**20150228 (토)**

1. c\

2.

**20150226 (목)**

1. css 1픽셀이 레티나 디스플레이 (아이폰 5)에 이르러서는

css 1픽셀 = 4개의 물리 픽셀

물리 픽셀 4 = 논리 픽셀 1

- 아이폰 4는 640\*960 (물리 픽셀), 실제로 css는 320\*480

1. 공통 스타일?

3. 선택자를 좀더 구체적으로 지정했을 때 우선순위가 높아진다.

예: #main > main

그래서 나중의 추가 작업 등을 고려해서 낮은 레벨의 main 을 쓰는 것도…

그냥 🡪 클래스 🡪 아이디… 등의 순서로 선택자 우선 순위가 쓰여진다

4.

box-shadow: 0 0 4px #999;

box-shadow: 0 0 4px rgba(0,0,0,0.3);

ie7, 8은 첫 번째 칸만을 인식하고

그 상위 버전은 아래 칸마저도 인식하면서 더 정교하게 디스플레이한다.

5. 소숫점 앞에 0은 생략이 가능하다.

(0,0,0,.3)

이런 형태로…

6. height 보다 min-height는 최소한 그 이상이 될 때 자동으로 커지는 기능을 한다.

7. 정병찬 선생 코드 관련

<http://github.com/dece24kr/fecamp2/tree/gh-pages/html>

<http://dece24kr.github.io/fecamp2/html>

8. 수직마진 중첩 현상!

9. lnb a{

이런 빈 칸은 자손까지

lnb>a 에서 꺽쇠는 자식 선택자를 의미. 그러면 당연히 a를 인식하지 못한다.

10. mobile에서 손가락이 클릭할 수 있는 크기가 32픽셀! 너비와 높이가 최소한 32픽셀이 되어야 모바일에서 클릭할 수 있다.

11. hover는 마우스를 올리면

focus:는 키보드

12. 부모 자식 간에 수직 마진이 중첩이 될 경우에는…

border 또는 padding을 줄 경우에는 수직 마진의 중첩 현상이 생기지 않는다.

13. WCAG 2 AA Compliant

snook.ca 사이트에서 compliant를 예스로 받아야 웹 접근성이 확보되는 거다.

**20150214 (토)**

1. inline과 block의 동일한 css가 선택되기 위해서 한 칸 띄운다.

2. inline요소에는 width와 height가 적용되지 않는다. Block과 차이점 중에 하나임.

3. inline요소는 수직 라인만 적용되지 않는다.

4. inline-block은 inline의 특성을 가지고 있어서 하나의 글자처럼 인식한다. 그래서 글자끼리 공백이 약간 있듯이 하나하나가 마치 자간을 가지고 있는 특징 때문에 이런 현상이 발생한다. 보통은 4픽셀 정도 생기는데, 그게 일정하지 않다. 자간은 폰트 사이즈에 영향을 받기에 일정하지 않는거다.

5. inline에서만 line-height가 적용된다.

6. class는 여러 가지 밸류 값을 가질 수 있는데, 띄어쓰기 공간이 중복을 의미한다.

7. position이 static이 자기 뜻대로 왼쪽, 오른쪽 등을 가질 수 없다.

8. 오프셋(기준점)은 position relative는 탑과 레프트가 기준점이다.

9. absolute의 오프셋은 부모 요소인데, 포지션이 static이 아닌 요소를 오프셋 왼쪽/윗쪽 8픽셀

10. fixed는 뷰 포트를 기준으로 오프셋으로 보이기에 그 자리에…

11. float right, left, none 이렇게 3개가 있다.

12. float는 일종의 z 축으로 뜨는 현상에 비유할 수 있다.

13. clear는 float의 속성을 해제한다는 뜻.

14. 부모와 자식 간의 탑 마진도 겹친다. 부모와 자식 간에도 큰 놈의 마진이 작은 마진을 이긴다.

15. ie8 이상의 브라우저를 지원하는 쪽으로 이번 수업시간 진행할 것.

16. https에서 s는 security. 암호화나 중요한 금융 거래에는 붙는다.

17. 태블릿 사이즈 🡪 아이패드 세로 768픽셀

18.

@media (min-width: 768px){

body{

background: red;

}

@media all and (min-width: 768px){

body{

background: red;

}

가 기본적으로 생략.

19. 모바일 코드에서 미디어 쿼리를 작성하지 않는 것은 미디어쿼리를 사용하지 않는 모바일 기기도 많기 때문에. 미디어 쿼리는 css3 버전임.

🡪 Mobile First CSS 기법.

20. text-shadow: x y blur color

**20150212 (목)**

1. 인라인 엘리먼트에는 수직 패딩을 쓰지 않는다. 인라인 요소에는 패딩을 적용해도 의미가 없다. 좌우 패딩은 유효하지만 (겹치지 않는다).

span{

display:block;

border: 10px solid #000;

padding: 10px;

margin: 10px;}

2. 수직 마진 중첩 현상: ???

- 20픽셀을 주려면 어떻게 해야하지?

3. P엘리먼트 하나 쓰면…

1만큼 탑 마진과 바텀 마진 각각 갖고 있다. 그래서 밀어내고, 침범하지 못한다.

그래서, P ELEMENT를 또 쓰면

그 두개의 P 엘리먼트의 사이가 2만큼의 수직 마진이 생긴다. 그래서 CSS는 문장 하나 1만큼의 공간을 주기위해 1 만큼의 수직 마진만 주는 걸로 정해버린다.

CSS의 스펙: 탑 마진과 바텀 마진이 겹쳤을 때 수직 마진을 겹치게끔 한다.

4. 절대값이 큰 수직 마진이 살아남고, 작은 마진은 죽는다. 버려진다.

수직 마진이 중첩될 때…

5. +와 –가 만났을 때는 값을 계산해버리는 두 박스 사이의 결과가 결정이 된다.

6. –와 –가 만났을 때는…

🡪 디스플레이와 마진 속성..

다음은 plot에 대한 속성 연구…

**20150211 정찬명 강사의 FE.CAMP] 첫째주 강의 요점 정리.   
// 페이스북이 접근성이 좋아서 이곳에 작성합니다.**

**# SW 설치  
1. Chrome 브라우저 설치. + HTML5 outliner( [http://goo.gl/VFCPuW](http://l.facebook.com/l.php?u=http%3A%2F%2Fgoo.gl%2FVFCPuW&h=QAQFsrenv&enc=AZOdKs_oupqhTa2CRUObnMStOckRSaCM_omkKg7X2hwMFmOLHXJ7QSISBne4TTYoy4-LPaeWIM3ltEMvpmRJPHbyp4-ujXUQk6aNs50Ne8Z8pVVKpj4V1w9r9afWTmFgMQ1y8kojX0hLC7J5LNmDZ3q6&s=1" \t "_blank) ) 확장기능 설치.   
2. SublimeText3 설치.  
3. github client 설치.(윈도우: [https://windows.github.com/](https://www.facebook.com/l.php?u=https%3A%2F%2Fwindows.github.com%2F&h=dAQEha87r&enc=AZNRfRqZ_QOqtYjbD2ZF9BOny8cguX4F0UKMfhcSVYaCCVC2xFtz5fDW5YX6lgHrcxv44CSU1PO1ELZj-LCwF6MIEF725SJ-wNNu10HDthk3imAMZHlaM7M7JCkKie8534bQxh3hkCegTJtHVtmL_BIz&s=1" \t "_blank) 맥:[https://mac.github.com/](https://www.facebook.com/l.php?u=https%3A%2F%2Fmac.github.com%2F&h=eAQE3pBmL&enc=AZNA_rf-hn61Zl3I1NLMuhxsO_HSbOrBAHJ94eQKCaA1XN52yP7U7v0jXRhDh0Dai2q-G9iCUt6n1dZCIdepomDav6Ja7pIqx9joPuQatcWUbp3rvbf8B_s2IJbZ-44KijC59ScOJkNOAgqHipvokgBB&s=1" \t "_blank))**

**# 저장소 설정  
1. [http://github.com/](http://l.facebook.com/l.php?u=http%3A%2F%2Fgithub.com%2F&h=IAQGJbLKv&enc=AZP5DhEJ2w5Jh234Musqv9XbG-DV3Zwq8-YASzyTcLtdAruJDsabQVFpxK-kM7D-LgOCQB7bcJksmOEPC-uTqae1pAQOJpofK2Xai7nieWLY_Ou6rVcDxKcuDEUCWDMxavdCXXow5LJjlBXysQNbywTh&s=1" \t "_blank) 사이트에 가입하면 웹 페이지 호스팅 및 도메인([http://userid.github.io/repository/](http://l.facebook.com/l.php?u=http%3A%2F%2Fuserid.github.io%2Frepository%2F&h=cAQExIBxA&enc=AZNjSzJTY3mJB0ZrG8tYjsKq3VQYiAUq10V_njdvqAY8xO4kMiC8XmCj25klDeQamUGweLZNyoPdxBsvPthjMLpkPF9-wBI3DyEtLq44sQdokIsCfnyEfjXe6cpzQnywCyJKLNAo-1XV0KSvzLSCic3k&s=1" \t "_blank)) 서비스를 무료로 이용할 수 있음. 최초 회원 가입 후 반드시 이메일 인증을 진행해야 제한 없이 서비스를 이용할 수 있음.  
2. /fecamp2/ 라는 이름의 저장소를 생성한 다음 깃허브 클라이언트에서 원격 저장소를 로컬에 복사(clone).  
3. /fecamp2/ 저장소의 master 브랜치 이름을 gh-pages 으로 변경하면[http://userid.github.io/fecamp2/](http://l.facebook.com/l.php?u=http%3A%2F%2Fuserid.github.io%2Ffecamp2%2F&h=fAQEW9LKT&enc=AZNlQTighcvsWDTN_IDuxNp8VSw_snUsj2NttohK2iZAaZqLazJcHB3PGjnrNYXEftUDsRB4klcFmRcds0iPkSbahlm5VNHHz2Jwrx2DKOMOKAQxl9AS_q-Mh-XdUKsw092skJyfzAwWyWnPuf830lGr&s=1" \t "_blank) 도메인을 사용할 수 있다.  
4. /fecamp2/ 로컬 저장소에 /html/ 디렉토리를 생성하고 /html/ 디렉토리에 index.html 파일을 생성 index.html 파일에 "Hello World!" 라는 문자열을 작성한 다음 원격 저장소에 push(commit & sync)한다. commit은 로컬 저장소에 스냅샷(변경 사항에 대한 기록)을 저장하는 것이고, sync는 로컬 저장소의 파일과 스냅샷을 서버에 업로드하여 동기화 하는 것이다.**

**# 크롬 브라우저 사용  
1. Ctrl+(+), Ctrl+(-), Ctrl+(0) 키를 이용하여 화면을 확대/축소/원래대로 할 수 있다. 시각 장애인(약시)들이 이런 기능을 사용한다. 화면을 크게 보여주기 위해 강사도 이런 기능을 쓴다. 참고로 알아두자.  
2. 마우스 오른 버튼을 누르고 HTML 코드에 대한 요소 검사를 할 수 있다. 또는 Ctrl+Shift+C 단축키를 사용한 다음 확인하고 싶은 영역을 클릭해도 된다. 코드를 확인할 수 있는 부분을 인스펙터(inspector)라고 부른다.**

**# SublimeText3 편집기 사용법  
1. HTML 태그 이름을 작성하고 Tab 키를 누르면 열고 닫는 태그가 자동으로 완성된다.  
2. Ctrl+Shift+D 단축키를 사용하면 현재 라인을 복사하여 다음 라인에 붙여넣기 된다.  
3. Ctrl+Space 단축키를 사용하면 HTML 속성과 값 또는 CSS 속성과 값을 자동 완성할 수 있는 기능이 활성화 된다.  
4. Ctrl+/ 단축키를 사용하면 HTML 또는 CSS 코드의 주석을 작성할 수 있다. 이것은 JavaScript 코드에서도 마찬가지.**

**# HTML 실습  
1. DTD(Document Type Definition) <!DOCTYPE html> 를 작성하지 않으면 웹 브라우저는 쿽스 모드(quirks mode) 상태(Internet Explorer 5.5 버전의 렌더링을 흉내내는 모드)로 웹 페이지를 표시한다. 브라우저가 쿽스모드로 동작할 때 웹 페이지는 표준에 따르지 않는 방식으로 렌더링 되고 브라우저마다 서로 다른 모습으로 깨져 보이게 된다. 그러므로 DTD는 반드시 선언하고 DTD 앞에는 줄 바꿈이나 공백을 두지 않아야 한다.  
2. 뷰포트(viewport)는 웹 브라우저에서 내용을 표시하기 위한 영역을 말한다. 데스크톱 브라우저와 모바일 브라우저는 뷰포트 개념이 서로 다르다. 데스크톱 브라우저는 화면에 보이는 영역이 뷰포트가 되지만 모바일 브라우저는 브라우저의 크기(스마트폰 단말의 크기)와 관계없이 웹 페이지의 실제 크기가 뷰포트가 된다. 풍경을 담는 사진에 비유한다면 데스크톱은 카메라의 사각형 앵글이 뷰포트가 되고 모바일 단말은 앵글과 관계 없이 실제 세상에 존재하는 풍경 자체가 뷰포트가 되는 것이다. meta 요소를 사용하면 모바일의 뷰 포트를 데스크톱과 동일하게 변경할 수 있다. <meta name="viewport" content="width=device-width">  
3. <html lang="ko"> 웹 페이지에 주로 사용된 언어(human language)가 무엇인지 선언하여 검색엔진 최적화 및 웹 접근성에 대응한다.  
4. <meta charset="utf-8"> 현재 페이지에 사용된 코드와 모든 문자열의 문자셋을 선언한다. 문자셋 값을 "utf-8" 으로 선언해야 지구상의 모든 문자열을 오류 없이 표현할 수 있다.  
5. HTML 요소의 기본 골격은 <html><head></head><body></body></html> 이렇다. 이 요소들은 한 페이지에 한 번만 선언할 수 있고 선언 위치를 바꿀 수 없다.  
6. HTML 태그(<요소 속성="값"></요소>)의 문법 형식은 요소(element), 속성(attribute), 값(value). CSS의 문법 형식은(선택자{속성:값}) 선택자(selector), 속성(property), 값(value)으로 구성되어 있다.  
7. <div> 요소는 의미는 없고 용도만 있는 태그이다. 콘텐츠 블럭을 의미 없이(또는 스타일을 표현하기 위해) 감쌀 필요가 있을 때 사용한다. <span> 요소 또한 의미는 없고 용도만 있는 태그이다. 인라인 텍스트를 의미 없이(스타일 때문에) 감쌀 필요가 있을 때 사용한다.  
8. <header>, <footer> 요소는 페이지 전체 또는 특정 내용의 도입부와 맺음말 부분에 사용하는 의미를 가진 태그. <main> 요소는 한 페이지에 한 번만 사용할 수 있으며 콘텐츠의 본문 영역으로써 다른 페이지에서 반복되는 내용은 수록하지 않도록 한다. <nav> 요소는 사이트의 주된 탐색 링크 섹션을 의미한다. <aside> 요소는 부수적인 섹션을 의미한다.  
9. <section>, <article>, <nav>, <aside> 요소는 섹션을 만드는 요소다. 섹션을 만드는 요소에는 제목 <h2> 요소를 사용할 것을 강력하게 권장한다.  
10. <ul> unordered list, <ol> ordered list, <li> list item.  
11. 모든 HTML 요소는 블록 요소이거나 인라인 요소이다. 블록 요소는 콘텐츠 덩어리를 담는 요소이고, 인라인 요소는 텍스트(문자열)를 담는 요소이다. 블록 요소는 user agent stylesheet 에 의해 {display:block} 이라고 선언되어 있다. 인라인 요소에는 그런 선언이 없다. 블록 요소는 인라인 요소를 자식으로 포함할 수 있으나 인라인 요소는 블록 요소를 자식으로 포함할 수 없다. 예를 들면 <div><span></span></div> 이 코드는 유효하지만 <span><div></div></span> 이 코드는 인라인 요소가 블록 요소를 감싸고 있으므로 유효하지 않다. 예외적으로 <a> 요소는 인라인 요소임에도 불구하고 블록 요소를 자식으로 두고 감쌀 수 있다.  
12. HTML 주석은 <!-- --> 이렇게 생겼고, CSS 주석은 /\* \*/ 이렇게 생겼다. SublimeText3 에서 Ctrl+/ 단축키로 쉽게 작성할 수 있다**

**20150211 자료**

|  |
| --- |
|  |
|  | <!doctype html>  <!-- |
|  | html을 가장 최신버전으로 사용하겠다는 의미로 작성한다. |
|  | 이를 작성하지 않을 경우 가장 구버전의 html을 사용한다는 의미가 된다. |
|  | doctype은 document type의 약자이다. |
|  | 모든 html 문서의 가장 첫번째 줄에 작성하여야 한다. |
|  | --> |
|  | <!-- |
|  | 모르는 것이 있다면 apes0123@gmail.com이나 각 선생님들에게 물어보면 된다. |
|  | 라인으로 물어봐도 친절하게 답해주니 대략적인 스터디에 참고하도록 하자. |
|  | --> |
|  | <!-- |
|  | 꺽쇠(<)와 느낌표(!), 그리고 두개의 대시(-)(-)에 담기는 내용을 주석이라고 부른다. |
|  | 주석은 여러줄을 포함할 수 있으며, 주석의 끝은 두개의 대시(-)(-)와 하나의 닫는 꺽쇠(>)로 구성한다. |
|  | 코드 상에 존재하고 소스보기에서는 볼 수 있지만, 유저는 이를 볼 수 없다. |
|  | 코드를 개발하는 사람에게 설명을 해주기 위해 많이 사용한다. |
|  | --> |
|  | <html> |
|  | <!-- 모든 html 문서의 기본 요소이다. html문서가 온전한 html문서임을 의미한다. |
|  | 생략이 가능하나 생략을 권고하지는 않는다. |
|  | --> |
|  | <head> |
|  | <!-- 들여쓰기는 키보드의 Tab키로 사용할 수 있다. 혹은 스페이스바를 4번 눌러도 동일한 효과를 얻을 수 있다 --> |
|  | <!-- 들여쓰기란 코드를 안쪽으로 밀어넣는걸 의미한다 --> |
|  | <!-- 이 코드에서 head가 html의 여는 태그(<html>)와 닫는 태그(</html>) 사이에 있다. 이는 head가 html의 자식임을 의미한다 --> |
|  | <!-- head는 컴퓨터에게 전달할 정보를 담는 html 요소이다 --> |
|  | <meta charset='UTF-8'> |
|  | <!-- 웹사이트에서 한글, 일본어 등 비영어권 언어가 깨지지 않도록 하는 코드이다 --> |
|  | <!-- 외계어를 보고싶지 않다면 반드시 써주자 --> |
|  | <title>NAVER</title> |
|  | <!-- title은 해당 웹페이지의 제목을 나타낸다 --> |
|  | <!-- NAVER는 NAVER, 내 블로그인 techhtml.org의 title은 ChoEun이다 --> |
|  | <link rel="stylesheet" href="style.css"> |
|  | <!-- link요소는 css를 가져올 때 유용하게 사용하는 요소이다 --> |
|  | <!-- css와 html은 다른 언어이기 때문에 css파일을 가져오는 일련의 작업을 진행해줘야한다 --> |
|  | <!-- rel이란 relationship의 약어로, link요소를 통해 가져오는 파일이 css라는 의미이다 --> |
|  | <!-- css는 Cascading Style Sheet의 약자이다 --> |
|  | <!-- 경로를 지정하는 방법은 절대경로와 상대경로가 있다. 이는 2번째 수업에서 상세히 다룬다 --> |
|  | </head> |
|  | <body> |
|  | <!-- body안에는 유저가 직접 볼 수 있는 콘텐츠를 넣는다. 여기서 유저란 여러분들의 웹사이트에 방문한 사람을 의미한다 --> |
|  | <!-- 일반적인 html요소의 대부분이 이 안에 들어가게 된다 --> |
|  | <h1>HTML</h1> |
|  | <!-- h1 요소는 페이지의 헤딩을 의미한다. 헤딩이란 장 제목, 절 제목정도로 생각하면 되겠다 --> |
|  | <!-- h1,h2,h3,h4,h5,h6이 있으며, 낮은 숫자에서 높은 숫자로 갈 수록 중요도가 낮아진다 --> |
|  | <!-- 예를 들어 h1 밑에 하위 제목이 있다면 h2를 사용하면 된다 --> |
|  | <!-- 이는 실습에서 자세히 다루도록 하자 --> |
|  | <p><abbr>HTML</abbr>은 Hyper Text Markup Language의 약자이다. 현재 <a href="http://www.w3.org/tr/html5">HTML5</a>가 가장 최신 버전이다</p> |
|  | <!-- p요소는 paragraph, 즉 문장을 나타내는 HTML요소이다. 각 HTML요소에는 의미가 있으며, 그 의미에 맞게 잘 활용하여야 한다 --> |
|  | <!-- abbr은 abbreviation, 즉 약자를 나타낼 때 쓰는 HTML 요소이다. --> |
|  | <!-- a는 anchor, 닻이라는 뜻이나 HTML에서는 링크를 표현하고 싶을 때 a 요소를 사용한다 --> |
|  | <!-- href(hyper reference) 속성을 사용하여 링크의 경로를 지정한다 --> |
|  | <!-- 이 또한 실습에서 자세히 다루어보도록 하자 --> |
|  | </body> |
|  | </html> |
|  | <!-- 일반적으로 HTML요소는 여는 태그(<html>)와 닫는 태그(</html>)이 한 세트이다. |
|  | 이를 늘 염두해두고 코딩을 하길 바란다. |

|  |
| --- |
|  |
|  | **h1{**  **/\*** |
|  | **CSS는 selector(h1), 여는 중괄호({), 닫는 중괄호(})로 구성한다.** |
|  | **selector안의 h1은 현재 에제에서 사용하는 것이 h1이고, 실제 사용시 여러가지를 사용할 수 있다.** |
|  | **예를 들어 지금 html파일에 있는 요소가, <html>, <body>, <h1>, <p>, <abbr>, <a>이고, 이 각각에 스타일을 지정하는 것이 가능하다.** |
|  | **<head>안에 있는 요소들에는 CSS로 스타일을 주는 것이 불가능하기 때문에 CSS에서 설명을 하지 않았다.** |
|  | **이는 아래로 이어지는 코드를 참고하도록 하자.** |
|  | **\*/** |
|  | **font-family: 'helvetica neue', 'APPLE SD Gothic Neo', 'malgun gothic', sans-serif;** |
|  | **/\*** |
|  | **속성명 (font-family)과 속성값('helvetica neue', 'APPLE SD Gothic Neo', 'malgun gothic', sans-serif) 사이에 띄어쓰기는 해주어도 되고 안해주어도 된다.** |
|  | **콜론(:)이 있어야 한다는 것만 늘 기억하자.** |
|  | **속성명과 속성값을 모두 기입한 후에는 반드시 세미콜론(;)을 작성하여야 한다.** |
|  | **\*/** |
|  | **/\*** |
|  | **font-family는 해당 요소에서 사용할 폰트를 지정한다.** |
|  | **지정하는 각 폰트 이름은 작은 따옴표(') 사이에 입력하며, 폰트의 구분을 쉼표(,)로 한다.** |
|  | **사용자가 사용하는 OS(맥은 Machintosh, 대부분의 PC는 Windows, 간혹 Linux)에 해당하는 폰트가 없다면, 뒤로 넘어가서 체크한다.** |
|  | **예를 들어, windows에는 'helvetica neue' 폰트와 'APPLE SD Gothic Neo'폰트가 없다.** |
|  | **이 경우에는 windows에 포함되어있는 폰트인 'malgun gothic'으로 폰트를 보여주게 된다.** |
|  | **APPLE SD Gothic Neo는 한글과 영문을 둘 다 표현하기 때문에, 위 코드와 같이 영문 폰트를 먼저 작성하고 한글 폰트를 작성하게 되면, 영문은 helvetica neue 한글은 APPLE SD Gothic neo로 보여지게 된다.** |
|  | **sans-serif는 고딕체를 의미하는 고유명사이다.** |
|  | **명조체를 보여주고 싶다면 serif를 가장 마지막에 지정하면 된다.** |
|  | **OS별로 사용하는 기본 폰트는 아래에 써져 있다.** |
|  | **가장 이상적인 셋은 위의 코드이다.** |
|  | **\*/** |
|  | **font-weight: 400;** |
|  | **/\*** |
|  | **font-weight는 폰트의 굵기를 표현한다.** |
|  | **폰트 굵기는 100부터 900까지로, 100단위로 올라간다.** |
|  | **즉 단위는 100,200,300,400,500,600,700,800,900이다.** |
|  | **일반적으로 100이 얇은 폰트, 400이 보통, 700이 굵은 폰트이다.** |
|  | **아래는 OS별 기본 폰트가 지원하는 굵기 현황이다. (사이트 : http://techhtml.org/lab/fonts/)** |
|  | **작업할 때 참고하도록 하자.** |
|  | **[Windows]** |
|  | **- Dotum (돋움)(다른 폰트를 지정하지 않고, sans-serif로 할 경우 기본 폰트)** |
|  | **Normal : 400** |
|  | **Bold : 700** |
|  | **- Gulim (굴림)** |
|  | **Normal : 400** |
|  | **Bold : 700** |
|  | **- Malgun Gothic (맑은 고딕)** |
|  | **Normal : 400** |
|  | **Bold : 700** |
|  | **[Mac]** |
|  | **- APPLE SD GOTHIC NEO (애플 SD 고딕 네오)(다른 폰트를 지정하지 않고, sans-serif로 할 경우 기본 폰트)** |
|  | **Thin : 100** |
|  | **UltraLight : 200** |
|  | **Light : 300** |
|  | **Regular : 400 (Default)** |
|  | **Medium : 500** |
|  | **SemiBold : 600** |
|  | **Bold : 700** |
|  | **Heavy : 800** |
|  | **- Helvetica Neue (헬베티카 뉴)** |
|  | **UltraLight : 100** |
|  | **Light : 300** |
|  | **Regular : 400** |
|  | **Medium : 500** |
|  | **Bold : 700** |
|  | **\*/** |
|  | **font-size: 28px;** |
|  | **/\*** |
|  | **font-size는 글자의 크기를 지정하는 CSS 속성이다.** |
|  | **font-size등을 지정할 때는 반드시 px등을 명시해야한다.** |
|  | **주로 쓰는 단위가 몇가지 있다. 2번째 수업에서 상세히 다루자** |
|  | **\*/** |
|  | **line-height: 32px;** |
|  | **/\*** |
|  | **line-height는 행간을 의미한다.** |
|  | **기본 값은 브라우저별로 다르기 때문에 font-size를 지정해줬다면 line-height를 항상 함께 사용하도록 하자.** |
|  | **이 경우는 단위가 여러가지가 있지만, 기본적인 단위는 역시 px을 주로 사용한다.** |
|  | **\*/** |
|  | **color: #555;** |
|  | **/\*** |
|  | **color 속성은 글자의 색상을 지정한다.** |
|  | **해쉬코드(#555)값이나 rgb(0,0,0) 혹은 rgba(0,0,0,1), 혹은 예약어(black, white) 등을 사용할 수 있다.** |
|  | **rgb는 아시는 바와 같이 Red, Green, Blue이며, a는 alpha값이다.** |
|  | **alpha값은 0에서 1까지의 사이로 100% 기준이고, 0.1은 opacity : 10%이다. 즉 0.01은 opacity:1%라 생각하면 된다.** |
|  | **이는 모든 색상 관련된 코드에서 동일하게 사용할 수 있다.** |
|  | **\* 최근에는 hsl, hsla도 지원한다. hsl은 아는 바와 같이 (hue-saturation-lightness)이다.** |
|  | **\*/** |
|  | **font-size: 32px;** |
|  | **/\*** |
|  | **css는 위에서 아래로 읽는 방식이기 때문에, 만약 같은 속성을 아래에 다시 작성했을 경우 위의 속성이 묻힌다.** |
|  | **즉 h1의 font-size는 32px이 된다.** |
|  | **\*/** |
|  | **}** |
|  | **html {/\* html 요소에 스타일을 지정한다 \*/** |
|  |  |
|  | **}** |
|  | **body {/\* body 요소에 스타일을 지정한다 \*/** |
|  |  |
|  | **}** |
|  | **h1 {/\* h1 요소에 스타일을 지정한다 \*/** |
|  |  |
|  | **}** |
|  | **p {/\* p 요소에 스타일을 지정한다 \*/** |
|  |  |
|  | **}** |
|  | **abbr {/\* abbr 요소에 스타일을 지정한다 \*/** |
|  |  |
|  | **}** |
|  | **a {/\* a 요소에 스타일을 지정한다 \*/** |
|  |  |
|  | **}** |

**20150205(목)**

1. SEO

- 웹 페이지의 랭크를 먼저 보고

- 그 다음에 title의 적합도를 기반으로 검색 엔진 최적화

🡪 따라서 우선 컨텐츠가 좋아야 하고

1. Title은 웹 접근성 차원에서도 중요하다. 페이지마다 유니크하고, 잘 전달이 되어야 한다.
2. element > attribute > value

<html lang=”ko”>를 하는 이유는 구글의 세부 검색에서 attribute를 ko로 지정한 게 접근성 차원에서도 중요하다…

1. 웹 접근성. Html 속성에 언어를 지정하는 게 접근성에서 유리하다. Lang 속성. Lang 속성은 global 속성! 그래서 <p lang=”en”></p>를 하기도 한다. 그 태그 속성에서는 en이 지엽적으로 정의가 되기도 한다.
2. Charset: character set 문자열 셋이 utf-8 (W3C에서 글로벌 공통으로 권장하는 파일)로…

<meta charset=”utf-8”> 또는

<meta charset=”utf-8” /> 둘 다 허용. 버전의 호환성 때문에 양 쪽으로 다 허용

**20150207 (토)**

1. <http://fecamp.in/>

2. 전용 폴더 만들기

3. <http://dece24kr.github.io/fecamp2/html/>

정찬명강사 전용

4. 1024\*768이 넘는 화면을 320픽셀로(모바일) 줄이다보니 글자가 작게. 그래서 meta 태그를 넣음

- <meta name="viewport" content="width=device-width">가 의미하는 것은…

viewport의 넓이를 width를 device의 width로 한다는 뜻

- 모바일에서의 viewport는 데스크탑의 viewport하고는 다름, 보이는 화면이 viewport가 아니라 문서의 크기를 viewport로 간주한다. 🡪 viewport 의 개념이 데스크탑과 모바일이 다르다.

- <meta name="viewport" content="width=device-320px">라고 할 수도 있지만, 320px로 해버리면 거기에 맞는 모바일 디바이스에만 최적화되서 호환성이 떨어질 수도 있다.

5. <meta name="viewport" content="width=device-width, initial-scale=2">

1은 기본이지만, 2로 하면 320 픽셀 안에 160만 들어오게 된다. 🡪 배율과 관련이 있다.

\* sublime에서 커맨드 + / 하면 인식하지 않는 노트를 할 수 있다.

6. <meta name="viewport" content="width=device-width, initial-scale=1, minimum-scale=1, maximum-scale=2">

- maximum-scale=2를 해줘야 모바일에서 확대가 가능하다.

7. minimum-scale과 maximum-scale을 무효화시키는 것도 있다.

<meta name="viewport" content="width=device-width, initial-scale=1, minimum-scale=1, user-scalable=no">

8 <meta name="viewport" content="width=device-width”>

W3C에서는 viewport나 width=device-width를 정의하지 않았고, 관행상 사실적 기준으로 업계가 쓰는 기준인 셈

9. 태그가 존재하지 않으면 css를 표현할 수가 없다. Html을 기반으로 하는 것이기 때문에…

10. <p style="background:yellow">Hello World!</p>

노란색 부분은 CSS 문법, html 문법이 아님.

근데, p element를 지정해준 css 문법은 🡪 Block Element!

11. css의 속성은 property라고 함. Html의 속성은 element. 물론 값은 value…

css는 세미콜론 ; 으로 속성 간 구분을 해준다.

12. head tag에서는 순서는 문제가 아니지만

13. css 스타일을 수정하는 방법

1) inline style 방식

<p style="background:yellow;text-decoration:underline">Hello World!</p>

2) embedded style 방식: 반복해서 적용…

p{background:red}

3) External style 방식

.. 🡪 원래 디렉토리라는 뜻

rel: reations

href: hyper text reference

결과적으로 inline > embedded > external 순으로 먼저 적용된다.

14. 로컬에서 직접 여는 방식: 커맨드 + o로 해서 html파일을 직접 열고. 그래서 방식을 확인

15. <http://nn.cloudvent.net/html5/>

저번 기수 작성 문서. 미디어 쿼리로…

css에서… if문

4주 후에 작성할 내용.

반응형 웹: 하나의 소스로 데스크탑 + 모바일 다 보이게…

15. div 태그는 유일하게 의미는 없고, 그룹핑을 하는 태그. 의미 없이…

덩어리를 그룹핑…

모든 페이지가 아니라, 특정 페이지만…

id=”container” 혹은 “컨테이너” 등 우리가 직접 정할 수 있는 것. 유니크해야 한다.

<div id="container”>

16. #은 찾는 거

#container

class는 . # 대신에 .

id와 class의 차이는? 질문… 🡪 아이디보다는 클래스를 먼저 고려하는 게 좋다?

id는 한페이지에 하나만 쓴다. id 값은 url이 될 수 있다. Id는 자원의 주소도 된다.

Class는 한페이지를 넘어선다.

17. margin: 0 auto 0 auto;

margin은

18. 타이틀은 큰 것 > 작은 것보다는

작은 것 > 큰 것으로 가는 게 ui, ux 차원에서 좋다

19. header or footer는 div와 달리 의미가 있고, 한 페이지에서 여러 번 등장할 수 있다.

20. 이런 코딩의 경우… div (단순한 분류)가 부모 element, header는 자식 element가 되고…

<div class="container">

<header class="header"></header>

21. 부모 element로 하나의 칼럼으로 wrapper 역할을 하는 div가 필요할 수도 있다.

이게 훨씬 유지 보수가 편하다.

<main></main>

<nav></nav>

<aside></aside>

순서는 이런 순서이고… 실제로 보이는 것과는 달리.

main (본문 컨텐츠, 다른 페이지에 반복되는 것은 아니다),

nav(링크들의 목록, 해당 웹 페이지의 주요한 내비게이션 역할)

aside

nav, aside는 한 페이지에서 두 번 이상 등장할 수 있다

22. class=”body”의 경우 자바나 css에서 .body를 찾아야…

23. #main은 페이지에서 id가 main을 찾게 된다.

24. lnb = local navigation bar

25. display block 이라고 설정되어 있지 않다면

모든 설정이 display inline으로 설정되어있다

block은 하나 더 입력하면 줄 바꿈을 하고

inline은 한 줄에 계속해서…

두 가지가 있는데… 질문?

26. .header>h1{

>는 자식 선택자를 의미

이럴 때 뒤에서부터 h1을 다 선택하고 난 다음 .header 밑에 있는 h1을 선택한다.

27. form 태그는 검색창 등 뭔가를 서버와 연동하게 하는 역할을 한다.

28. input 은 입력하는 기능을… 보통 type이라는 속성을 쓰는데… text, search, radio, checkbox 등…

29. required라는 속성은 뭔가를 입력하지 않으면 진행을 하지 않게 하는 속성

30. section, article, navigation, aside 태그는 그 웹페이지의 개요 역할을 하기에 w3c에서 제목을 쓰도록 강력하게 권하고 있다.

🡪 안쪽에다가 ‘제목’을 제공해야 한다. 검색최적화를 위해서도 필요

\* html5 outliner라는 프로그램은 w3c에서 권장하는 제목을 제대로 썼는지 확인하는 프로그램.

31. ul (unordered list 비순차 목록) 태그

ol(ordered list: 순차 목록) 태그

ul 태그는 자식 태그로 li 태그를 써주어야 한다. 그리고 그 사이에는 아무 것도 없어야 한다.

List item MUST!!!

Ul과 li를 쓰면 자동으로 뷸릿이 생겨난다.

32. html 문법 규칙은 반드시 지켜야 한다.

33. a href=”#”은 링크를 찾는 거고 #은 특정 정보

a는 inline 엘리먼트

html5부터 a만 블럭 엘리먼트를 감쌀 수 있다.

다른 엘리먼트는 블럭 엘리먼트를 감쌀 수 없다.

34. css에서도 한글이 깨지는 것을 방지하기 위해서는 @charset “utf-8”; 로 지정해야 깨지지 않는다.

20150207

정찬명 강사 서머리부분

[FE.CAMP] 첫째주 강의 요점 정리.   
// 페이스북이 접근성이 좋아서 이곳에 작성합니다.

# SW 설치  
1. Chrome 브라우저 설치. + HTML5 outliner( [http://goo.gl/VFCPuW](http://l.facebook.com/l.php?u=http%3A%2F%2Fgoo.gl%2FVFCPuW&h=2AQHj9HhJ&enc=AZN6cHQGoGSs_qiBHswpagkAkIQyKHD5PH0Lm-C7oc5kc8MObD-kJnjyYFAacGxTRZCjPRJYfVmS4q3NHdUcQXDhH_jdqLYxsfjVIPUci0tQErLUFdsWVbYezX_Iju4uEmY4ztChy_KJGfHQoesIQ3PM&s=1" \t "_blank) ) 확장기능 설치.   
2. SublimeText3 설치.  
3. github client 설치.(윈도우: [https://windows.github.com/](https://www.facebook.com/l.php?u=https%3A%2F%2Fwindows.github.com%2F&h=kAQGyy91t&enc=AZNYk8c4CXMyybPQd_9ukF1_LXk78vLQrmRJViFghJbuCxscZwwoSV6XJESWCQ0gvrHnpiaaa91uqj4AotOJUkrLfnhPtGutepLa2Ofany2yDLWi3nxDLlGYP-G8DMMNEGZzAF3i3MqZEz7F1bDBwjxW&s=1" \t "_blank) 맥:[https://mac.github.com/](https://www.facebook.com/l.php?u=https%3A%2F%2Fmac.github.com%2F&h=vAQGZxrpr&enc=AZN1uz4Owiv32If0EI-RcSc3KUeM4GW35vDoww1lBTtoyZKJZzM-Ta8OXgzziQtTZrw_u5b5Qpaa3Ci-8eh6R9L9VY_dnPi2Aw4cWfEFAlRl5nuYxKQu-Isis3kmi-6Tz7HVeOx_XZeZlvTl5AX5-xat&s=1" \t "_blank))

# 저장소 설정  
1. [http://github.com/](http://l.facebook.com/l.php?u=http%3A%2F%2Fgithub.com%2F&h=MAQFJnl5G&enc=AZOo9f9jF36MACUn8FraWDRoIVEU0vgMf1CVmEQ2U2uhp3m1L6-D3g17KNfYl686F7zINzyMaVPSxpscN4rWn_tpI5M2EbmoCwqwt0OEq8hN-WvYvsGqst1atykifINreyQ39-CL-gDQGBaOVU5vnwqo&s=1" \t "_blank) 사이트에 가입하면 웹 페이지 호스팅 및 도메인([http://userid.github.io/repository/](http://l.facebook.com/l.php?u=http%3A%2F%2Fuserid.github.io%2Frepository%2F&h=fAQGBHVc1&enc=AZOJ06T_i228h2EgUQY-IjNVnxrRdpRrPmiiLFMapEQi1DFqqmgA0g85YS2QdmKLsVjLoyhzfkfNNGtGDzzyaRixCAVE4_7lntUt4u--duwh0vbsuE7GUsA55m11pjwBfnzppZzkhgT91dCKmDMTKysp&s=1" \t "_blank)) 서비스를 무료로 이용할 수 있음. 최초 회원 가입 후 반드시 이메일 인증을 진행해야 제한 없이 서비스를 이용할 수 있음.  
2. /fecamp2/ 라는 이름의 저장소를 생성한 다음 깃허브 클라이언트에서 원격 저장소를 로컬에 복사(clone).  
3. /fecamp2/ 저장소의 master 브랜치 이름을 gh-pages 으로 변경하면[http://userid.github.io/fecamp2/](http://l.facebook.com/l.php?u=http%3A%2F%2Fuserid.github.io%2Ffecamp2%2F&h=jAQEl2Ehx&enc=AZPm5zQZruhwBXXyZPsfLecUDg7oZk2qCUtQ_eUtM3xrbSNeG8Hq5a8mPZ8KAu0hv1wEaNv1ByTE2ZYzTnjj1rq5Etef6u_gOOTcT2HqKcGcks1e8Dk_unSlBlrwmbkd4XA0jMderKdK6u3ZIJFVRzgz&s=1" \t "_blank) 도메인을 사용할 수 있다.  
4. /fecamp2/ 로컬 저장소에 /html/ 디렉토리를 생성하고 /html/ 디렉토리에 index.html 파일을 생성 index.html 파일에 "Hello World!" 라는 문자열을 작성한 다음 원격 저장소에 push(commit & sync)한다. commit은 로컬 저장소에 스냅샷(변경 사항에 대한 기록)을 저장하는 것이고, sync는 로컬 저장소의 파일과 스냅샷을 서버에 업로드하여 동기화 하는 것이다.

# 크롬 브라우저 사용  
1. Ctrl+(+), Ctrl+(-), Ctrl+(0) 키를 이용하여 화면을 확대/축소/원래대로 할 수 있다. 시각 장애인(약시)들이 이런 기능을 사용한다. 화면을 크게 보여주기 위해 강사도 이런 기능을 쓴다. 참고로 알아두자.  
2. 마우스 오른 버튼을 누르고 HTML 코드에 대한 요소 검사를 할 수 있다. 또는 Ctrl+Shift+C 단축키를 사용한 다음 확인하고 싶은 영역을 클릭해도 된다. 코드를 확인할 수 있는 부분을 인스펙터(inspector)라고 부른다.

# SublimeText3 편집기 사용법  
1. HTML 태그 이름을 작성하고 Tab 키를 누르면 열고 닫는 태그가 자동으로 완성된다.  
2. Ctrl+Shift+D 단축키를 사용하면 현재 라인을 복사하여 다음 라인에 붙여넣기 된다.  
3. Ctrl+Space 단축키를 사용하면 HTML 속성과 값 또는 CSS 속성과 값을 자동 완성할 수 있는 기능이 활성화 된다.  
4. Ctrl+/ 단축키를 사용하면 HTML 또는 CSS 코드의 주석을 작성할 수 있다. 이것은 JavaScript 코드에서도 마찬가지.

# HTML 실습  
1. DTD(Document Type Definition) <!DOCTYPE html> 를 작성하지 않으면 웹 브라우저는 쿽스 모드(quirks mode) 상태(Internet Explorer 5.5 버전의 렌더링을 흉내내는 모드)로 웹 페이지를 표시한다. 브라우저가 쿽스모드로 동작할 때 웹 페이지는 표준에 따르지 않는 방식으로 렌더링 되고 브라우저마다 서로 다른 모습으로 깨져 보이게 된다. 그러므로 DTD는 반드시 선언하고 DTD 앞에는 줄 바꿈이나 공백을 두지 않아야 한다.  
2. 뷰포트(viewport)는 웹 브라우저에서 내용을 표시하기 위한 영역을 말한다. 데스크톱 브라우저와 모바일 브라우저는 뷰포트 개념이 서로 다르다. 데스크톱 브라우저는 화면에 보이는 영역이 뷰포트가 되지만 모바일 브라우저는 브라우저의 크기(스마트폰 단말의 크기)와 관계없이 웹 페이지의 실제 크기가 뷰포트가 된다. 풍경을 담는 사진에 비유한다면 데스크톱은 카메라의 사각형 앵글이 뷰포트가 되고 모바일 단말은 앵글과 관계 없이 실제 세상에 존재하는 풍경 자체가 뷰포트가 되는 것이다. meta 요소를 사용하면 모바일의 뷰 포트를 데스크톱과 동일하게 변경할 수 있다. <meta name="viewport" content="width=device-width">  
3. <html lang="ko"> 웹 페이지에 주로 사용된 언어(human language)가 무엇인지 선언하여 검색엔진 최적화 및 웹 접근성에 대응한다.  
4. <meta charset="utf-8"> 현재 페이지에 사용된 코드와 모든 문자열의 문자셋을 선언한다. 문자셋 값을 "utf-8" 으로 선언해야 지구상의 모든 문자열을 오류 없이 표현할 수 있다.  
5. HTML 요소의 기본 골격은 <html><head></head><body></body></html> 이렇다. 이 요소들은 한 페이지에 한 번만 선언할 수 있고 선언 위치를 바꿀 수 없다.  
6. HTML 태그(<요소 속성="값"></요소>)의 문법 형식은 요소(element), 속성(attribute), 값(value). CSS의 문법 형식은(선택자{속성:값}) 선택자(selector), 속성(property), 값(value)으로 구성되어 있다.  
7. <div> 요소는 의미는 없고 용도만 있는 태그이다. 콘텐츠 블럭을 의미 없이(또는 스타일을 표현하기 위해) 감쌀 필요가 있을 때 사용한다. <span> 요소 또한 의미는 없고 용도만 있는 태그이다. 인라인 텍스트를 의미 없이(스타일 때문에) 감쌀 필요가 있을 때 사용한다.  
8. <header>, <footer> 요소는 페이지 전체 또는 특정 내용의 도입부와 맺음말 부분에 사용하는 의미를 가진 태그. <main> 요소는 한 페이지에 한 번만 사용할 수 있으며 콘텐츠의 본문 영역으로써 다른 페이지에서 반복되는 내용은 수록하지 않도록 한다. <nav> 요소는 사이트의 주된 탐색 링크 섹션을 의미한다. <aside> 요소는 부수적인 섹션을 의미한다.  
9. <section>, <article>, <nav>, <aside> 요소는 섹션을 만드는 요소다. 섹션을 만드는 요소에는 제목 <h2> 요소를 사용할 것을 강력하게 권장한다.  
10. <ul> unordered list, <ol> ordered list, <li> list item.  
11. 모든 HTML 요소는 블록 요소이거나 인라인 요소이다. 블록 요소는 콘텐츠 덩어리를 담는 요소이고, 인라인 요소는 텍스트(문자열)를 담는 요소이다. 블록 요소는 user agent stylesheet 에 의해 {display:block} 이라고 선언되어 있다. 인라인 요소에는 그런 선언이 없다. 블록 요소는 인라인 요소를 자식으로 포함할 수 있으나 인라인 요소는 블록 요소를 자식으로 포함할 수 없다. 예를 들면 <div><span></span></div> 이 코드는 유효하지만 <span><div></div></span> 이 코드는 인라인 요소가 블록 요소를 감싸고 있으므로 유효하지 않다. 예외적으로 <a> 요소는 인라인 요소임에도 불구하고 블록 요소를 자식으로 두고 감쌀 수 있다.  
12. HTML 주석은 <!-- --> 이렇게 생겼고, CSS 주석은 /\* \*/ 이렇게 생겼다. SublimeText3 에서 Ctrl+/ 단축키로 쉽게 작성할 수 있다.