|  |  |
| --- | --- |
| .gridContainer {  display: grid;  grid-template-rows: 1fr 8fr 1fr;  grid-template-columns: 8fr 2fr;  grid-template-areas:  "head head"  "content photo"  "footer footer";  }**//Define grid areas in below css tags**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **//Javascript reference**  getElementById("id");  getElementsByTagName("tag");returns array getElementsByClassName("className");  **//button reference**  <button onclick=  **//ES6 for of, only used for arrays or strings**  let campuses=[“x”,”y”,”z”];  for(let i of campuses){console.log(i)}  **//ES6 function parameters, default parameter**  function hello(name=”world”){  return `hello ${name}`}  console.log(hello()) //returns hello world  **//ES6 arrow function**  var sum = (x,y) =>x + y  console.log(sum(3,4)) //displays 7  **//ES6 rest parameters**  function sortColours(...colourNames) {return colourNames.sort(); } //displays the contents of the array, and will display new content if added  **//local storage**  Local storage works very much like cookies except allows far more space than cookies, not auto retransmit every request to server, and is related to browser.  **//Jquery**  Lightweight and contains all the common DOM, event, and AJAX functions  **//Located in head section of html**  <head><script src=“js/jquery.js”></script></head>  **//selector syntax, with options**  $(selector).(html,append,css,val,click,hide,addclass)  **//by inclusion or absence of an attribute:**  $("tag[attribute]").action();  $(tag:not([attribute])").action();  **//Contains another specific tag:**  $("tag:has(tag)").action;  **//Contains specific text:**  $("tag:contains('text')").action();Ex. $("p:contains('Hello')").action(  Starts With **^=** Ends With **$=** Match **~=** | **//Ajax Syntax**  .ajax( {  type: "POST",// GET or POST  url: "example.json",// Address to load  dataType: "json",//html, xml, json, text  success: parseJSON,//function if successful  error: function (e) {}  **//getJSON alternative to AJAX**  getJSON("JSON data file",function(data){});  **//JSONP**  same origin-policy – restricts what can be received from other remote servers and is a safety mechanism. JSONP(JSON with Padding) provides us a way to access data from another origin. It does this by having the server return JSON data wrapped in a function call (the "padding") which can then be interpreted by the browser  $.ajax({ url:  http: something.html,  dataType: "jsonp",  success: jsonCallback});  **//Local vs Session Storage**  Local is persistent, session storage is cleared after the session closes  **//If have multiple tags, can choose which tag to apply changes**  $("h1:**nth(0**)").css("font-size", "1.5em"); $("h1:**first**").html("Title Line"); $("p:**last**").html("Copyright");  **//Mobile Vs Desktop**  Higher latency (slower)  Screen size is smaller  Shorter battery life  Testing and debugging is a challenge  Browsers can behave differently  Mobile users are in a distracted environment and usually have immediate needs  **//Features of mobile**  Portable, personal companion, easy and fast to use, and only one web  **//Questions when building a mobile site**  What is the user looking for within a mobile perspective?  What features do the users want?  Where will the user be when accessing your site?  Will a version be needed for older phones  //Best Practices  Maintain consistency across pages...fonts, colors, etc.  Use Responsive Design for orientation changes  Provide visual separation of sections using colors, for example, instead of whitespace  Change colors based on light conditions such as sun, shade, indoor, low lighting so easy to read  Always include accessibility support |
|  |  |