HTTP message=request/status line+header+body Request line METHOD PATH PROTOCOL\_VERSION

4xx: Client Error 5xx: Server Error

Response ETAG 缓存multiple requests through one connection: reduce connection

MIME (Multi-purpose Internet Mail Extensions) In http Content-Type: text/html

Unicode byte order mark: U+FEFF endian

UTF-8-compatible with asci easy recovery, variable length encoding.

0000 - 007F: 00000000 0zzzzzzz -> 0zzzzzzz //0080 - 07FF: 00000yyy yyzzzzzz -> 110yyyyy 10zzzzzz

0800 - FFFF: xxxxyyyy yyzzzzzz -> 1110xxxx 10yyyyyy 10zzzzzz

<link rel="stylesheet" href="style.css"> <iframe src="b.html" width="200" height="300">

**<form** action="http://www.google.com/search" method="GET"**>**

**<input** name="q" type="text"**>**`**<input** type="submit"**>**

**</form>**

<label for="e1">Email:</label><input type="text" id="e1" name="email">

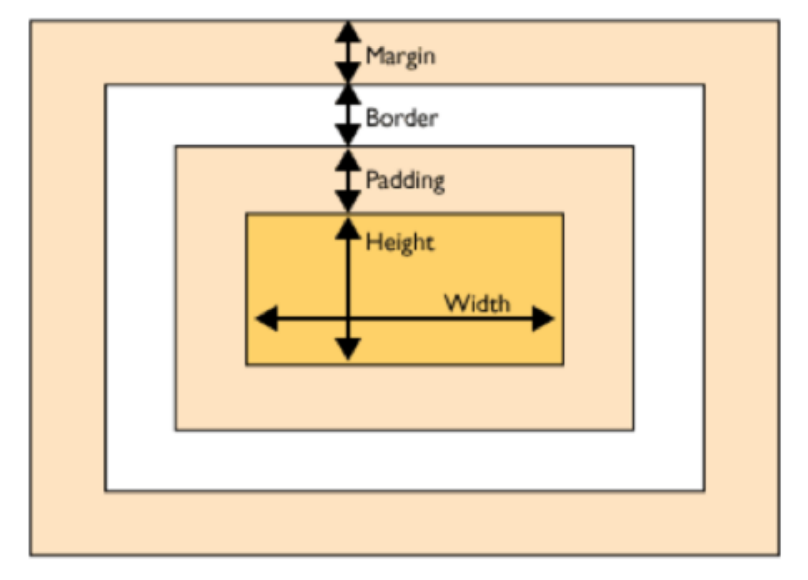
Cascading rule: Specificity Source order //Inheritance

[enabled] has attribute named enabled.

[ target ="\_blank"] attribute " target " has the value " \_blank "

div,p /\* multiple selectors using comma\*/ div p /\* p is a descendent of div \*/

div > p /\* p is a direct child of div \*/ div + p /\* p is the adjacent sibling of (i.e. , immediately follows ) div \*/ div ~ p /\* p is a general sibling of div \*/



1em = 16 px (~ width of M)

margin: 1em 2em 3em 4em; /\* top, right , bottom , left \*/

/\* if any value is missing, use a " symmetric " value \*/

Inline elements do not create a separate block–ignore width, height, margin-top, and margin-bottom properties

Position:

relative: positioned relative to is normal position

absolute: positioned relative to its nearest positioned ancestor

fixed: positioned relative to the “viewport” (viewable client area) static: default value. Element is unpositioned

Responsive Web Design different devices

Fixed: elements have fixed width. Resizing the window does not change the appearance of the page – Fluid: elements uses “percentage” of page width. Elements dynamically resize to fit window width

Viewport

<meta name="viewport" content="width=device-width, initial-scale=1">

Media queries

@media screen and (max-width: 800px) {*/\* CSS rules \*/*}

screen, print, speech, and all (default)

Media features orientation, min-width, max-width, min-height, max-height, resolution, ...

Boolean operators {,=OR, and=AND, not=NOT Precedence: not > and > ,}

CSS Flexbox

display: flex;

flex-direction: row/column;

nav **{flex:** 1 2 100px**;}** */\* flex-growth flex-shrink flex-basis \*/*

#main **{flex:** 2 1 200px**;}**

aside **{flex:** 1 2 100px**;}**

Encryption layer: encrypt transport

HTTP layer: interpret request and serve response

Application layer: generate dynamic content

Storage/Data layer: store and retrieve data

Mapping url->web.xml deployment design/request router

**JSP<**%= request.getParameter/getAttribute("first\_name") %>

Java PrintWriter out = response.getWriter();/request.getParameter("first\_name")

Problem: jsp(template)get too complex/program (how to separate design/logic)

Model-View-Controller (MVC)Model: data storage and access/ View: result presentation

/Controller: application logic

AJAX: Asynchronous JavaScript and XML

Event-driven programming

Programmer specifies mapping from events to actions: callback functions

for (e of array){ stmt; } //loop over array-like elements

try { throw 1; } catch (e if e === ".."){ stmt } finally { stmt }

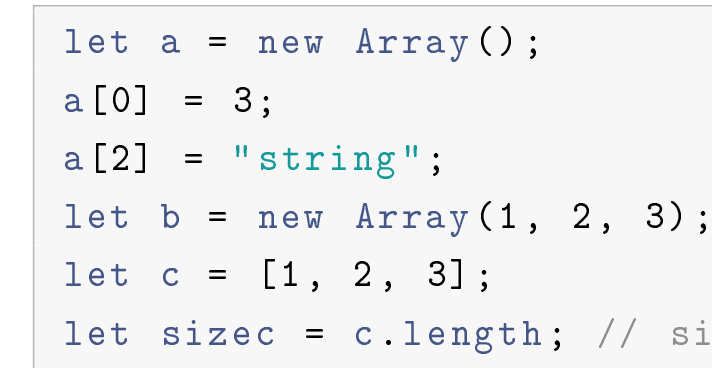
const n = 42, let a=’a’

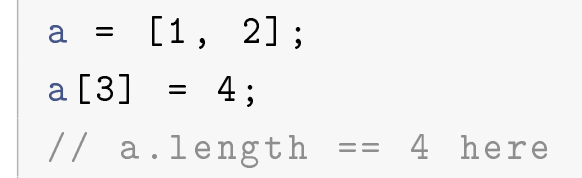
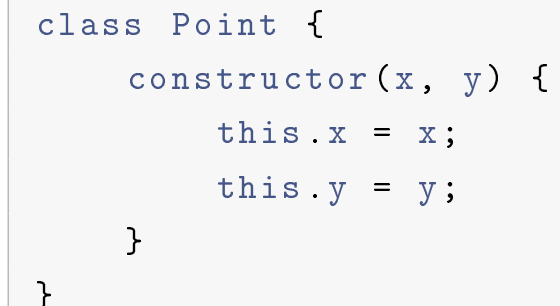
number, string, boolean (null undefined)

false:0,“”,null,undefined,NaN

number<->string implicit conversion, explicitly Number(), String(), Boolean(), parseFloat(),parseInt()

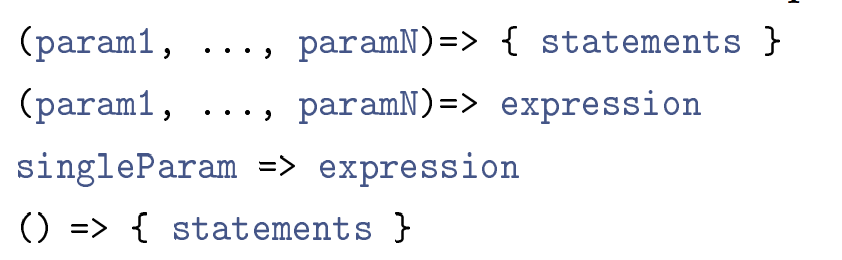
undefined == null ; // true | undefined === null ; // false

JSON , [{ "x": 3, "y": "Good"}, { "x": 4, "y": "Bad"}], cannot be functions or undefined

JSON.stringify(), JSON.parse()

this = DOM element(event handler)

this = window (browser)

anony func keeps its this when declared

export function xx(bb){cc} import { square,xx}/\* from './ lib '; let lib = require ('./ lib ');(nodejs)

DOM element is node, attribute attached to ele node.root(document) root->node(html) document.childNodes[0].childNodes[x]/parentNode Element.attributes

document.getElementByID('id '); document.getElementByTagName('h1 '); document.getElementByClassName ('class ');

nodeType: element, attribute, text, comment

nodeName: tag and attribute names for element and attribute nodes, #text for text node

nodeValue: inside text for text and comment nodes. attribute value for attribute nodes. null otherwise

document.createElement(), document.createTextNode() and appendChild(), removeChild(), replaceChild()

document.body.onclick/ <body onclick="ChangeColor(event);">

onload, onunload, onclick, oninput,onmouseover(if false,f will not be performed)

event bubbuling, x invoke event, all its parent has event async(no docu.write)

window.history.back(), window.history.forward()

alert(), confirm(), prompt()

windows.location: the URL of the current page

By setting this property, we can load a different page

<Book edition="1" xmlns=<http://oak.cs.ucla.edu/cs144> xmlns:s="http://xml.com/shopping">

<ISBN>135-383-9038</ISBN>

<s:Price>$100</s:Price>

</Book>

<a:E1 xmlns:a="http://a.com/"> <b:E2 xmlns:b="http://a.com/"> ab same

function sendAjaxRequest(event){let request = "google-suggest.php?q="+encodeURI(this.value);

xmlHttp.open("GET", request);

xmlHttp.onreadystatechange = showSuggestion;

xmlHttp.send(null);}

// update the page with the response

function showSuggestion() {

if (xmlHttp.readyState == 4) {

htmlCode = xmlHttp.responseText.replace(/</g,'&lt;').replace(/>/g,'&gt;');

document.getElementById("suggestion").innerHTML = htmlCode;}}

Same origin: server(Cross-Origin Resource Sharing)Access-Control-Allow-Origin: <http://www.google.com> JSONP (JSON with Padding)  <script src="url">  myFunc({"x": 10, "y": 20});

Animation: ticker.style.left = String(loc) + "px"; box = document.getElementById("box"); setInterval(shrinkBox, 100);

localStorage["object"] = JSON.stringify(obj); localStorage.removeItem("username"); localStorage.clear();

BACK: use #/ window.location.hash history.pushState(object, title, url) and history.replaceState

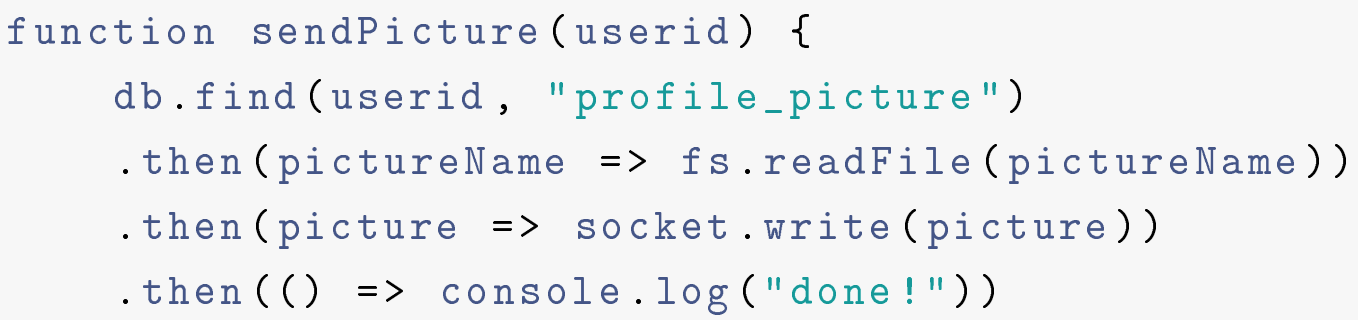
writing **component** classes to manage those **templates**

application logic-> **services**, and (components and services)-> **modules**.

Component directive(<app-search > </app -search>), structural directive, attribute directive

Interpolation{{expr}} Property binding [dom property]="expr" event binding(event)="stat" Two-way binding [(ngModel)]="property"

Node:ejs,middleware



return await doubleAfter2Seconds (x) + await doubleAfter2Seconds (x) + await doubleAfter2Seconds (x); 6s//await a + await b + await c 2s

Transport encryption (SSL)|HTTP server (apache/nginx)|Application server (tomcat/node)|Persistence/Storage layer (MySQL/Mongo)

Content distribution network (CDN)

Scale up: buy a larger, more expensive server Scale out: add more machines

Load balancer to each layer

Replication/partition

Readonly-replication write to both

GFS client-master-file name mapping-(chunkserver chunkserver chunkserver)

Consistency Availability Partition Tolerance

Basically Available: system works basically all the time

Soft-state: does not have to be consistent all the time

Eventual consistency: will be in a consistent state eventually

PREPARE -> VOTE-COMMIT/VOTE-ABORT

Async transaction commit first fail rollback