerabilities.

Actors: Internal/External/Partner

Impact: Loss of Availability, Loss of Confidentiality, Loss of Integrity

Threats: particular path a hacker can use to exploit vulnerability.

STRIDE: Spoofing, Tampering, Repudiation (removing all traces of attack), Information disclosure, DOS, Elevation of privilege.

Vulnerabilities: Top 5 classes: Injection, Broken authentication, cross-site scripting, insecure references, security misconfig

Create Policies: Usage, Authentication, Legal

Secure By Design: Requirements -> Design -> Implementation -> Testing -> Deployment

Authentication Factors:

Knowledge Factors: password/pin/question

Ownership Factors: Key/FOB/Card/Phone

Inherence Factors: Fingerprint/signature/DNA/gait

Cipher: Message that is scrambled

Key: number or phrase

Block Ciphers: DES (Data encryption standard) AES (advanced encryption standard)