

언어/프로그래밍 기본 개념들

〈HTML/CSS/JS〉



HTML:

Hypertext Markup Language

'화면들에 이것들이 이런 구조로 놓여있어라' 하고 갖다놓는 수단

CSS:

Cascading Style Sheets

'HTML이 올려놓은 것들을 이렇게 보이도록 해라'라고 꾸며주는 문서
언어 축에도 못 낄

JavaScript:

원래는 browser에서 웹사이트를 돌리는 목적으로 만들어졌었는데, node.js가 브라우저 JS를 바깥 세상으로 꺼내오면서 위상이 올라감.

웹사이트에서 돌아가는 JS는 browser에서 다양한 일을 수행하고 HTML로 올려놓은 요소들을 변형시키거나 직접 만들어낸다.

즉 3개는 한 SET라고 봐도 됨.

갖다놓고(HTML) 꾸미고(CSS) 시킨다(JS).

>> 근데 이걸 혼자 다 하는게 VUE / REACT / SVELTE // SPRING BOOT도 비슷한 느낌?

```
index.html x
1 <html>
2 <head>
3   <link rel="stylesheet" href="style.css">
4   <script defer type="text/javascript"
5     src="script.js"></script>
6 </head>
7 <body>
8   <div id="calculator">
9     <span>알파한 계산기</span><br>
10    <input id="formula-input"
11      type="text"
12      placeholder="수식을 입력하세요."/>
13    <div id="calc-history"></div>
14  </div>
15 </body>
16 </html>
```

```
style.css x
1 #calculator {
2   background-color: #ffbb24;
3   border-radius: 12px;
4   width: 240px;
5   margin: 24px;
6   padding: 24px;
7   text-align: center;
8 }
9
10 #calculator span {
11   font-size: 1.5em;
12   font-weight: bold;
13   color: white;
14   text-shadow: 0px 0px 2px rgba(0, 0, 0, 0.33);
15 }
16
17 #calculator #formula-input {
18   width: 100%;
19   margin-top: 8px;
20   height: 36px;
21   line-height: 36px;
22   font-size: 1.1em;
23   letter-spacing: 3px;
24   border: 0;
25   text-align: center;
26 }
27 #calculator #formula-input:focus {
28   outline-width: 0;
29 }
30
31 #calculator #calc-history div {
32   height: 36px;
33   line-height: 36px;
34   margin-top: 1px;
35   background-color: rgba(255, 255, 255, 0.8);
36 }
37 #calculator #calc-history div.invalid {
38   color: tomato;
39   font-weight: bold;
40 }
```

```
script.js x
1 var formulaInput = document.getElementById("formula-input");
2 var calcHistDiv = document.getElementById("calc-history");
3
4 formulaInput.addEventListener("keyup", function(e) {
5   if (e.code === "Enter")
6     calculate();
7 })
8
9 function calculate () {
10
11   // 입력칸의 문자열이 사칙연산 형식이 맞는지 확인
12   var fm = formulaInput.value;
13   var formulaRegex = /^\.d+(.d+)?[+|-*\/]{1}\.d+(.d+)?$/;
14   var formulaValid = formulaRegex.test(fm);
15
16   var resultText = "노";
17   if (formulaValid) {
18     // 형식에 맞을 시 식을 계산하고 결과 문자열을 설정
19     var answer;
20     eval('answer=' + fm);
21     resultText = fm + " = ";
22     resultText
23     += (answer % 1 > 0 ? answer.toFixed(2) : answer.toString());
24   }
25
26   // calc-history 상자에 넣을 또 다른 상자를 생성하고 내용을 설정한 뒤 삽입
27   var resultDiv = document.createElement("DIV");
28   resultDiv.appendChild(document.createTextNode(resultText));
29   if (!formulaValid)
30     resultDiv.classList.add("invalid");
31   calcHistDiv.insertBefore(resultDiv, calcHistDiv.firstChild);
32
33   // 입력칸은 빈칸으로
34   formulaInput.value = "";
35 }
```

알파한 계산기

수식을 입력하세요.

노

$324.98 - 98.5742 = 226.41$

$578 + 42 = 620$

위의 HTML / CSS / JS로 만들어진 계산용 웹페이지

〈MVC 웹 프레임워크〉

M: Model - database에 저장되는 data의 형식을 지정하고 save/load 하는 작업들에 관한 code들이 model part에서 이루어진다

V: View - html/css 등의 요소들이 View part에서 작성

C: Controller - model의 data를 view에 연결해서 사용자가 GUI화면을 통해 data를 읽고 쓰고 지우고 할 수 있도록 전반적인 제어를 하는 part

MVC 웹 프레임워크- 동적 웹을 만들 때 쓰는. MVC 구조의 기본 설계가 갖춰진 상태인 웹 프레임워크.

웹을 만들기 위해 기본적인 골격을 갖춘 프로그램이라고 생각하면 될듯.

대표적인게 Java 언어로 동작하는 **Spring** 프레임워크

Python의 Django(애는 MCV라고 안 하고 MTV(Model/Template/View)라고 하긴 함)

Ruby의 Ruby on Rails

출처: <https://youtu.be/AERY1ZGoYc8>

〈프론트엔드〉 - 사용자 컴퓨터의 브라우저에서 돌아감

〈백엔드〉 - 서버에서 돌아감

〈SPA 프레임워크〉

: Browser에서 동작하는 JavaScript언어로 만들어진 프레임워크 >> 프론트엔드에서 동작함.

- 즉 사용자의 컴퓨터에서 동작함. 댓글 하나 달리는데 그걸 다시 서버로 가져가서 정보 가져오고 다시 또 사용자의 컴퓨터로 전송하는 비효율적인 단계가 필요없이, 그냥 사용자의 컴퓨터에서 댓글에 관한 부분만 작업하도록 하는.. 그래서 Single Page Application임
즉, Browser에서 최초에 한 번만 page 전체를 load하고 이후부터는 특정 부분들은 Ajax란 기술을 통해 데이터를 바인딩하는 방식.

Vue / Angular(망함) / React /Svelte가 대표적인 SPA 프레임워크

출처: <https://youtu.be/iE29ljbbow0>

-지속적으로 UPDATE 할 것-