

<<< JavaScript & Form >>>

html : 웹페이지의 content 정의

css : 웹페이지의 layout, style 정의

javascript : 웹페이지에 action 추가 (Dom 구조를 변화시킴)

JS에서 Style 변경시키기

```
document.getElementById(id).style.property = new style;
```

```
document.getElementsByClassName(class)[0].style.property = new style;
```

```
array.push(element) == array[array.length] = element
```

testContent : element에 텍스트 추가/값 가져오기, innerHtml과 거의 동일

form : server와 request-response cycle을 할 수 있음

label : inline

☆☆ name vs id ☆☆

name과 id는 모두 html의 input의 속성이다. 둘 다 element들을 구분해서 접근하는데 사용하는 공통점이 있다.

JavaScript에서 name은 document.getElementsByName('name')[i]에 사용하고,

id는 document.getElementById(id')에 사용한다.

name은 서버로 보내는 HTTP request에 사용할 때 element를 구분하기 위해 사용하고, id는 javascript, css에서 사용할 때 element를 구분하기 위해 사용한다.

추가로 name은 서버로 보내는 HTTP request에 사용되기 때문에 서버단에서 접근이 가능하지만 id는 접근이 불가능하다.

<<< D3.JS >>>

D3.js : 주어진 데이터를 시각적으로 표현하는 JavaScript 라이브러리

☆☆ D3.js의 특징 ☆☆

- 아이디어에 따라 다양한 그래프를 그릴 수 있음 (그래프가 고정된 일반적인 그래프 그리기 라이브러리(ex. chart.js)나 API와 가장 큰 차이점)
- 애니메이션 or 버튼 조작에 따른 상호작용 가능 (애니메이션이 다른 그래프 라이브러리와 비교하여 아주 부드럽게 동작)
- 스마트폰이나 테블릿에서도 동작

<<< D3.JS 실습 >>>

[html]

<svg>안에 <text>의 속성 text-anchor : start(글씨의 start가 중심선에), middle(글씨의 middle이 중심선에), end(글씨의 end가 중심선에)

[css]

stroke : 도형의 선 색

stroke-width : 도형의 선 굵기

opacity : 투명도

[js]

select("#myGraph")

selectAll("rect")

data(dataSet)

enter()

`append("svg/rect")`

`attr("x/y/width/height", 숫자/function(d, i) { return ... })`

`on("click", function() { ... })`

`transition()` : (부드럽게 변함) 개체의 어떤 특성을 어떻게 변환시킬 것이라는 것을 알림

`duration(1500)` : 변환이 ?밀리초 동안 진행 (1500밀리초 = 1.5초)

`delay(1500)` : 변환이 ?밀리초 뒤에 진행, i*밀리초 주의

`style("fill/stroke", "색")`