언어/프로그램 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 낌

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).

index.html

```html
<html>
  <head>
    <link rel="stylesheet" href="style.css">
    <script defer type="text/javascript"
      src="script.js"></script>
  </head>
  <body>
    <div id="calculator">
      <span>얄짝한 계산기</span><br>
      <input id="formula-input"
        type="text"
        placeholder="수식을 입력하세요."/>
      <div id="calc-history"></div>
    </div>
  </body>
</html>
```

style.css

```css
#calculator {
  background-color: #ffbb24;
  border-radius: 12px;
  width: 240px;
  margin: 24px;
  padding: 24px;
  text-align: center;
}

#calculator span {
  font-size: 1.5em;
  font-weight: bold;
  color: white;
  text-shadow: 0px 0px 2px rgba(0, 0, 0, 0.33);
}

#calculator #formula-input {
  width: 100%;
  margin-top: 8px;
  height: 36px;
  line-height: 36px;
  font-size: 1.1em;
  letter-spacing: 3px;
  border: 0;
  text-align: center;
}
#calculator #formula-input:focus {
  outline-width: 0;
}

#calculator #calc-history div {
  height: 36px;
  line-height: 36px;
  margin-top: 1px;
  background-color: rgba(255, 255, 255, 0.8);
}
#calculator #calc-history div.invalid {
  color: tomato;
  font-weight: bold;
}
```

script.js

```js
var formulaInput = document.getElementById("formula-input");
var calcHistDiv = document.getElementById("calc-history");

formulaInput.addEventListener("keyup", function(e) {
  if (e.code === "Enter")
    calculate();
})

function calculate () {

  // 입력칸의 문자열이 사칙연산 형식이 맞는지 확인
  var fm = formulaInput.value;
  var formulaRegex = /^\d+(.\d+)?[+\-*/]{1}\d+(.\d+)?$/;
  var formulaValid = formulaRegex.test(fm);

  var resultText = "노";
  if (formulaValid) {
    // 형식에 맞을 시 식을 계산하고 결과 문자열을 설정
    var answer;
    eval('answer=' + fm);
    resultText = fm + " = ";
    resultText
      += (answer % 1 > 0 ? answer.toFixed(2) : answer.toString());
  }

  // calc-hitory 상자에 넣을 또 다른 상자를 생성하고 내용을 설정한 뒤 삽입
  var resultDiv = document.createElement("DIV");
  resultDiv.appendChild(document.createTextNode(resultText));
  if (!formulaValid)
    resultDiv.classList.add("invalid");
  calcHistDiv.insertBefore(resultDiv, calcHistDiv.firstChild);

  // 입력칸은 빈칸으로
  formulaInput.value = "";
}
```

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(얘는 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 / React가 대표적인 SPA 프레임워크

출처: https://youtu.be/iE29lbjbow0

-지속적으로 UPDATE 할 것-