Node.js 연결 및 HTML / CSS

Web 개발 기초

1. Node.js 기본 연결

2. HTML / CSS

1. Node.js 기본 연결

Node.js?

NodeJS 는 구글 크롬의 자바스크립트 엔진 (V8 Engine) 에 기반해 만들어진 서버 사이드 플랫폼입니다.

비동기 I/O 처리 / 이벤트 위주: Node.js 라이브러리의 모든 API는 비동기식입니다, 멈추지 않는다는거죠 (Non-blocking). Node.js 기반 서버는 API가 실행되었을때, 데이터를 반환할 때까지 기다리지 않고 다음 API 를 실행합니다.

빠른 속도: 구글 크롬의 V8 자바스크립트 엔진을 사용하여 빠른 코드 실행을 제공합니다.

단일 쓰레드 / 뛰어난 확장성: Node.js는 이벤트 루프와 함께 단일 쓰레드 모델을 사용합니다. 이벤트 메커니즘은 서버가 멈추지않고 반응하도록 해주어 서버의 확장성을 키워줍니다. 반면, 일반적인 웹서버는 (Apache) 요청을 처리하기 위하여 제한된 쓰레드를 생성합니다. Node.js 는 쓰레드를 한개만 사용하고 Apache 같은 웹서버보다 훨씬 많은 요청을 처리할 수 있습니다.

1. Node.js 기본 연결

Node.js?

서버 사이드 언어로 Javascript 를 쓴다.

>>> 프론트 엔드와 백 엔드를 하나의 언어로 다룰 수 있다.

1. Node.js 기본 연결

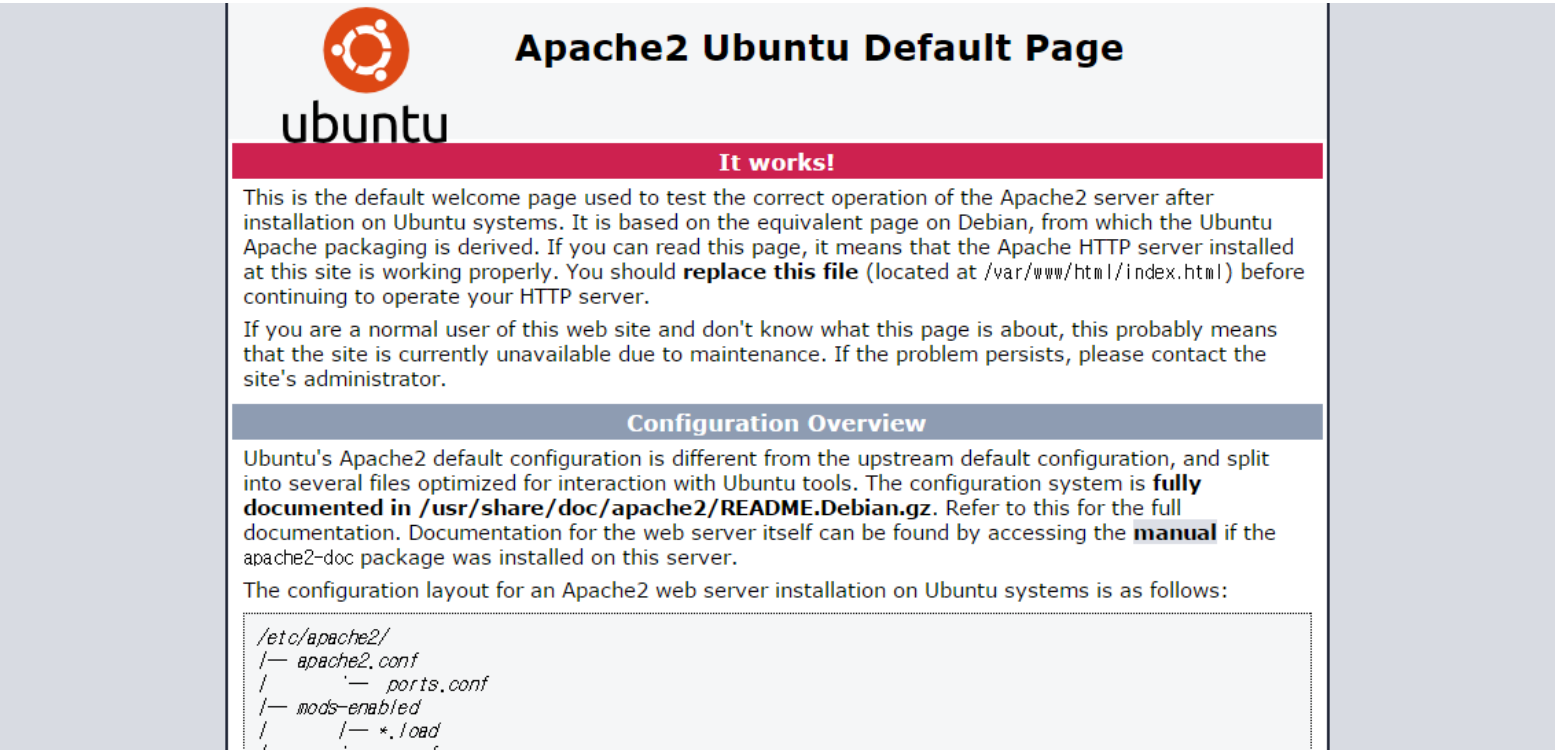
Node.js - http 연결

먼저 웹브라우저에 자신의 public ip를 입력해보자

1. Node.js 기본 연결

Node.js - http 연결

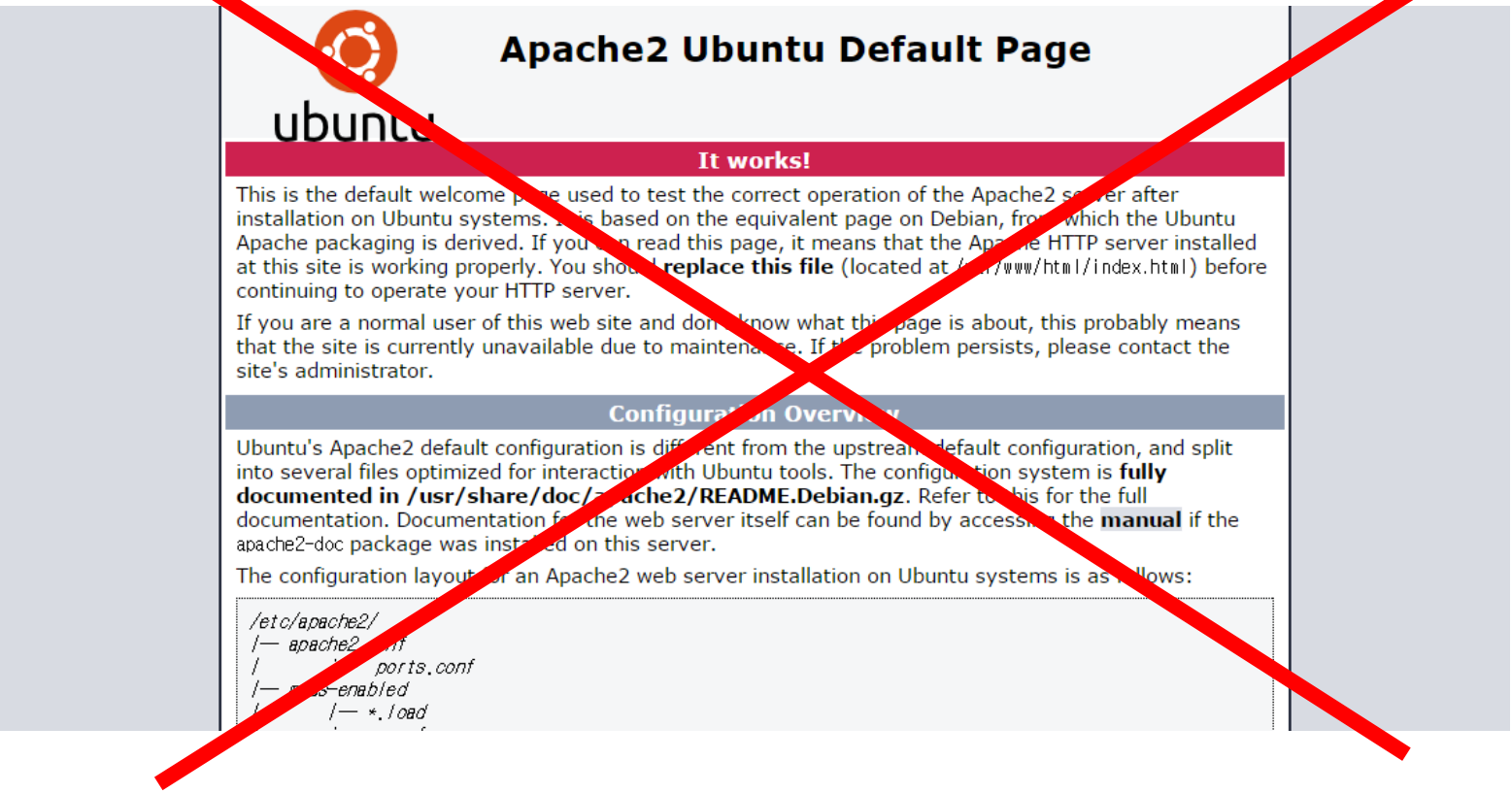
이런 Apache2 Default Page 가 나온다.



1. Node.js 기본 연결

Node.js - http 연결

우리는 우리가 만든 페이지가 나오게 하고 싶다!



1. Node.js 기본 연결

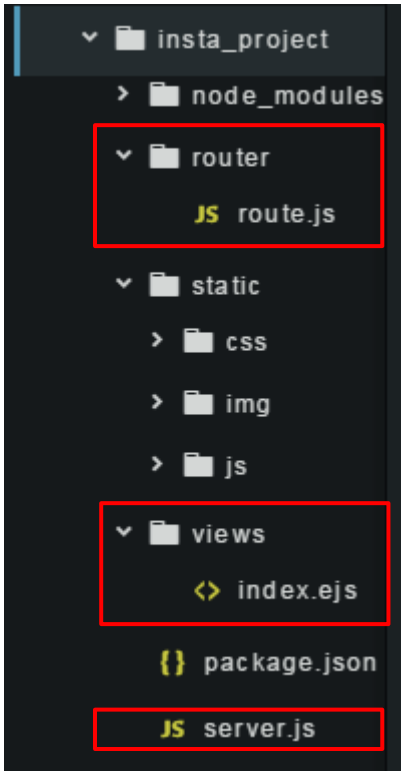
Node.js - http 연결

먼저 Atom 에서

최 상단 폴더에 server.js

Router 폴더에 route.js

Views 폴더에 index.ejs 파일을 만들어준다.



1. Node.js 기본 연결

Node.js - http 연결

먼저 server.js 파일 작성

```
var express = require('express');
var path = require('path');
var app = express();
var bodyParser = require('body-parser');

app.use(express.static(path.join(__dirname, 'static')));
app.set('views', __dirname + '/views');
app.set('view engine', 'ejs');
app.engine('html', require('ejs').renderFile);

var server = app.listen(3000, function(){
 console.log("Express server has started on port 3000");
});

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({
  extended: true
}));

var router = require('./router/route')(app);

```

1. Node.js 기본 연결

Node.js - http 연결

그다음 route.js 파일 작성

```
1
2   module.exports = function(app, fs, passport, upload)
3   {
4       app.get('/',function(req,res){
5           res.render('index',{
6               title: "insta_project"
7           });
8       });
9
10  };
```

1. Node.js 기본 연결

Node.js - http 연결

마지막으로 index.ejs 작성

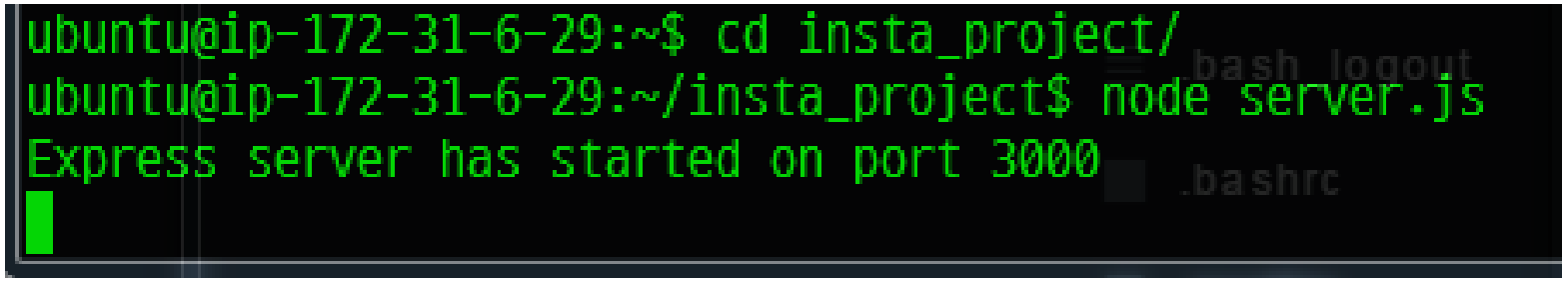
```
<html>
  <head>
    <title>
      <%=title%>
    </title>
  </head>
  <body>
    <div>
      안녕하세요!
    </div>
  </body>
</html>
```

1. Node.js 기본 연결

Node.js - http 연결

터미널을 열고
$ cd insta_project
$ node server.js

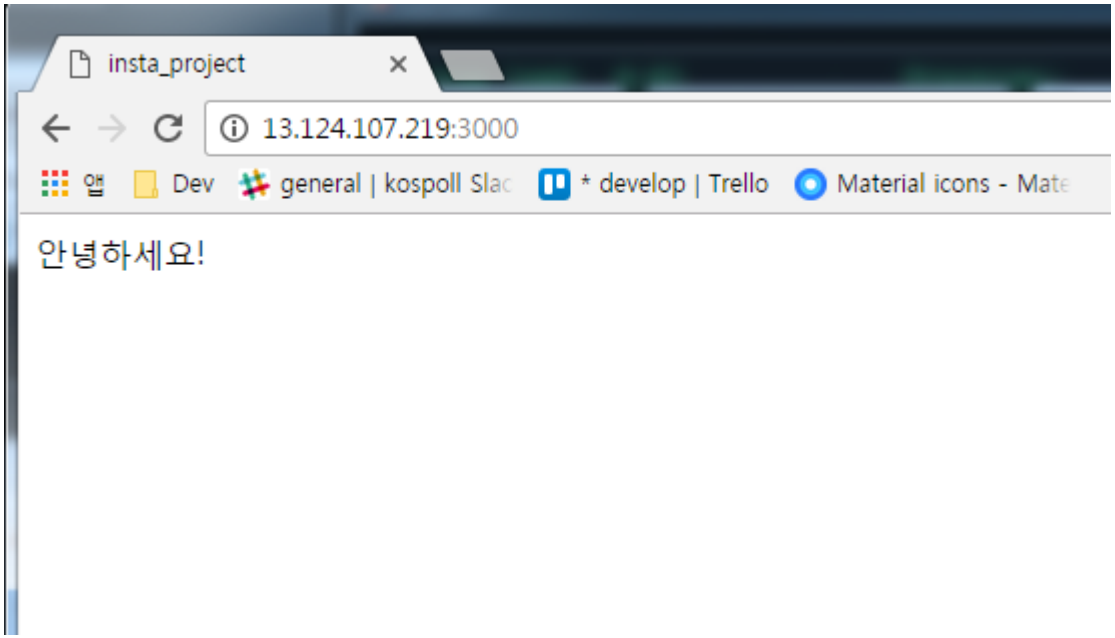
를 하면 http 연결이 실행된다.



1. Node.js 기본 연결

Node.js - http 연결

웹브라우저를 켜고
http://본인public ip:3000
을 입력하면 내가 만든 index.ejs 파일이 뜬다!



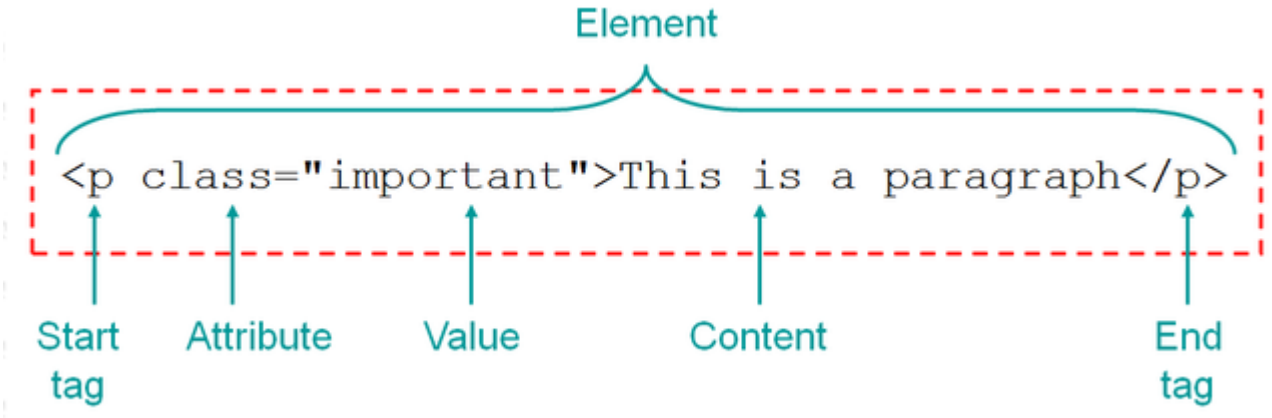
2. HTML / CSS - 1

HTML ?

HTML(HyperText Markup Language)

마크업 언어(Markup Language)

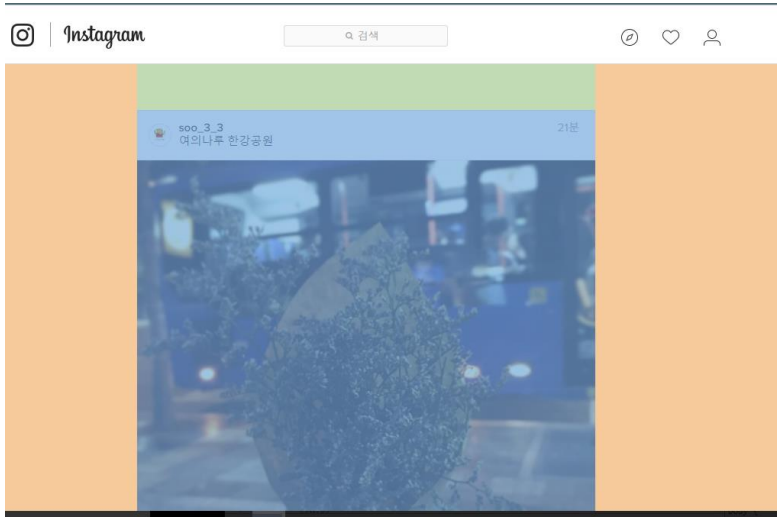
HTML은 제목, 단락, 목록 등과 같은 본문을 위한 구조적 의미를 나타내는 것뿐만 아니라 링크, 인용과 그 밖의 항목으로 구조적 문서를 만들 수 있는 방법을 제공한다.



2. HTML / CSS - 1

HTML ?

브라우저에서는 이렇게 보이는 것이



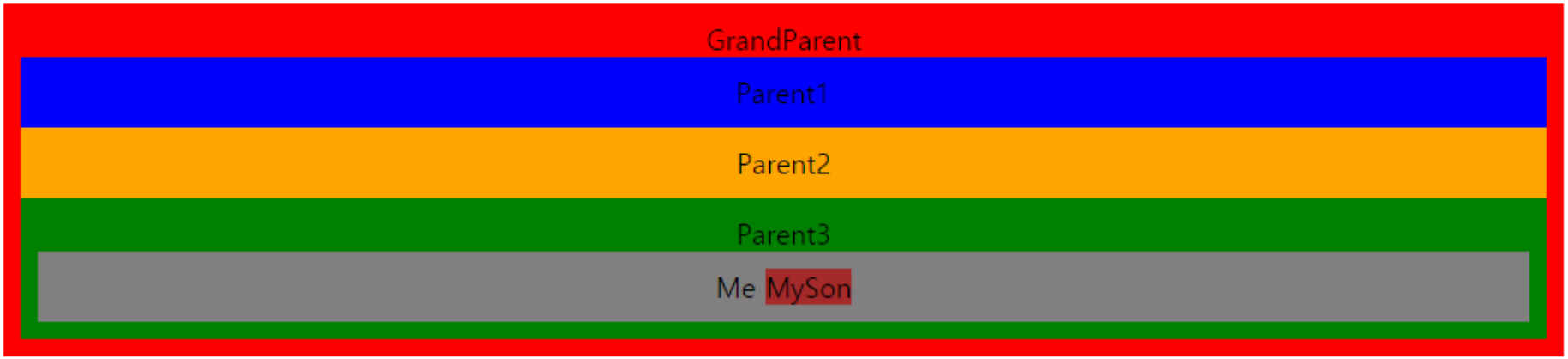
HTML 코드로는 이렇게 보인다!

```
<html lang="ko" class="js logged-in wf-proximanova-n3-active wf-proximanova-n4-active wf-proximanova-n6-active wf-proximanova-n7-active wf-active">
  <!--<![endif]-->
  ▶<head>…</head>
  ▼<body class>
    ▼<span id="react-root">
      ▼<section data-reactroot class="_8f735"> == $0
        ▶<main class="_6ltyr _rnpza" role="main">…</main>
        ▶<nav class="_onabe _5z3y6" role="navigation">…</nav>
        ▶<footer class="_oofbn" role="contentinfo">…</footer>
          <!-- react-empty: 743 -->
        </section>
      </span>
```

2. HTML / CSS - 1

HTML ?

HTML 언어를 작성할 때 중요한 것은 많지만.. 그 중에 몇 가지를 꼽는다면.
- 태그가 선언된 순서
- 태그의 부모,자식,형제 관계



2. HTML / CSS - 1

HTML ?

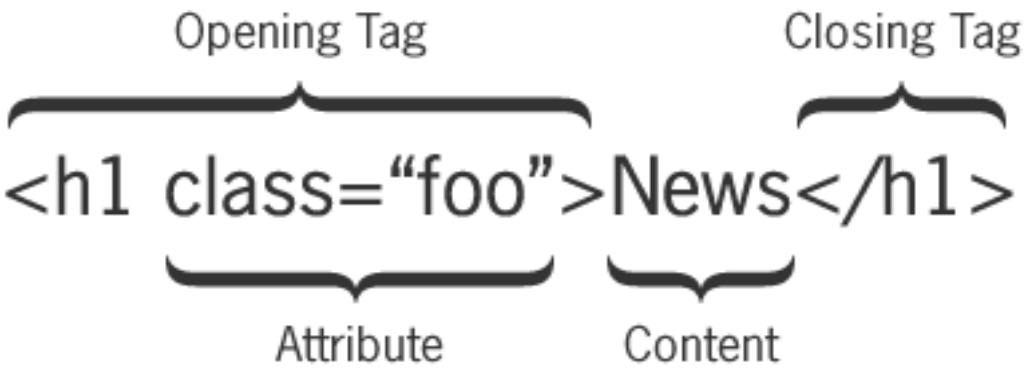
HTML 언어를 작성할 때 중요한 것은 많지만.. 그 중에 몇 가지를 꼽는다면.
- 태그가 선언된 순서
- 태그의 부모,자식,형제 관계

```
<h1>Family Tree</h1>
<div class="GrandParent">
    GrandParent
    <div class="Parent1">
        Parent1
    </div>
    <div class="Parent2">
        Parent2
    </div>
    <div class="Parent3">
        Parent3
        <div class="Me">
            Me
            <span class="MySon">MySon</span>
        </div>
    </div>
</div>
```

2. HTML / CSS - 1

HTML ?

HTML Tag란 무엇인가?
=> Html 문서를 구성하는 요소들. 이런 태그들을 element 라고도 표현한다.



2. HTML / CSS - 1

HTML ?

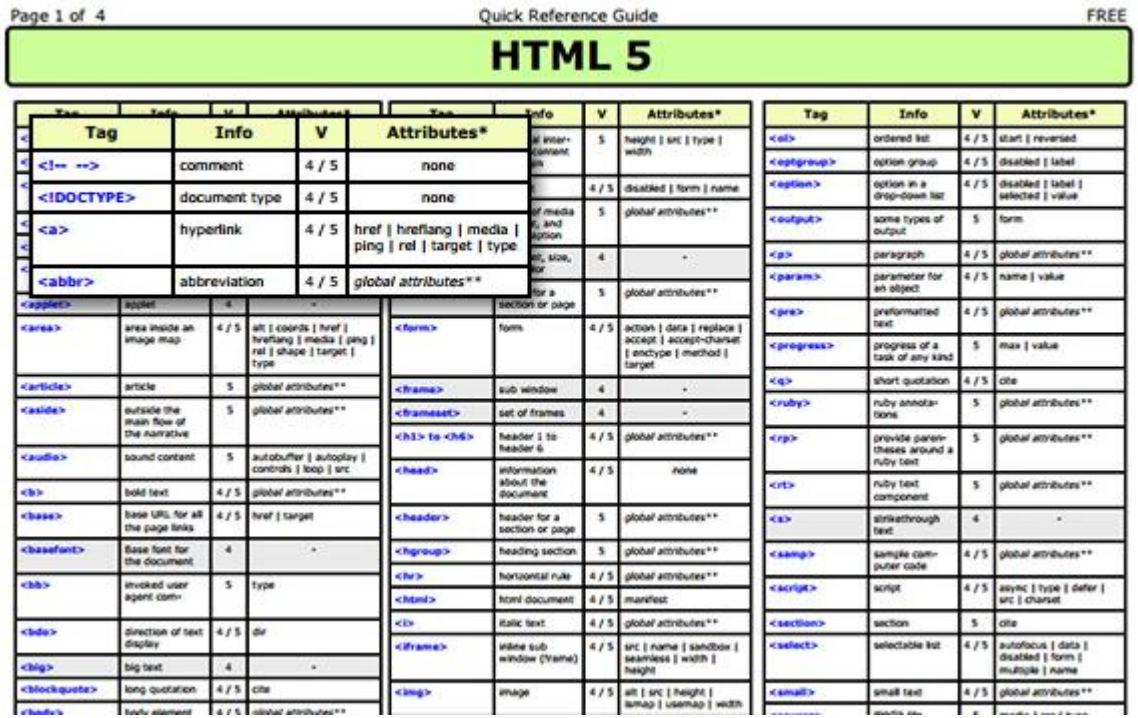
많이 쓰이는 태그들

** ** : division 또는 section 분리에 쓰이지만, 속성을 마구 추가해주면, 많은 것들에 쓰인다.
**** : 일반적으로 텍스트들을 감쌀 때 쓰인다. 마찬가지로 속성을 마구 추가해주면, 많은 것들에 쓰인다.
<a> : anchor의 약자로, 보통 하이퍼링크를 적용시킬 때 쓰인다.
** ** : paragraph의 약자로 문장/문단에 쓰인다.
**
** : break의 약자로, 워드에서 엔터 처럼 쓰인다.
<img/> : image를 로딩 할 때 쓰인다.
**
** : list의 약자로 연속되는 콘텐츠들을 나열 할 때 쓰인다.
<script></script> : 자바스크립트를 추가하거나, 내부에 선언 할 때 쓰인다.
<link/> : css파일등과 같은 파일들을 추가할 때 쓰인다.
<title></title> : 브라우저 최상단에 뜨는 이름에 쓰인다.
<meta/> : 로봇들에게 제공할 정보들에 쓰인다.
<input/> : 텍스트 또는 파일 등과 같은 유저가 입력 가능한 액션을 제공한다.
<table></table>: 표를 만들 때 기본적으로 쓰이는데, 사실 엄청 강력한 기능이다.

2. HTML / CSS - 1

HTML ?

Tag의 종류들은?



Page 1 of 4 Quick Reference Guide FREE

HTML 5

Tag	Info	V	Attributes*
<!-- -->	comment	4 / 5	none
<!DOCTYPE>	document type	4 / 5	none
<a>	hyperlink	4 / 5	href | hreflang | media | ping | rel | target | type
<abbr>	abbreviation	4 / 5	*global attributes***

Tag	Info	V	Attributes*
<applet>	applet	4	-
<area>	area inside an image map	4 / 5	alt | coords | href | hreflang | media | ping | rel | shape | target | type
<article>	article	5	*global attributes***
<aside>	outside the main flow of the narrative	5	*global attributes***
<audio>	sound content	5	autobuffer | autoplay | controls | loop | src
<b>	bold text	4 / 5	*global attributes***
<base>	base URL for all the page links	4 / 5	href | target
<basefont>	Base font for the document	4	-
<bb>	invoked user agent com-	5	type
<bdo>	direction of text display	4 / 5	dir
<big>	big text	4	-
<blockquote>	long quotation	4 / 5	cite

Tag	Info	V	Attributes*
		5	height | src | type | width
		4 / 5	disabled | form | name
		5	*global attributes***
		4	-
		5	*global attributes***
<form>	form	4 / 5	action | data | replace | accept | accept-charset | enctype | method | target
<frame>	sub window	4	-
<frameset>	set of frames	4	-
<h1> to <h6>	header 1 to header 6	4 / 5	*global attributes***
<head>	information about the document	4 / 5	none
<header>	header for a section or page	5	*global attributes***
<hgroup>	heading section	5	*global attributes***
<hr>	horizontal rule	4 / 5	*global attributes***
<html>	html document	4 / 5	manifest
<i>	italic text	4 / 5	*global attributes***
<iframe>	inline sub window (frame)	4 / 5	src | name | sandbox | seamless | width | height
<img>	image	4 / 5	alt | src | height | ismap | usemap | width

Tag	Info	V	Attributes*
	ordered list	4 / 5	start | reversed
<optgroup>	option group	4 / 5	disabled | label
<option>	option in a drop-down list	4 / 5	disabled | label | selected | value
<output>	some types of output	5	form
<p>	paragraph	4 / 5	*global attributes***
<param>	parameter for an object	4 / 5	name | value
<pre>	preformatted text	4 / 5	*global attributes***
<progress>	progress of a task of any kind	5	max | value
<q>	short quotation	4 / 5	cite
<ruby>	ruby annota- tions	5	*global attributes***
<rp>	provide paren- theses around a ruby text	5	*global attributes***
<rt>	ruby text component	5	*global attributes***
<s>	strikethrough text	4	-
<samp>	sample com- puter code	4 / 5	*global attributes***
<script>	script	4 / 5	async | type | defer | src | charset
<section>	section	5	cite
<select>	selectable list	4 / 5	autofocus | data | disabled | form | multiple | name
<small>	small text	4 / 5	*global attributes***

엄청 많다. 그럼 이것들을 다 외워야 할까?

2. HTML / CSS - 1

HTML ?

정답은 Yes and No

쓰다 보면 외워지고, 쓰다 보면 쓰던 것들만 쓰게 된다.

2. HTML / CSS - 1

CSS ?

종속형 시트 또는 **캐스케이딩 스타일 시트**(Cascading Style Sheets, CSS)는 마크업 언어가 실제 표시되는 방법을 기술하는 언어로, HTML과 XHTML에 주로 쓰이며, XML에서도 사용할 수 있다. W3C의 표준이며, 레이아웃과 스타일을 정의할 때의 자유도가 높다.

- Html에 스타일을 입히기 위한 언어

- HTML style?

HTML엘리먼트에 속성을 부여할 수 있는데, 그 중에 외형적인 모습에 관한 속성을 붙일 수 있다.

Ex) color, backgroun-color,width,height등등

크게 3가지 방법이 있는데.

1) inline-style : html 태그에 직접 적용

ex)

2) internal-style : head 태그 안에 <style>태그로 입력

3) external-style : 따로 확장자 css로 끝나는 파일로 작성

2. HTML / CSS - 1

CSS ?

Inline-style 방식

index.php | style.css | style_explaination.php

```
1  <!DOCTYPE HTML>
2  <head>
3  <title>인스타그램 만들거야!</title>
4  <meta charset="UTF-8">
5  </head>
6  <body>
7      <div style="width:100px;height:100px;background-color:#000;"></div>
8  </body>
```

Internal-style 방식

```
1  <!DOCTYPE HTML>
2  <head>
3  <title>인스타그램 만들거야!</title>
4  <meta charset="UTF-8">
5  <style>
6  div {
7      width: 100px;
8      height: 100px;
9      background-color: #000;
10 }
11 </style>
12 </head>
13 <body>
14     <div></div>
15 </body>
```

2. HTML / CSS - 1

CSS ?

External-style 방식

```html
<!DOCTYPE HTML>
<head>
<title>인스타그램 만들거야!</title>
<meta charset="UTF-8">
<link rel="stylesheet" href="styles/external.css" />
</head>
<body>
    <div></div>
</body>
```

external.css | index.php | style.css

```css
div {
    width: 100px;
    height: 100px;
    background-color: #000;
}
```

2. HTML / CSS - 1

CSS ?

세가지 방법 모두 결과는 같지만, 장단점이 있다.

- **Inline-style**

장점 : 엘리먼트에 직접 입력 가능해서, 직관적이고 코딩하기 편하다.

단점 : 여러 엘리먼트에 동일한 스타일을 적용시킬 때는 코드가 길어진다.

- **internal&external style**

장점 : 코드가 정리가 되고, 여러 엘리먼트에 동시적용이 가능하다. 큰 프로젝트의 경우 오히려 코드가 짧아진다. 다양한 트릭들을 적용할 수 있다.

단점 : big-picture 없이 짜게 되면, inline-style과 별반 차이가 없을 수 있다. 수정 할 때 번거로울 수 있다.

>>>>> 결론은 두 가지를 골고루 섞어 써줘야 한다.

2. HTML / CSS - 1

CSS의 응용 – 1. 반응형 웹

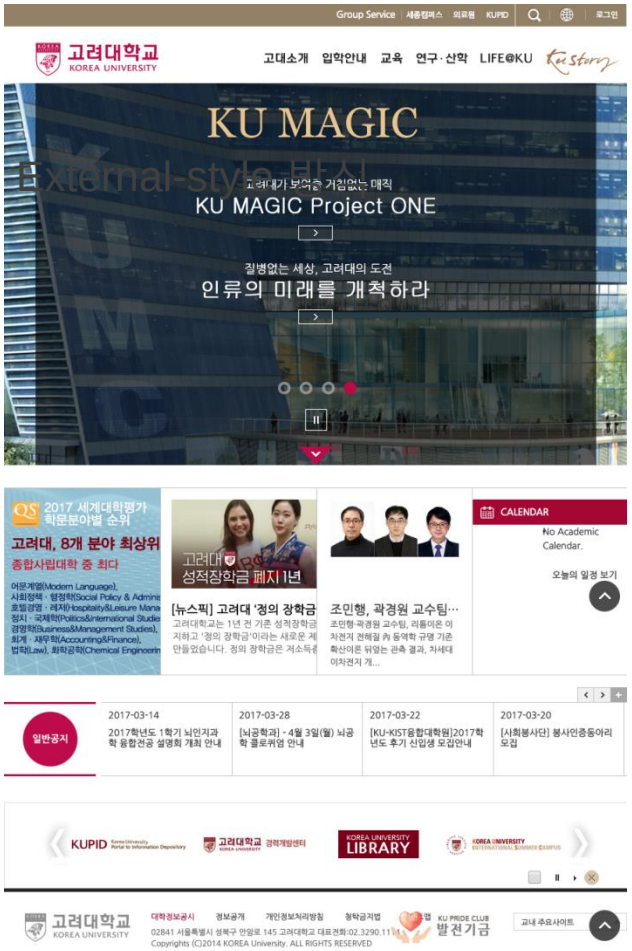
Responsive web(반응형 웹) : 브라우저의 크기에 따라 html 구성요소들의 크기 및 위치를 조정하는 웹 테크닉

>>> 스마트폰이 나오고 나서 모바일의 스크린 사이즈를 지원해야 하는 경우가 생김.

- 크게 3가지를 보통 지원함 : PC, 테블릿, 스마트폰
- 만약 이렇게 만들지 않으면, 세 가지 경우에 대해 새롭게 만들거나, 하나의 모습으로만 세 가지에 다 보여줘야 함.

>>>Css의 media-query라는 기능을 이용해서 적용함.

2. HTML / CSS - 1



PC 화면

모바일 화면

2. HTML / CSS - 1

CSS의 응용 – 2. 특정 요소에만 스타일 적용

특정한 상황에 해당하는 엘리먼트에 대해 '제한적' 스타일을 적용 할 수 있다.

EX) - 짝수 번째 엘리먼트

- 특정 태그의 밑에 있는 엘리먼트

마지막 엘리먼트의 라인 없애기나, 테이블에서 짝수 번째 행에 대에서는 다른 색을 주는 것들

인라인 스타일을 쓰게 되면, 일일이 스타일을 다르게 입력 해줘야 한다. 또한 부모에 의한 제한 적인 태그는 구현 불가하다. 이럴 때 css의 기술을 이용!

자. 이제 만들어 보자!

만드는 건 좀 쉬고 와서 하자.

다음 시간에는…

1. 기초 형태 만들기

2. 디테일 한 형태 만들고 겉 모양 완성