Week 11

HTML (HyperTextMarkupLanguage) :

웹 상에서 정보를 입력하고 보여주는 문서를 작성할 때 사용하는 언어

HTTP(HyperTextTransferProtocol): 프로토콜 - 서버와 브라우저간의 통신적 약속?개념

브라우저와 서버의 요청(request)/응답(response) 과정을 약속한 양식과 규칙의 체계

주로 html문서를 주고받는데 사용한다. 현재 1.1 버젼까지 나왔다

80번 포트를 기본으로 사용한다.

HTTP를 통해 전달되는 자료는 http:로 시작하는 URL(Uniform Resource Locator)(인터넷 주소)(네트워크 상 리소스가 어디있는지 알려주기 위한 약속(일종의 집주소라고 생각하면 된다.)로 조회할 수 있다.)

인터넷 작동원리

브라우저(사용자) 파일이나 정보 요청(url) -> http 프로토콜로 요청 - > 서버 인식 후 해당 파일 조사해서 응답 -> http 프로토콜로 응답 -> 브라우저 수신

브라우저로 편지를 써서 우체부(http)에게 주면 우체부가 편지를 분류해 해당 주소(URL)에 있는 사람에게 보내고 편지 받은 사람이 편지 보낸 내용을 읽고 편지내용에 요구하는 물건이나 정보를 다시 편지나 택배로 싸서 우체부에게 주고 그 우체부가 다시 나에게 전달하는 형식이다.

URL

예를들어

http://example.com:80/toys?p=foo#hello

이런 url이 있다고 해보자

1. http: Protocol - 어떤 방법으로 파일을 전해주고 수신받는지에 대한 약속이다. 프로토콜은 https, ftp, ssh, ssl등 여러가지 가있다.

2. example.com : Hostname - 해당 주소의 주인의 이름이라는 뜻이다. 원래 주소는 IP address를 쓰는데 127.0.0.1(localhost) 처럼 숫자로 되어있다 이것을 기억하기 어려우므로

DNS(Domain Name Service)를 통해 example.com이라고 이름을 붙여 표시하는 것이다.

3.  :80 : Port -  http 는 위에서 기본적으로 80번 포트를 사용한다고 했다. 80xx로 x에 두자리를 더 붙일수 있다 00~99까지 이다.

4. /toys : path - 요구사항있는 곳으로 가기위한 길

5. ?p=foo :Query parmeter  -요구 사항이 있는 곳에서 해당 정보가 있는 번호 (정보의 아이디 값이라고 생각하면 된다)

6. #Hello : fragment - 많은 분량의 페이지 정보중 해당 지점을 볼수 있도록 옮김 (서버로 요청하지 않고 브라우저 상에서 동작함)

그럼 사용자가

<http://example.com/foo/logo.png?p=1#tricky>

라고 요청을 보내면 http 가 처리하는 방식을 보면

request

GET /foo/logo.png?p=1 HTTP/1.1

1. GET : method - 정보를 전송할 때 사용하는 방식 이다 . POST, HEAD,PUT  등 여러 전송방식 있다.

2. /foo/logo.png?p=1 : 요청한 파일의 위치

3. HTTP/1.1 : http 1.1 버전의 프로토콜 사용

그럼 서버가 요청을 받고 다시 브라우저로 넘길 때 response 하는 방식을 보면

response

HTTP/1.1 200 OK

1. HTTP/1.1 : http 1.1 버전의 프로토콜 사용

2. 200 : statust code - 응답 상태 표시

3. OK : reason phrase - 응답 상태에 대한 이유

200 OK : 해당 파일 이 있어서 응답 양호

302 Found : 지금 주소에는 없지만 해당 주소 다른곳에 있다.

404 Not Found : 해당 파일이 존재 하지 않아 페이지를 로드 할 수 없다.(응답할 수 없다)

500 Server Error : 요청 중 서버가 고장났다. (서버측 문제)

<태그 이름 속성="값"></태그>

(모든 태그가 닫히는 태그가 있어야 하는 것은 아니다)

block 요소

* 줄 바꿈 속성 을 갖고 있음
* 기본 width 가 100%
* 블록 요소를 포함할 수 없지만 예외도 있다.(예외 , div , li, dd, address(p태그 만 가능) , ...등 -> 이 요소들은 블록 요소도 포함할 수 있다. )
* inline은 포함할 수 있다.

<h1></h1>~<h6></h6>

* 제목 태그
* 숫자가 작을 수록 글자의 크기가 크다

<p></p>

* paragraph를 만드는 태그

<ul></ul>

* <li>의 부모 태그
* 비순차 목록을 만드는 태그 -> 옆에 동그라미 같은 걸 만들어줌

<ol></ol>

* <li>의 부모 태그
* 순차 목록을 만드는 태그 -> 옆에 1. 2. 3. ... 같은 걸 만들어줌

<li></li>

* 목록을 만드는 태그

<div></div>

* 범용 그룹화 요소
* 기능이 없지만 기능이 없어서 정말 많이 쓰인다.

<form></form>

* 입력값을 서버로 전송해주는 태그
* 속성이 3가지가 있다.

속성 :

name = "내용" : 폼을 구분짓기위한 폼의 고유한 이름을 부여하는 속성

method = "내용" : 보낸 내용을 어떻게 보낼 것인지 전송 방식을 정해주는 속성( get , post, ...)

action ="주소" : 입력 값을 보낼 페이지를 정해주는 속성

Inline 요소

* 줄 바꿈 속성이 없다
* 내용만큼 width를 가짐
* width, height의 속성을 사용할 수 없음
* 블록요소를 포함할 수 없고 인라인 요소만 포함 가능

<br>

* 줄 바꿈 태그
* br태그 다음부터 줄이 바뀐다.

<b></b>, <em></em>, <strong></strong>

* 강조 태그
* 태그 안 내용이 진하게 표시된다.

<a></a>

* 링크 태그, 링크를 걸어주는 태그이다.
* 속성이 3가지가 있다.

href="주소" : 어디에 링크를 걸건지 주소를 입력하는 속성

target="속성" : 해당 주소를 어떻게 열 것인지 정하는 속성(예, \_blank -> 링크를 눌렀을 때 새로운 페이지에서 열린다.)

title="내용" : 태그 내용에 마우스를 올렸을 때 나오는 툴팁의 내용을 정하는 속성

<img>

* 사진을 넣을 때 쓰는 태그
* src 속성을 이용해 src="주소, 파일경로/이미지파일 이름' 형식으로 사진을 집어넣는다.
* alt 속성은 사진을 로드할 수 없을 때 나오는 내용을 정해주는 속성이다.

<span>

* 범용 인라인 요소 그룹화 태그
* 아무 기능이 없지만 인라인 요소이고 아무 기능이 없어서 정말 많이 쓰이는 태그

부모태그 <table><\table>

<table> : 표를 생성하는 태그

속성 :

width="가로 길이" : 표의 가로의 길이를 정하는 속성

border="선의 두께" : 선 두께를 정하는 속성

cellspacing = "셀 간격" : 셀과 셀의 간격을 정하는 속성

자식태그<tr><\tr>

<tr> : 행을 생성하는 태그

<td><\td>

<tr>로 생성된 행의 열을 생성한다

속성

rowspan="행 개수" : 지정한 행 개수 만큼 행 합치기

colspan="열 개수" : 지정한 열 개수 만큼 열 합치기

css 를 적용하는 방법은 3가지가 있다.

Inline : 태그 안에 style="내용;"으로 사용하는 방법 style 안에 내용은 css의 문법을 사용해야 한다.

Internal : head태그 안에 <style></style>태그를 넣고 스타일 태그안에 css를 작성하는 방법

External : <link href="css파일 주소">를 이용해 외부의 css파일을 불러와 적용시키는 방법

지금 11주차는 inline 방법으로만 알고 있으면 될 것같다.

태그 또한 수많은 태그가 있는데 w3schools를 검색해서 들어가거나 생활코딩에도 태그를 정리한 페이지가 있으니 거기에서 보면서 찾아서 하는 것도 도움이 된다.

부록 - 생활코딩

1일차

1.[프로젝트의 동기](https://opentutorials.org/course/3084/18438)

2.[기획](https://opentutorials.org/course/3084/18437)

3.[코딩과 HTML](https://opentutorials.org/course/3084/18445)

4.[HTML 코딩 실습 환경 준비](https://opentutorials.org/course/3084/18448)

5.[기본 문법 -태그](https://opentutorials.org/course/3084/18392)

<2018.04.02 변경 : atom 으로 c++도 가능해서 그냥 아톰으로 바꿨다.>

atom emmet 패키지 다운 받고 사용법도 익히는 걸 추천

본 내용은 이제 html파일을 만들어 브라우저에서 파일을 열어 확인하고 태그라는 문법을 배웠다.

<strong>내용</strong>

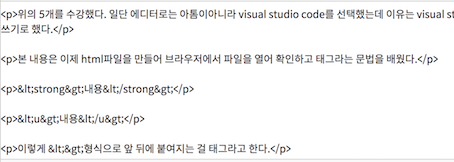
<u>내용</u>

이렇게 <>형식으로 앞 뒤에 붙여지는 걸 태그라고 한다.

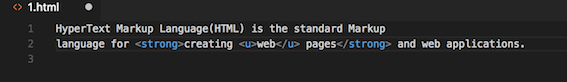
앞에 있는 태그를 start tag(열리는 태그)라 하고 뒤에 구별하기 위해 /를 붙인 것을 end tag(닫히는 태그)라 한다.

태그는 중복해서 쓸수있다는 점을 배웠다

소스를 눌러 확인해보니



이렇게 <P> </P>라는 태그가 보인다



https://khlug.org/image/4205abb8f0fc30c3.png

2일차

1.[혁명적인 변화](https://opentutorials.org/course/3084/18400)

2.[통계에 기반한 학습](https://opentutorials.org/course/3084/18452)

3.[줄바꿈](https://opentutorials.org/course/3084/18403)

4.[HTML의 중요성](https://opentutorials.org/course/3084/18488)

5.[최후의 문법 속성과 img](https://opentutorials.org/course/3084/18407)

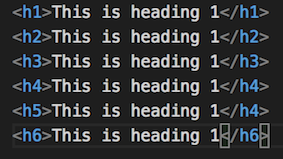
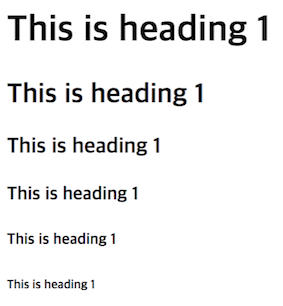
6.[부모자식과 목록](https://opentutorials.org/course/3084/18408)

1.[혁명적인 변화](https://opentutorials.org/course/3084/18400)

<h1>태그를 배웠다

정리.

* h는 1부터 6까지 쓸수있고 그 이상은 안 된다.
* h태그는 제목을 나타내는 태그 그냥 쓰는 글자보다 조금 더 두껍다.
* 1에서 6으로 갈수록 글자의 크기가 작아진다.

2.[통계에 기반한 학습](https://opentutorials.org/course/3084/18452)

html을 꾸밀 때 어떤 태그들이 많이 쓰이는지 통계를 낸 것을 보았다. 한 웹페이지를 만들 때 보통 25~26개 정도의 태그가 쓰인다는 것을 알았고

또한 그렇게 생각해보니 생각보다 공부 할 것이 많이 없다는 것을 알려주었다. 정말 자주쓰이는 것만 알고 나머지는 검색을 통해 알아간다 -> 정말 좋은 공부법이다!

이 통계에 기반한 학습은 정말 프로그래밍 공부를 떠나서 나에게 이제 공부를 어떻게 해야하는지 알려주고 나도 할 수 있다는 자신감을 얻게해주는 것 같다.

3.[줄바꿈](https://opentutorials.org/course/3084/18403)

<br>태그, <p><\p>태그

< br> 태그는 줄바꿈(new line)태그인데  다른것 과 다르게 닫히는 태그가 없다

닫히는 태그가 없는 태그는 무언가를 설명하지 않는 태그

예로는 img, input, br, hr, meta가 있다고 한다.

<br>을 여러번 입력해서 단락을 구분 할 수있지만 소스를 볼 때 단락 구별은 <p>태그가 더 보기 좋은 것 같다.

<br>태그는 위치, 크기, 여백이 제어가 안된다.

<p><\p>태그는 단락, 문단 태그(paragraph)태그인데 페이지에서 단락을 표시할 때 사용하는 태그이다. 한 단락을 지정해서 시작할 때 <p>로 열고  끝날 때 <\p>로 닫아주면 된다.

CSS를 배우면 p 태그는 위치나 크기 여백 같은 것을 완전히 자유롭게 제어할 수 있다.

 속성 중 style이 있는데 style이라는 것으로 위에 언급한 위치,크기, 여백을 조절할 수 있는 것 같다.

4.[HTML의 중요성](https://opentutorials.org/course/3084/18488)

웹의 접근성

이미지는 존재하지 않는 정보

검색엔진에 의존을 많이 하는 현대 사회에서

검색되어지지 않는 정보 -> 웹페이지 검색에서 우선순위가 많이 밀리는 정보-> 나에게는 없는 정보

이런 측면들을 보면 HTML은 정말 중요한 부분이다. 태그를 의미에 맞게 쓴다는 것이 정말 중요함을 배웠다.

5.[최후의 문법 속성과 img](https://opentutorials.org/course/3084/18407)

사진을 넣고 싶으면 img태그를 쓰는데

<img src="브라우저 주소">로 웹브라우저에 있는 주소로 사진을 불러올 수 있고

해당 html파일 과 같은 폴더에 이미지가 있다면

<img src="이미자 파일 이름">만으로 사진을 불러올 수있다.

width라는 속성도 있는데

width="숫자 or %"로  크기도 조정 가능하다

여기서 나온 속성은 style,src, width이 있다.

6.[부모자식과 목록](https://opentutorials.org/course/3084/18408)

태그에는 부모 자식 처럼 계속 따라 붙어다니는 태그들이 있다. 자식이 있으면 부모가 있고 부모가 있으면 자식이 항상 있는 것 처럼 어떤 태그들은 연관된 태그가 항상 따라붙어서 쓰인다는 것이다.

어떤 태그는 2대인 것도 있고 어떤 태그는 3대인 것도 있다.

예로 나온 것이 <li>와 <table>이다 목록을 만들기 앞서 리스트<목록>를 만들어주는 태그 <li>를 보자.

* 목록

부모 태그<ul><\ul>(unordered tag), <ol><\ol>(ordered tag)

<ul>(unordered tag) : 무질서하게, 정렬하지 않고 목록을 만들어주는 것 같다.

<ol>(ordered tag): <ol>이 붙으면 목록 앞에 넘버링이 되고 목차를 지우고 생성해도 넘버가 자동으로 매겨진다.

Tip : 목록을 구별할 때 쓰는 경계 개념

자식 태그<li><\li>

<li> : 목차 앞에 <li>를 붙여져 자동으로 줄 바꿈이 되고 목차의 형태로 만들어준다.

* 표

부모태그 <table><\table>

<table> : 표를 생성하는 태그

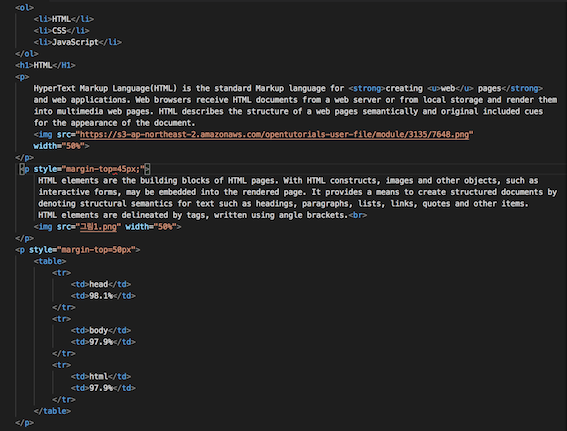
자식태그<tr><\tr>

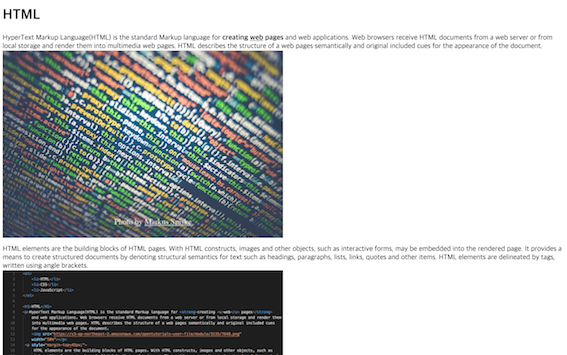
<tr> : 행을 생성하는 태그

자식의 자식(?)태그 <td><\td>

<td> : <tr>로 생성된 행의 열을 생성한다.

Tip :이런 부모자식태그를 쓸 때 보기 복잡하므로 파이썬에서 들여쓰기를 하는 것처럼 tab으로 내용을 들여서 써 보기 좋게 만들어준다.





3일차

1. [문서의 구조와 슈퍼스타들](https://opentutorials.org/course/3084/18409)
2. [HTML태그의 제왕](https://opentutorials.org/course/3084/18418)
3. [웹사이트 완성](https://opentutorials.org/course/3084/18431)
4. [원시웹](https://opentutorials.org/course/3084/18889)
5. [인터넷을 여는 열쇠 : 서버와 클라이언트](https://opentutorials.org/course/3084/18890)
6. [웹호스팅(github pages)](https://opentutorials.org/course/3084/18891)

1. [문서의 구조와 슈퍼스타들](https://opentutorials.org/course/3084/18409)

* <title></title>   웹페이지의 제목을 설정하는 태그

* utf-8 :(컴퓨터는 0과 1로만 저장하는데 그 0과 1을 어떻게 저장하는지에 대한 방법을 약속해놓은 형식 중의 하나)한글을 사용할 수 있도록 하는 저장 형식.

현재 내가 쓰는 아톰은 utf-8형식으로 저장해서 한글을 사용 할 수 있는데 웹페이지는 그렇지 않으므로 웹페이지도 utf-8형식으로 열어야지 한글을 볼 수 있는 것이다.

따라서 <meta charset="형식"> 으로 웹페이지가 내가 지정한 형식으로 페이지를 열게 해준다.

웹페이지에서 한글을 볼려고 하므로 <meta charset="utf-8">을 한다.

* <meta charset="형식"> : 웹페이지가 내가 지정한 형식으로 페이지를 열게 해주는 태그(char : character(문자) set :규칙)

* <body></body> :  본문이라는 것을 알려주는 태그(하나의 약속)

* <head></head>  : 본문을 설명하는 태그(하나의 약속)

* <html> : head와 body를 모두 포함하는 태그

                         맨 위에 처음으로 쓰고 그 위에 관용적으로 <!doctype html>이라고 이 문서가 html형식이라는 것을 나타낸다.

2.[HTML태그의 제왕](https://opentutorials.org/course/3084/18418)

* <a></a> : 링크를 거는 태그

속성 href="주소" -> hypertext reference -> 이 주소의 하이퍼텍스트를 참고해라라는 뜻

속성 target="\_blank" -> 빈 페이지에 링크를 연다

속성 title="내용" -> 링크의 툴팁의 내용을 작성한다.

3. [웹사이트 완성](https://opentutorials.org/course/3084/18431) -> 첨부 파일

4.[원시웹](https://opentutorials.org/course/3084/18889)

info.cern.ch 최초의 웹사이트라고 하는 곳이다 소스코드를 보니 내가 지금 껏 배운 것들이 대부분이다.

이제 내가 만든 이 웹페이지를 인터넷 상에서 공유를 할 수 있도록 웹 서버를 알아야 한다.

5.[인터넷을 여는 열쇠 : 서버와 클라이언트](https://opentutorials.org/course/3084/18890)

주소를 쓰고 그 주소에 대한 정보를 요구하는 곳 : 클라이언트

정보를 요구받으면 그 정보를 응답하는 곳  : 서버

두가지 방법

1. 웹서버를 내 컴퓨터에 설치하는 것

2.웹호스팅 업체를 이용하는 것

6.[웹호스팅(github pages)](https://opentutorials.org/course/3084/18891)

[github의 pages기능](https://github.com/)(https://github.com/)

무료 웹 호스팅 업체이다

나는 html만으로 웹을 만들었기 때문에 서버에서 따로 해줄 것이 없다고 한다.

이런 특성을 정적(static)이라고 하고 이 정적인 특성만을 가지면 무료로 웹호스팅을 해주는 업체가 많다

동적(dynamic)인 것들도 있는데  이런 것들은 대부분 유료 로 이용해야 한다고 한다.

<https://wjdwls0630.github.io/WEB1/>. github에 내가 지금 까지 만든 페이지를 생성했다

4일차

1. [웹서버 운영하기](https://opentutorials.org/course/3084/18892)
2. [웹서버 운영하기 :맥](https://opentutorials.org/course/3084/18894) -> (1) 맥에 웹서버 설치하기, (2) 웹서버와 http, (3)웹서버와 웹브라우저의 통신
3. [수업을 마치며 1](https://opentutorials.org/course/3084/18896)
4. [수업을 마치며 2](https://opentutorials.org/course/3084/18897)
5. [수업을 마치며 3](https://opentutorials.org/course/3084/18898)
6. [부록 : 코드의 힘](https://opentutorials.org/course/3084/18598) -> (1)[부록 : 코드의 힘-동영상 삽입](https://opentutorials.org/course/3084/18453) ,  (2)[부록 : 코드의 힘- 댓글 기능 추가](https://opentutorials.org/course/3084/18594), (3)[부록 : 코드의 힘 - 채팅 기능 추가](https://opentutorials.org/course/3084/18597), (4)[부록 : 코드의 힘 - 방문자 분석기](https://opentutorials.org/course/3084/18899)

1.[웹서버 운영하기](https://opentutorials.org/course/3084/18892)

아파치(Apache)를 이용해보기.(오프소스) 옛날에 아파치 톰캣? 이라는 것을 클릭해본적 있던 기억이 있는데 그 때 생겼던 궁금증이 풀릴 것 같다.

나는 맥북을 이용하므로 맥에 웹서버 설치하기 라는 동영상을 들었다.

2.[웹서버 운영하기 :맥](https://opentutorials.org/course/3084/18894)

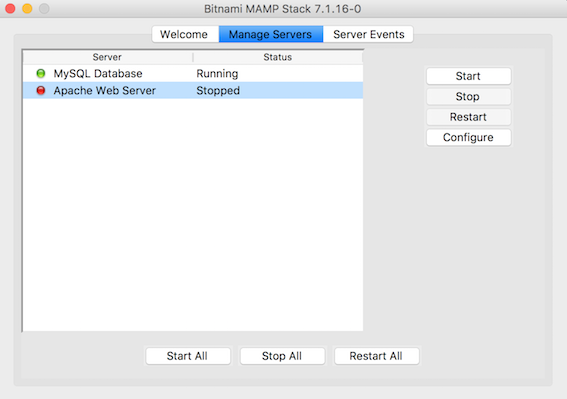
(1) 맥에 웹서버 설치하기

bitnami mamp :  자동으로 아파치가 설정할 필요없이 설치가 되게 해줌 MAMP는 M(Mac) A(Apache) M(MySQL) P(PHP) 의 약자이다.

따라서 bitnami MAMP를 깔면  Apache, MySQL,PHP가 깔린다.

다 깔고나면 management console 프로그램이 실행이 되고 이 것은 아파치 웹서버를 관리하는 도구라고 한다.

console프로그램에 manage server라는 탭에 들어가면



이렇게 뜨는데 보이는 것과 같이 Apache Web Server라는 웹서버를 조작할 수있다

이  console프로그램의 파일명은 manager-osx 다. 스포트라이트, 아니면 Application 파일에서 찾을 수있다.

(2) 웹서버와 http

이 아파치 웹서버로 만든 페이지를 들어가보면 http://127.0.0.1:8080/index.html이라고 주소가 뜨는데

index.html은 MAMP디렉터리 안에 apache2 안에 htdocs라는 폴더안에 존재한다.

127.0.0.1 -> IP Address(Internet Protocol Address) 이다.

8080 -> Port

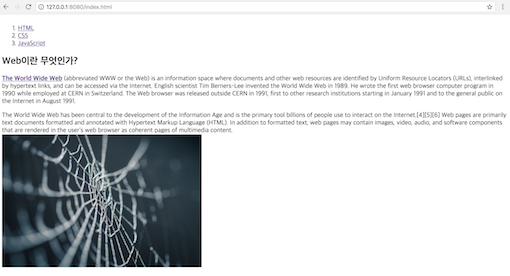
[http://](NULL) -> Hyper Text Transfer Protocol

결론은 웹브라우저가 웹서버에 저장되어있는 어떤 파일을 요청을 할 때 해당 웹서버를 부를 주소값 (127.0.0.1)이 필요하고

port는 내 컴퓨터에 맥에 기본으로 저장된 웹서버 말고 아파치를 깔았으므로 서로 다른 웹서버임을 구별하기 위해 어떤 통로로 들어가겠다 라는 것이라고 한다.

그래서 해당 웹서버를 부를 주소값을 입력하고 그 다음에 열어볼 파일을 입력하면 내가 만든 웹서버에서 그 파일을 찾고 그걸 웹브라우저에게 표시해주는 것이다.

그럼 내가 이제 apache2파일에 들어있는 htdocs파일에 내가 만든 웹 코드를 집어넣으면 http://127.0.0.1:8080/index.html에 내가 만든 index.html이 읽히면서 웹브라우저에 표시될 것이다.



http://127.0.0.1:8080/index.html에 내가 만든 html이 열리는 것을 확인 할 수 있다.

(3)웹서버와 웹브라우저의 통신

하지만 내 컴퓨터에 웹브라우저, 웹서버를 설치해서 이렇게 보는 것은 의미가 없다. 내 웹서버에 내 컴퓨터가 접속하는 것이므로 다른 컴퓨터로 내 웹서버에 접속하게 끔 해보는 실습을 한다.

다른 컴퓨터 말고 스마트폰을 이용해보자

일단 웹브라우저가 웹서버에 접속하기 위해서는 웹서버가 설치된 컴퓨터 주소(IP Address)를 알아야하는데 내 것은 이미 알고있으므로 그 주소를 이용해 위 주소 처럼 접속해보자.

그 전에 위의 ip주소를 보면 127.0.0.1인데 저 주소는 내 컴퓨터가 자신을 가르키는 대명사라고 생각하라고 한다.

그리고 컴퓨터에 나오는 내 ip주소는 내 컴퓨터를 나타내는 고유명사라고 생각하라고 영상에서 말한다.

여튼 그러면 스마트 폰을 이용해 접속하려면 http://내 IP주소:8080/index.html을 입력해야된다.

입력하기 전에 내 웹서버가 설치된 컴퓨터(노트북) 과 웹브라우저가 설치된 컴퓨터(스마트폰)이 같은 네트워크에 접속해있어야 되므로 같은 wifi에 연결을 했다

이렇게 설정을 모두 한 후 핸드폰  사파리로 접속해보니 정말로 내가 만든 웹페이지가 스마트폰에 나오는 것을 확인할 수 있었다.

3.[수업을 마치며 1](https://opentutorials.org/course/3084/18896)

4.[수업을 마치며 2](https://opentutorials.org/course/3084/18897)

5.[수업을 마치며 3](https://opentutorials.org/course/3084/18898)

지금 내가 공부하는 이유를 설명을 잘 해주시는 것 같다.

다음 진도가 CSS로 알고있는데  CSS는 웹페이지를 꾸미는 말 그대로 디자인 해주는 언어라고 한다.

생활코딩 영상만 보고있으면 나도 금방 할 수있을거라는 자신감이 생겨서 좋다. 빠르게 CSS를 공부해봐야겠다.

6.[부록 : 코드의 힘](https://opentutorials.org/course/3084/18598)

(1)[부록 : 코드의 힘-동영상 삽입](https://opentutorials.org/course/3084/18453)

* <iframe></iframe>  : 동영상을 삽입할 수있는 태그

속성 :

 width="크기"   -> 가로 의 크기를 정하는 속성

height="크기"  -> 세로의 크기를 정하는 속성

src="주소, 파일 명" -> 동영상을 embeded 하는 주소 , 컴퓨터에 있는 파일이면 파일명(단 , 해당 html과 같은 디렉터리 안에 있어야 한다.)

frameborder="0 or 1" -> 경계선 표시 유무 , 0이면 없고 1이면 경계선이 있다

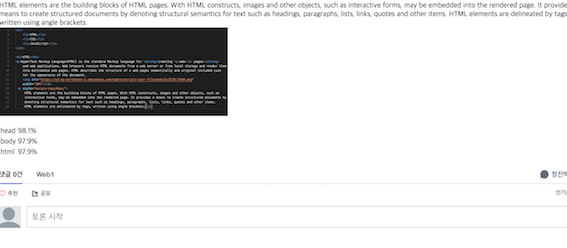
allowfullscreen -> 동영상의 전체화면 기능을 사용할 수 있게 해준다. (사용 못하게 하려면 해당 코드를 지우면 됨)

(2)[부록 : 코드의 힘- 댓글 기능 추가](https://opentutorials.org/course/3084/18594)

댓글 기능을 다는 것은 굉장한 기술력을 요한다. 스팸 차단 기능 등 백엔드 기능이 필요하므로 지금 내가 당장 댓글 기능을 쓸 수는 없다.

그래서 다른 이가 만든 댓글 기능을 가져다 써보는 방법을 배웠다 DISQUS 와 LiveRE라는 사이트에 접속을 해서 댓글 기능을 추가해보자.

DISQUS 사이트를 이용했다. 파일로 열면 댓글 기능이 보이지 않는데 웹서버를 통하거나 github에 만들었던 내 웹페이지에 파일을 수정해서 올리면



이렇게 댓글 기능이 달리는 걸 확인 할 수있다

(3)[부록 : 코드의 힘 - 채팅 기능 추가](https://opentutorials.org/course/3084/18597)

 www.tawk.to라는 사이트를 이용한다. 무료로 채팅 기능 소스코드를 제공해주는 사이트이다.

이것도 마찬가지로 무료 소스로 간단하게 이런 채팅기능을 내가 만든 사이트에 추가할 수있고 또한 tawk 홈페이지에서 관리자로 간단하게 사이트 별로 관리가 가능하다.

앞으로 무언가 기능을 추가하고 싶으면 이렇게 구글링 을 통해 찾아보는 습관을 들여야겠다

(4)[부록 : 코드의 힘 - 방문자 분석기](https://opentutorials.org/course/3084/18899)

이번에도 오픈소스를 이용해 기능을 추가한다.

google analytics라는 것 인데  <https://www.google.com/analytics/> 해당 주소로 들어가면 내가 만든 웹페이지의 로그 정보를 관리할 수있다

구글링을 통해 검색하면 정말 다양한 기능의 오픈소스들이 있는 것 같다 . 실습을 진행하면서 내가 필요한 기능들을 잘 찾아보고 써야겠다.

<https://wjdwls0630.github.io/WEB1/index.html> -> 내가 지금까지 만든 사이트이다