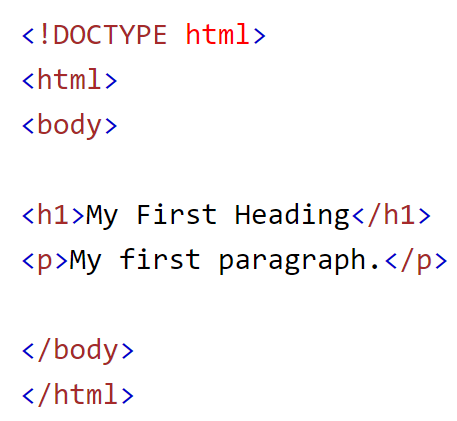
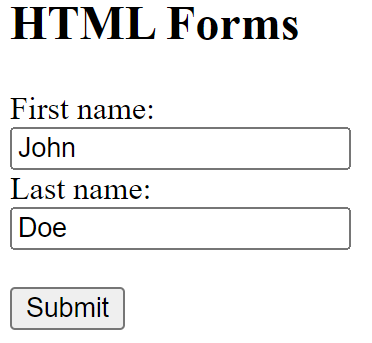
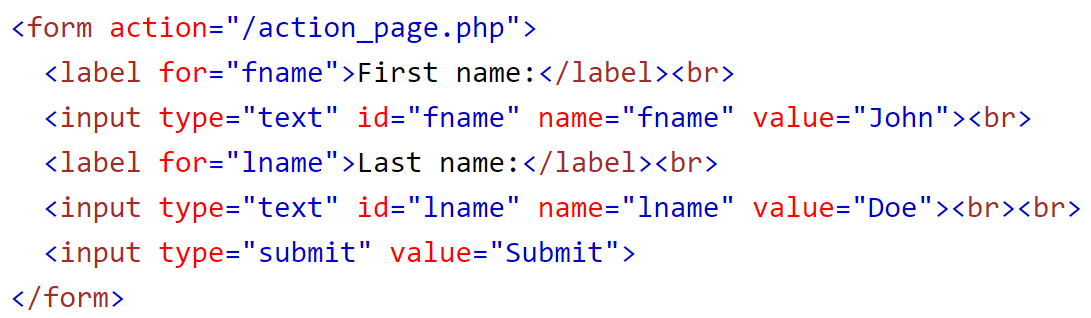
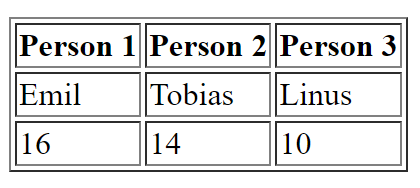
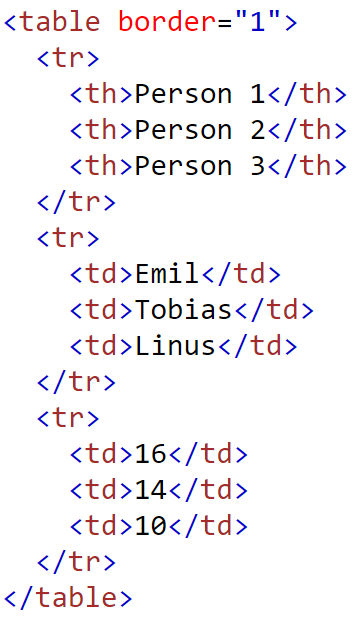
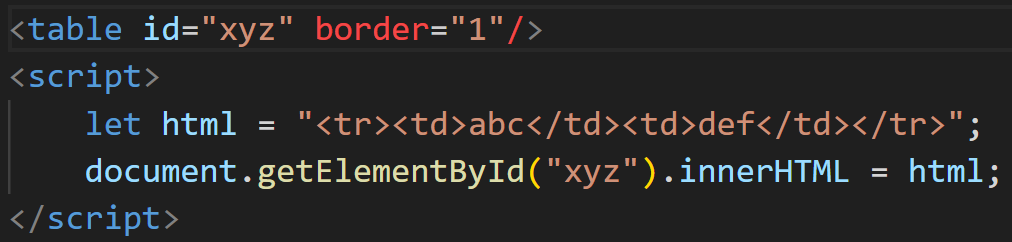
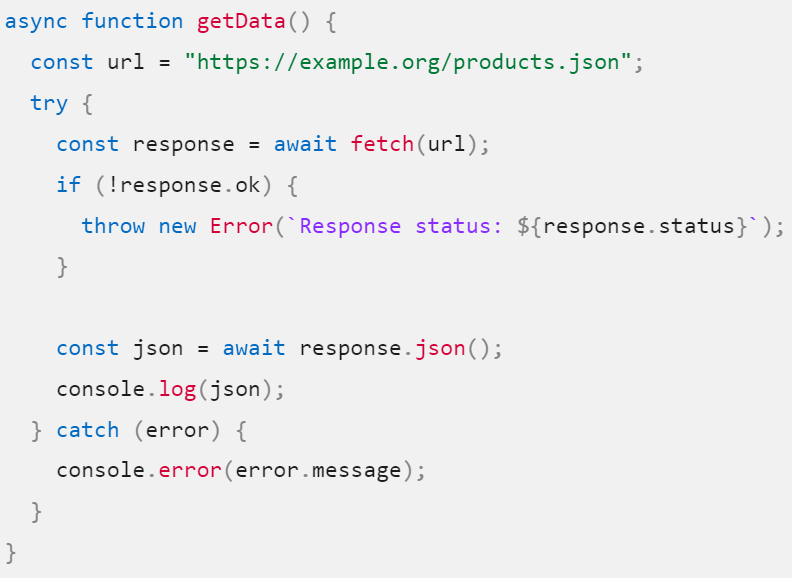
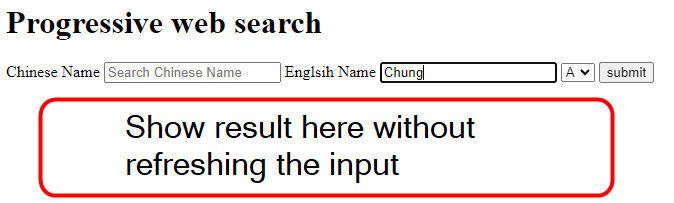
Minimal HTML for front-end to back-end development (Reference: w3school, mozilla)

1. What is HTML -- Hypertext mark-up language
   1. Markup by tags, indicated by < and > pair, e.g. <input>, <form>, etc.
   2. Two common forms
      1. <br/> -- line break, no content within, ended with a /
      2. <b>this is bold</b> -- bold, start with <tag> and ended by </tag>
2. Simple HTML examples -- https://www.w3schools.com/html/html\_basic.asp
   1. <!DOCTYPE html> … <body> - can ignore
   2. </body></html> - can ignore
   3. <h1> -- header 1 (h1 ~ h6, decreasing order)
   4. <p> -- paragraph (wrapping according to browser width)
3. Create/Test an HTML file
   1. Create a text file, key in the html and text, save as “.htm” or “.html” file.
   2. Open the file with Chrome (or drag the file to Chrome)
4. Basic HTML
   1. title
   2. h1 – h6, p, br, &nbsp;
   3. i, b, u – italic, bold, underline
   4. image: <img src="/photos/dog.jpg" alt="a dog" width="100" height="142">
   5. anchor (link): <a href=’’/abc/def.html”>click this</a>
5. Add Javascript in HTML
   1. Inside the file where you need the script, add <script> </script>, put your code between this pair, e.g.  
      <h1>Start my Javascript testing</h1>  
      <script>  
       for (let i=0; i<6; i++) document.write(i + “<br/>”);  
      </script>  
      <h3>Done testing.</h3>
   2. “document” is available in front-end (browser) only, use it to “send” html code
6.   Basic Input Form
   1. /action\_page.php is the “handler” (or API) in back-end
   2. Better add method=”post” after action=
   3. <input type=”submit” value=”Submit”> can be replaced by  
      <button type=”submit”>Submit</button>
   4. Input must have name attribute (name=”abc”) to send data from input to backend
   5. Common input types (note: values are always sent as string)
      1. type=”text”, default
      2. type=”password” – text, but input masked
      3. type=”hidden” – hidden, not shown in browser, cannot be changed
      4. type=”number”, type=”date”
      5. type=”radio” – radio button, one value only
      6. type=”checkbox” – check box, on or off
      7. type=”file” – file upload
   6. 2 other types of button:
      1. type=”reset”, type=”button”; or
      2. <button type=”reset”>Reset</button>, <button type=”button”>XXX</button>
   7. Selection box:   
      <select>  
       <option/>  
       <option value=’HK’>Hong Kong</option>  
       <option value=’CH’>China</option>  
      </select>
   8. Multiline text input : <textarea rows=”3” cols=”80”>xxx yyy zzz</textarea>
   9. If the form involves file uploading, should add the attribute enctype=”multipart/form-data” inside the <form…> tag
7. Basic HTML table (e.g. for listing search result)
   1. table – start and end of table
   2. tr – table row
   3. th – table header (usually bold)
   4. td – table data
8. Sample Javascript code to generate the table content:  
   
   1. id is used to uniquely identify any HTML object
   2. innerHTML is the property table, we update it to shown the dynamic content
9. Sample HTML code to fetch data from back-end API (reference: <https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch>)
10. Javascript events
    1. Format: <element event=**'*some JavaScript*'**> e,g.  
       <button onclick="document.getElementById('demo').innerHTML = Date()">The time is?</button>
    2. Some common events
       1. onclick
       2. onchange
       3. onmouseover
       4. onmouseout
       5. onkeydown
11. Sample HTML code to simulate form submission (without refreshing the page)  
      
      
      
      
    Note: use “new FormData()” if you are including file upload (multipart form), use “new URLSearchParams() ” otherwise. Also, when dealing with file, it will be formData.append(input.name, input), and formData.append(input.name, input.value) otherwise.