

[WEB Basic Study]

Hello, world!

오늘부터 WEB 클라이언트
프로그래밍을 배워봅시다!
어렵지 않아요 ㅎㅎㅎ
는 땡!

저는 우재우재우재우 입니다.
만나서 반가워요!
저도 부족해서 뭘 알려줄 수
있을까 걱정입니다 ㅠ

건축 디자인 전공! -> 자퇴 ->
NEXT 게임 클라이언트 전공!
-> NEXT WEB UI 전공!
나는야 똥테크 총대장!!!

저도 BASIC 밖에 못 알려드려요
같이 공부한다고 생각하고 합시다
쉽고 재미있게 WEB을 알 수 있도록
열심히 도와드리겠습니다 ㅋ

이번 스터디 학습 목표

WEB과 WEB UI가 무엇이고 필요한 기술이 무엇인지 안다.
클라이언트 개발에 대한 대략적인 이해를 갖는다.
클라이언트 프로그래밍에 필요한 HTML5 기술을 배운다.
만들기 원하는 웹 페이지를 만들어(그려)낼 수 있다.
2학기부터 진행할 개발경험프로젝트에서 이 정도는 해주자.

WEB이 뭔가요?

Browser, 넌 누구냐

우리는 다양한 기기에 있는 Browser를 통해 웹을 쉽게 경험하고 있어요!
Notebook, Mobile, Tablet, Smart TV etc...

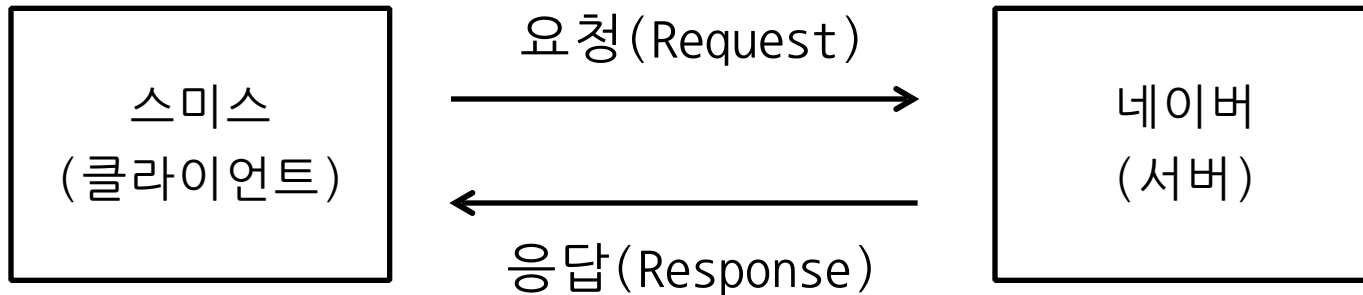
Browser가 인터넷만을 위한 도구라고
생각하겠지만 생각보다 Browser는
다른 일도 할 수 있어요!

그림 파일이나 텍스트 파일을
끌어다 넣어 봅시다!



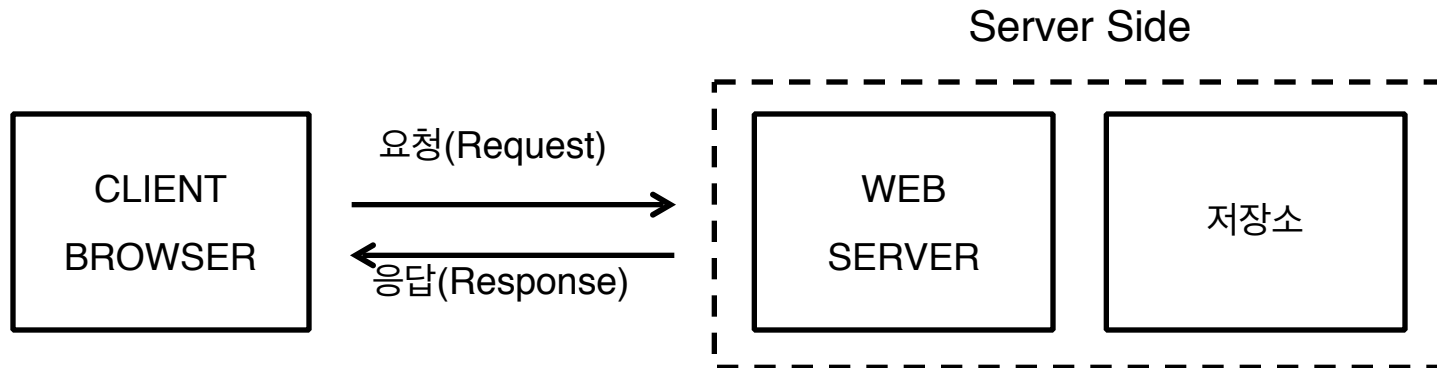
WEB은 이렇게 동작해요!

설마 NEXT에 와서 서버와 클라이언트를 처음 들어본 건 아니겠죠?
아, 그럴 수 있어요 ㅋㅋㅋ



가야 할 주소가 있고, 구체적인 요청사항이 있고, 약속된 방식이 있어요!

좀 더 자세히 봅시다



- 1) 특정 URI로 서버에 접근하여 resource를 요청한다. (Request)
- 2) 서버에서는 필요한 정보가 있는지 저장소를 확인한다.
- 3) 저장소에 있는 정보를 서버가 받아서 클라이언트에게 보낸다.
- 4) 클라이언트는 서버가 보내준 정보를 받는다.
- 5) 사용자가 쉽게 알아볼 수 있도록 Browser가 UI 형태에 맞게 보여준다.

감사합니다! 구글!!!

구글 개발자 도구를 켜봅시다!

윈도우라면 F12, 맥이라면 Command + Alt + I

Network 탭을 살펴보세요~

개발자 도구와 많이 친해지면 좋아요 :)

개발자 도구에게 인사해볼까요? 안녕~

그렇다면 WEB UI는 뭐지?

WEB UI에 대한 오해

눈에 보이는 부분만 개발하는 개발자다?

WEB UI 개발자는 웹 페이지를 예쁘게 만드는 디자이너다?

그래서 디자인 감각이 뛰어나야 한다?

서버 전공에 비해 공부가 적어도 괜찮다?

브라운 그리기

2014년 5월 23일

프로그램 연습(4분반)

c# 코드로 그림그리기 실습



〈 원 본 〉

1. 재우



2. 예찬



3. 철오



4. 중일



5. 용은



6. 승희



7. 상수



8. 승현



9. 원철(Paul)



누가 제일

잘 그렸나요?

역시 풀ㅋㅋㅋ

이상적인 UI 개발자?

UI 구성요소 간의 관계를 잘 이해하는 개발자!

만들고 싶은 UI를 상상한대로 만들 수 있는 개발자!

통신의 앞뒤를 모두 이해하는 개발자!

WEB UI를 구현하려면...

WEB Design

HTML

CSS

Java Script

HTML

CSS

Javascript

```
week1.html
1 <!doctype html>
2 <html>
3 <head>
4   <title>MyBlogHOME</title>
5   <meta charset="utf-8">
6   <link rel="stylesheet" type="text/css"
7     href="./reset.css" />
8   <link rel="stylesheet" type="text/css"
9     href="./week1.css" />
10 </head>
11 <body>
12   <header>
13     <div class="mainTitle"><p>MyBlogHOME
14       .com</p></div>
15     <div class="photo">
16       
17     </div>
18   </header>
19   <section>
20     <ul>
21       <li>
22         <div class="cardsContainer">
23           <div class="date">
24             February 26, 2015 at
25             SitePoint</div>
26           <div class="carsTitle">
27             Article 1</div>
28           <div class="button"></div>
29         </div>
30       </li>
31       <li>
32         <div class="
33           cardsContainer">
34           <div class="date">February 26,
35             2015 at SitePoint</div>
36           <div class="carsTitle">Article 2
37             </div>
38           <div class="button"></div>
39         </div>
40       </li>
41       <li>
42         <div class="cardsContainer">
```

```
week1.css
1 body {
2   margin: 0 auto;
3   width: 70%;
4 }
5
6 header {
7   position: relative;
8   width: 100%;
9   height: 120px;
10 }
11
12 .mainTitle {
13   position: absolute;
14   width: 90%;
15   height: 70px;
16   border-bottom: 2px solid #CCC;
17 }
18
19 .mainTitle p {
20   line-height: 70px;
21   font-size: 32px;
22   color: #333;
23 }
24
25 .photo {
26   position: absolute;
27   right: 10%;
28   top: 20px;
29 }
30
31 section {
32   float: left;
33   width: 65%;
34 }
35
36 .cardsContainer {
37   width: 100%;
38   height: 60px;
39   border-bottom: 2px solid #CCC;
40 }
41
42 .date {
43   font-size: 10px;
```

```
week3_OOP.js
1 //OOP js 1
2 function mySNS() {
3   this.name = "LINE";
4   return this.name;
5 }
6
7 var mySNS = {
8   name : "Line",
9   getName : function() {
10     return this.name;
11   }
12 }
13
14 function mySNS() {
15   this.name = "KAKAO";
16   this.getName = function() {
17     return this.name;
18   }
19 }
20 var myObj = new SNS();
21 myObj.getName();
22
23
24 //OOP js Pattern
25
26 //1. JS Object 활용
27 var oName = {
28   name : "js",
29   getName : function() {
30     return this.name;
31   }
32 }
33
34 //2. Constructor pattern
35 function Name(name) {
36   this.name = name;
37
38   this.getName = function() {
39     return this.name;
40   };
41
42   this.setName = function(newName) {
43     this.name = newName;
```

WEB UI 삼대장

이 정도 생각하면 편할까요?

HTML ————— 명사

CSS ————— 형용사

Java Script ————— 동사

실제로 어떻게 동작하죠?

百聞而不如一見

크롬 개발자 도구에서 살펴봅시다!

개발환경 세팅!

우리는 서브라임 텍스트를 쓸 것이외다.

그냥 배경이 까만 메모장이죠.

그렇지만 메모장보단 재미있는 기능이 많아요.

Brackets, codepen, jsbin.com 등 다른 툴도 많으니 살펴보시길...



이제 우리는
HTML이란 것을
해볼 것이오!!!

HTML(Hyper Text Markup Language)?

정의를 쓰기 너무 귀찮으니 그냥 위키를 봅시다!

HTML 맛보기!

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello, HTML!</title>
  </head>
  <body>
    <p>Hello, world!</p>
  </body>
</html>
```

HTML은 이 모양 이 꼴

요것은 태그(tag)요,



```
<a href= “ http://www.google.com ” >구글구글</a>
```



요것은 애트리뷰트(attribute)요.

W3schools에서 <a>태그를 더 자세히 알아보시다.

http://www.w3schools.com/tags/tag_a.asp

OMG !!!

HTML tag는 100개가 넘는다!

여기 가면 볼 수 있어요~

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element>

BLOCK vs INLINE

BLOCK element

Block element는 넓이와 높이를 지정할 수 있어요.

영역을 차지하고 있어서 위/아래/위/위/아래로 늘어섭니다.

<div>, <p>, , , <form> 등

INLINE element

Inline element는 block element 안에서 스타일을 넣기 위해 사용됩니다.

좌에서 우로 흐르듯이 쌓이죠.

, , <q>, , <input> 등

<div>: 내가 그렇게 만만하니

div(division) tag는 가장 많이 사용되는 block element입니다.

화면의 영역을 구별하는 데에 쓰이지만 특별한 의미를 갖진 않아요.

와꾸(레이아웃)를 잡을 때 편하게 쓰시면 됩니다!

**divdivdivdivdivdivdivdiv
divdivdivdivdivdivdivdiv
divdivdivdivdivdivdivdiv
divdivdivdivdivdivdivdiv**

솔까말 div만 써도 웹페이지 레이아웃을 잡을 수는 있어요!

속성(attribute)으로 id나 class를 주면 서로 구분 가능하겠죠?

먼저 id와 class의 차이는 무엇일까요?

만약 div로 다 가능하면 다른 block element들은 왜 있는 거죠?

Semantic HTML

적절한 곳에 적절한 태그를 쓰는 것을 추천!

div로 떡칠해도 되지만 의미를 생각해서!

<http://semantic-ui.com/kitchen-sink.html>



<u1>, , <d1> 삼총사

<u1>, , <d1>은 무언가를 열거할 때 쓰이는 List용 태그입니다!

실제로 코딩 해보고 어떻게 생겼는지 봅시다.

각각의 차이점이 무엇인가요?

다른 태그들은 쓰면서 배우죠

관찰...죠??

뭐 만들까요?

<http://magazine.good.is/>

이거 만듭시다!!! 참엔 힘들 거예요;;;

저를 미워하게 될지도...

다음 시간까지 HTML 구조만 잡아와 봅시다!

질문 및 건의사항

ex) 먹으면서 수업합시다. 제대로 좀 가르치쇼. 너 못 생겼다. 등등