

# (디지털컨버전스) 스마트 콘텐츠와 웹 융합 응용SW개발자 양성과정

---

2021년 6월 14일  
[ 26일차 복습 ]

- 수강생 : 김 민 규
- 강의장 : 강 남 C
- 수강 기간 : 2021. 05. 07 ~ 2021. 12. 08
- 수강 시간 : 15:30 ~ 22:00
- 이상훈 강사님 | 이은정 취업담임



## ▶ 내용 : <a>

```
1 + <!DOCTYPE html>
2 + <html lang="en">
3 + <head>
4 +     <meta charset="UTF-8">
5 +     <title>Title</title>
6 + </head>
7 + <body>
8 +     <h1>하이퍼링크!</h1>
9 +     <a href="https://golang.org/">Golang Homepage</a>
10 +     <h1>그림에 하이퍼링크!</h1>
11 +     <a href="https://golang.org/">
12 +         
13 +     </a>
14 + </body>
15 + </html> ⊖
```

## 주요 내용

--

<a href="주소">글자 </a>  
글자를 누르면 해당 주소에 접속이 된다.

글자가 아닌 그림을 삽입할 경우에도  
마찬가지로 적용이 된다.

--


2 + <html lang="en">
3 + <head>
4 +     <meta charset="UTF-8">
5 +     <title>Title</title>
6 + </head>
7 + <body>
8 + <div id="container">
9 +     <div id="whoami" style="width:200px; height:200px; float:left">
10 +         <h1>Name</h1>
11 +         <p>어마마마</p>
12 +     </div>
13 +     <div id="detail" style="width:200px; height:200px; float:left">
14 +         <h1>그다음 올 문장은 ?</h1>
15 +         <p>통촉하여 주시옵소서!</p>
16 +     </div>
17 +     <div id="img" style="width:200px; height:200px; float:left">
18 +         
19 +     </div>
20 +     <div style="width:200px; height:200px; float:right">
21 +         <p>고노 히카리와 ?!</p>
22 +     </div>
23 + </div>
24 + </body>
25 + </html> ☹
```

## 주요 내용

--

〈Div〉

Width

Height

Float

Div는 각각 구간을 분리해준다.

옆의 코드에서는

Container안에 whami와 detail, img를  
각각 분리하여 웹페이지에 표시가 된다.

또한 float을 통해 작성된 고노히카리와의  
경우 분리된 공간 중 맨 오른쪽정렬이  
이루어졌다.

즉 style float은 정렬하는 기능을 가진 코드

--

## ▶ 내용 : form

```
1 + <!DOCTYPE html>
2 + <html lang="en">
3 + <head>
4 +     <meta charset="UTF-8">
5 +     <title>Title</title>
6 + </head>
7 + <body>
8 +     <!-- Web에서는 크게 Get, Post 방식의 데이터 전송법이 있다.
9 +         여태까지 Get을 활용하였는데 이번에는 Post를 사용해봤다.
10 +         Post를 사용하는 이유에 대해서는 Spring 진도를 조금 더 나간 후에 진행하도록 한다.
11 +         form의 디폴트 method는 get이고 post를 사용할 경우 아래와 같이 사용한다.
12 +         또한 action은 맵핑될 URL 주소를 적는다.
13 +         method가 get이면 GetMapping, post면 PostMapping을 사용한다. -->
14 +     <form action="/memberRegister" method="post">
15 +         <input type="text" title="e-mail">
16 +         <input type="submit" value="가입">
17 +     </form>
18 + </body>
19 + </html> ☹
```

```
@PostMapping("/memberRegister")
public String memberRegister() {
    log.info("memberRegister()");

    // return "json 데이터 출력!";
    // redirect: 의 경우 특정한 URL 위치로 URL 맵핑을 강제할 수 있다.
    return "redirect:/";
}
```

## 주요 내용

--

<form>

Action

Method

Action은 맵핑이 될 url주소를 적고,  
Method 자리의 경우  
GetMapping, PostMapping을  
그에 맞추어  
get또는 post를 적어줘야한다.

--

<Input>

Type=submit -> 양식을 전송하는 버튼  
Title=e-mail -> 이메일을 작성할 수 있게  
되며 유효성 검증 매개변수가 존재  
Value - > 양식 컨트롤의 현재 값.  
이름/값 짝(name/value pair)의  
일부로서 양식과 함께 전송된다

--

Return :redirect:/"

특정 url위치로 url맵핑을 강제한다.  
즉 '/' url로 이동하게 된다.