++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++  
 WPT

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

1. Http 1.1 –text format, line blocking that blocks all request until doesnot get all resources, resource inlining, compresses data itself, default - persistent

Request message send in third part of handshake

Http 2 – binary format, multiplexing so one tcp connection required, push frame by server, hpack for data compression

Request line -> header line -> status line

Application layer

FTP - passive mode control and data 21 , tcp, client server architecture

1. Status code – information response 100-continue, 103-early hints

Success -200 OK, 201 created, 202 accepted, 204 no content

Redirection- 301 moved, 302 found, 304-not modified,

Error – 400-bad request, 401-unathorized, 403-forbidden, 404-not found,405-method not allowed, 406-not accepted,408 timeout, 429-too many request

Server- 500-internal error, 502 – bad gateway, 504-time out,

1. HTML

Align – horizontal (applet, col, hr, iframe, img, table, tb, td, th, tr, caption

Heading h1-h6

Anchor href, target(\_blank, \_parent, \_self, \_top) , type

Paragraph – default css – display :block, margin-top ,bottom :1em, margin-left, right: 0

Img – size for diff layout inline-block

List – ul, ol, -> li

Table - <table> th, td, tr default css border-spacing 2px , grey,

Iframe –

Attributes – rel – for relationship between document

Microdata – itemprops, itemscope, itemid, itemprop,

ARIA accessibility – aria-label

Object tag – container for an external sources

* Events

Window events - onerror, onbeforeprint, onload (after page is finished loading), onmessage, onunload (fires once a page has unloaded, broswers has been closed)

Form Events – onblur, onchange, onfocus, oninput, oninvalid, onsearch, onselect, onsubmit

KeyBorad event- onkeydown, onkeypress, onkeyup

Mouse event - onclick, onmousedown, onwheel

Drag event - ondrag, onscroll

Canvas

Audio - autoplay, controls, loop, muted, src, type

Video - same as audio height, poster, width

Geo-location - navigator.geolocation.watchPosition(), position.coors.location, latitude, speed

* Html forms

Input, textarea, checkbox, dropdown, radio, reset, button,

1. CSS

Internal, external, multiple

Font-family

1. JS(can have $, \_)

Data types - string, number, boolean, null, undefined, symbol, bigint

Var - if we redeclare it will not lose value, can use before declaration

Let - change values, cannot re declare, cannot use before declaration

Const

Comments - // /\*\*/

Scopes - block, local, function

String methods - length, slice, substring, substr, replace, trim(), charAt(), trim()

Number methods - toString(), toExp..(), toPrecision(), valueOf(), toFixed(2)

Date methods - getFullYear(), getDate(), date.parse(),

Arrays Methods - pop(), push(), shift(), unshift(),concat(),splice() adds some , slice() remove

* Objects are mutable

Closure Add = (function(){})()

Const num = new Number(12) ,……………

New Object() - {}

New Array() - []

New RegExp() - /()/

New Function() - (){}

Hoisting - using before declaration class cannot hoisted, function can hoisted

1. Object hierarchy

Window - history + document + location, document - link + form + anchor

DOM elements / events - addeventlistener(), appendChild(), blur(), childer, classList, className, contains(), dir, focus(), getAttribute(), getElementByTagName(), innerHtml, innertext, outerHtml, querySelector(), removeAttribute(),

Forms API - document.forms[index]

1. jQuery

$(document).ready(function(){}

Events - click() , hover(), focus(), blur(), on(), css(), submit(), change(), load(), unload(), scroll(), resize()

Animation - animate()

Traversal -text(), html(), val(), attr(),

Utility - remove(), prop(), parentNode.nodename

1. JSON - json.toString(), json.parse() (no “”)
2. AJAX - async javascript and xml

XMLHttpRequest(), xhttp.onreadystatechage , readystate, status

Open(method, url, async)

Send(“string”)

responseText

getJSON()

1. NODEJS

Fs.readFileSync(‘reas.txt’, ‘utf-8’)

Fs.writeFile(‘read.tx’, data, )

Fs.readFile

Fs.appenfFileSync(), Fs.statSync(), FS.readdirSync(), Fs.existsSync(), Fs.renameSync();

Async - do not block execution of the program and command is executed after previous command

* Http Module

createServer((req, res) => res.writeHead(200, content-type ) res.write(), res.end()

1. Express JS

* Response - res.app(), res.headersSent(), res.locals(), res.append(), res.attachment(), res.cookie(), res.download(), res.status().end(); res.get(‘content-type’) res.json(), res.location(), res.redirect(), res.render(), res.send()
* Request - req.baseurl, req.body, req.cookies, req.ip, req.params, rq.query,
* Middleware - app.use(req, res, next) , app.get,

1. React

* Functional and class components - no render - yes render, runs from top to bottom - class instantiated and different lifecycle method is kept alive, stateless as simply accept data and render it - stateful they have logic and state, no lifecycle methods - yes lifecycle methods,
* Elements and components - it is basic building doesnot have methods just to make it lighter and faster to render (cons tele = <h1></h1> ) , stateless - reusable, return DOM of element (function name() { return ()});
* Life cycle -
  + getDerivedStateFromProps() right before rendering elements
  + componentDidMount() - after component is rendered, useEffect(,[])
  + shouldComponentUpdate() - return Boolean whether react continue with render
  + componentDidUpdate() - after component is updated, useEffect(,[props])
  + componentwillUnmount() - component is removed useEffect( , return )
* State : this.state : {}, this.setState {} : properties belongs to components, component re renders when state is changed
* Props vs state :

Props : read only, immutable, send date from one components to other, stateless components can have props

State :asynchronous, mutable, holds info for components, cannot be accessed by child components, stateless components cannot have state

* React events : onClick(),
* Conditional rendering
* Lists and keys - map, filter, key
* useRef - access to dom nodes, modifying dom elements, useRef.current
* Composition vs inheritance
  + Containment - using props.childern when don’t know about children ahead of time,
  + Specialization - add special cases to the components
* Redux - managing state makes easy when data is changed in store then re - renders
  + Store - entire state of app lists, has dispatch(action)
  + Action - dispatched view which are payload can be read by reducers info of type of action, which will make state changes
  + Reducers - read payloads from actions then update store via state

1. Boot Strap

* Grid
  + .container (fixed width) .container-fluid (full width) .col-, sm, md, lg, xl
  + Typography - list-inline, blockquote text-center, lead, text-muted, class=”h1”
  + Table - table-striped, table-dark, table-bordered, table-hover
  + Image - .img-thumbnail, .img-fluid, round
  + Jumbo tron - showing hero elements, jumbotron, jumbotron-fluid
  + Wells - border around elements - .well -sm, -lg,
  + Alert - alert-primary, secondary, success, danger, warning, info, light, dark, (role-alert) alert-heading, alert-dismissible fade
  + Button - button , btn btn-primary, outlined
  + Button group
  + Badges and labels, badge-pill
  + Progress-bar
  + Pagination
  + List groups - active
  + Panel
  + Drop down
  + Collapse
  + Tabs
  + Navbar active disabled
  + Forms - form-group

1. Web Security

* Cross site Scripting - when attacker uses web application to send malicious code
* Cross site request forgery - attack forces end user to execute unwanted actions
* SQL injection -