

Homework 1-1

Step 1:

I first create a new file and name it index.html,



and I set up basic HTML structure including `<head>` and `<body>` tags.

```
<> index.html X
folder_1.1.1 > <> index.html > html
1  <!DOCTYPE html>
2  <html lang = "zh-TW">
3    <head></head>
4    <body></body>
5  </html>
```

Within the `<head>` tag, I create the desired information.

```
<> index.html X
folder_1.1.1 > <> index.html > html
1  <!DOCTYPE html>
2  <html lang = "zh-TW">
3    <head>
4      <meta charset = "UTF-8">
5      <meta name = "viewport" content = "width = device-width, initial-scale = 1.0">
6      <title>to-do list</title>
7      <link rel = "stylesheet" href = "styles.css">
8    </head>
9    <body></body>
10 </html>
```

Within the `<body>` tag, I create a `<div>` element with an id, called todo-list.

```
index.html X
folder_1.1.1 > index.html > html
1  <!DOCTYPE html>
2  <html lang = "zh-TW">
3      <head>
4          <meta charset = "UTF-8">
5          <meta name = "viewport" content = "width = device-width, initial-scale = 1.0">
6          <title>to-do list</title>
7          <link rel = "stylesheet" href = "styles.css">
8      </head>
9      <body>
10         <div id = "todo-list"></div>
11     </body>
12 </html>
```

Step 2:

I first create a new file named styles.css,



and I add the desired basic CSS styles for the todo-list container.

```
# styles.css X
folder_1.1.2 > # styles.css > #todo-list
1  #todo-list {
2      width: 300px;
3      padding: 10px;
4      border: 1px solid #ccc;
5  }
```

Step 3:

I add an `<input>` and a `<button>` elements in `<div id = "todo-list">` for entering to-do items and adding them to the list. Note that I give id to both elements for further usage in step 4.

```
index.html X
folder_1.1.3 > index.html > html
1  <!DOCTYPE html>
2  <html lang = "zh-Tw">
3    <head>
4      <meta charset = "UTF-8">
5      <meta name = "viewport" content = "width = device-width, initial-scale = 1.0">
6      <title>to-do list</title>
7      <link rel = "stylesheet" href = "styles.css">
8    </head>
9    <body>
10     <div id = "todo-list">
11       <input type = "text" id = "textbar">
12       <button type = "button" id = "Enter">Enter</button>
13     </div>
14   </body>
15 </html>
```

Step 4:

I first create a new file named script.js,



script.js

and I use the `addEventListener()` method to listen for the button “click” event.

```
JS script.js X
folder_1.1.4 > JS script.js
1  document.getElementById("Enter").addEventListener("click", button_clicks);
```

Within the `button_clicks` function, I create a `const` called `inputText` to retrieve input values from the user; and I use the `createElement` method to dynamically generate paragraphs to represent the to-do items.

```
JS script.js X
folder_1.1.4 > JS script.js > button_clicks
1 document.getElementById("Enter").addEventListener("click", button_clicks);
2
3 function button_clicks(){
4     let inputText = document.getElementById("textbar");
5     if (inputText.value.trim() != ""){
6         const new_todo = document.createElement("p");
7         new_todo.textContent = inputText.value.trim();
8         document.getElementById("todo-list").appendChild(new_todo);
9         inputText.value = "";
10    }
11 }
```

I also made some changes in `index.html` to link it to the `script.js`.

```
<> index.html X
folder_1.1.4 > <> index.html > html
1 <!DOCTYPE html>
2 <html lang = "zh-TW">
3     <head>
4         <meta charset = "UTF-8">
5         <meta name = "viewport" content = "width = device-width, initial-scale = 1.0">
6         <title>to-do list</title>
7         <link rel = "stylesheet" href = "styles.css">
8     </head>
9     <body>
10         <div id = "todo-list">
11             <input type = "text" id = "textbar">
12             <button type = "button" id = "Enter">Enter</button>
13             <script src = "script.js"></script>
14         </div>
15     </body>
16 </html>
```

Homework 1-2

Step 1:

I first create a new file and name it index.html,



and I set up basic HTML structure including `<head>` and `<body>` tags.

```
index.html X
folder_1.2.1 > index.html > html
1  <!DOCTYPE html>
2  <html lang = "zh-TW">
3    <head></head>
4    <body></body>
5  </html>
```

Within the `<head>` tag, I create the desired information.

```
index.html X
folder_1.2.1 > index.html > html
1  <!DOCTYPE html>
2  <html lang = "zh-TW">
3    <head>
4      <meta charset = "UTF-8">
5      <meta name = "viewport" content = "width = device-width, initial-scale = 1.0">
6      <title>Guess Number</title>
7      <link ref = "stylesheet" href = "styles.css">
8    </head>
9    <body></body>
10 </html>
```

Within the <body> tag, I use <h1> tag for Guess Number.

```
index.html X
folder_1.2.1 > index.html > html
1  <!DOCTYPE html>
2  <html lang = "zh-TW">
3      <head>
4          <meta charset = "UTF-8">
5          <meta name = "viewport" content = "width = device-width, initial-scale = 1.0">
6          <title>Guess Number</title>
7          <link ref = "stylesheet" href = "styles.css">
8      </head>
9      <body>
10         <h1>Guess Number</h1>
11     </body>
12 </html>
```

Step 2:

I add an <input> and a <button> elements in <body> with the desired format.

```
index.html X
folder_1.2.2 > index.html > html
1  <!DOCTYPE html>
2  <html lang = "zh-TW">
3      <head>
4          <meta charset = "UTF-8">
5          <meta name = "viewport" content = "width = device-width, initial-scale = 1.0">
6          <title>Guess Number</title>
7          <link ref = "stylesheet" href = "styles.css">
8      </head>
9      <body>
10         <h1>Guess Number</h1>
11         <input type = "number" id = "guess-input" placeholder="enter your guess">
12         <button type = "button" id = "submit guess">Submit</button>
13     </body>
14 </html>
```

Step 3:

I first create a new file named styles.css,



and I add the desired basic CSS styles for h1 and guess-input.

```
# styles.css X
folder_1.2.3 > # styles.css > #guess-input
1 h1 {
2   font-size: 48px;
3 }
4
5 #guess-input {
6   width: 500px;
7 }
```

Step 4:

I first create a new file named script.js,



script.js

use Math.floor and Math.random to generate a random number between 1 and 100 and assign it to a const called randNum,

```
JS script.js X
folder_1.2.4 > JS script.js > ...
1 const randNum = Math.floor(Math.random() * 100) + 1;
```

and apply the addEventListener() method to listen for the button “click” event.

```
JS script.js X
folder_1.2.4 > JS script.js > ...
1 const randNum = Math.floor(Math.random() * 100) + 1;
2 document.getElementById("submit guess").addEventListener("click", feedback);
```

Within the feedback function, I create a variable called `guess_input` to retrieve input numbers from the user. I use several if-else statements to compare the input numbers with the answer (i.e. `randNum`) and provide feedback until the user guesses the right number.

```
JS script.js X
folder_1.2.4 > JS script.js > feedback
1  const randNum = Math.floor(Math.random() * 100) + 1;
2  document.getElementById("submit guess").addEventListener("click", feedback);
3
4  function feedback(){
5      let guess_input = document.getElementById("guess-input").value;
6      if (guess_input == randNum)
7          document.write("Congrat! You guess the right number!");
8      else if (guess_input > randNum && guess_input < 101)
9          alert("Guess lower!");
10     else if (guess_input < randNum && guess_input > 0)
11         alert("Guess higher!");
12     else
13         alert("Invalid input!");
14 }
```

I also made some changes in `index.html` to link it to the `script.js`.

```
<> index.html X
folder_1.2.4 > <> index.html > html
1  <!DOCTYPE html>
2  <html lang = "zh-TW">
3      <head>
4          <meta charset = "UTF-8">
5          <meta name = "viewport" content = "width = device-width, initial-scale = 1.0">
6          <title>Guess Number</title>
7          <link rel = "stylesheet" href = "styles.css">
8      </head>
9      <body>
10         <h1>Guess Number</h1>
11         <input type = "number" id = "guess-input" placeholder = "enter your guess">
12         <button type = "button" id = "submit guess">Submit</button>
13         <script src = "script.js"></script>
14     </body>
15 </html>
```