

Day12

☰ 태그	
📅 날짜	@October 17, 2022

HTML 5는 웹 표준 기술이라고 한다. 어느 플랫폼(윈도우, 리눅스,, 맥....) 에서도 또 어떤 브라우저에서도 동일한 웹 화면을 구현하기 위한 약속이라고 할 수 있다.

웹 준으로 사이트를 작성하면 어떤 기기 어떤 브라우저 에서도 동일한 화면을 보여 줄수 있기 때문에 개발자와 디자이너들은 시간을 절약할 수 있다.

웹 표준을 지키는 HTML5로 문서를 작성하고 CSS3로 화면을 디자인하면 어떤 기기에서도 웹을 보여줄 수 있다.

결론 ... HTML5와 CSS3는 웹 에서 가장 기초적이고 가장 중요한 주제이다.

Web ???

World Wide Web, W3, 1989년 팀 버너스리에 의해 탄생 되었다. 인터넷에 연결된 사용자들이 서로의 정보를 공유하는 공간. 사실은 인터넷과 동일한 의미로 많이 사용되고는 있지만 정확히 말하는 웹은 인터넷 서비스 중의 인기있는 하나의 서비스 일 뿐이다. 하지만 현재는 인터넷과 웹이라는 단어가 혼용되어 동일하게 사용되는 경향이 있다 그만큼 인터넷에서 가장 큰 부분을 차지 한다.

1세대 웹

정적 static 웹, 단순히 HTML로 구성된 페이지

2세대 웹

HTML에 CSS 그리고 Java Script 을 활용한 동적 Dynamic 인 페이지

3세대 웹

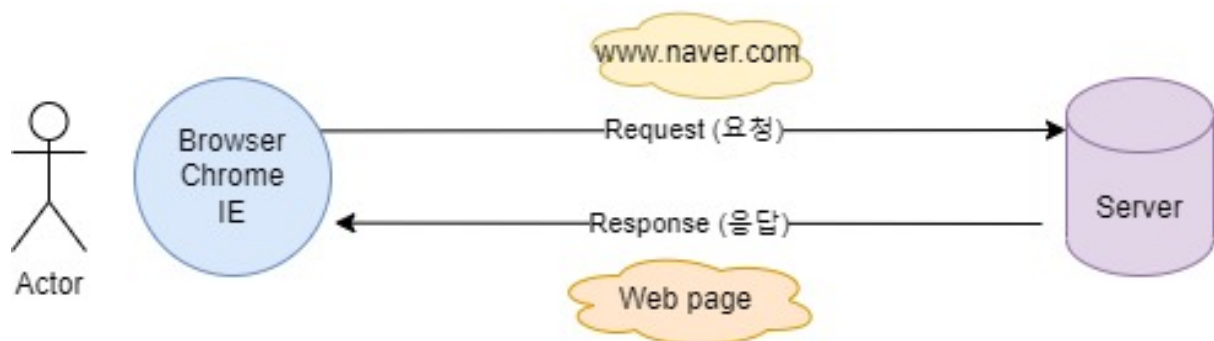
웹 페이지에서 사용자가 연관 제품 추천을 통해 주문을 하거나 유튜브에서 연관 동영상
을 추천해주는 등의 기능 넷플릭스에서 단순히 영화만 보는 것이 아니라 사용자가 좋아
할만한 영화들을 추천해주는 기능

4세대 웹

?????

WWW의 4요소 (팀 비너스리)

- 웹 브라우저
- 웹 서버
- HTTP - Hyper Transfer Protocol 어떻게 메시지를 주고 받을 지를 정한 규약
- HTML - 브라우저 화면 출력



1. 사용자가 웹 브라우저와 웹서버를 이용하여 통신한다.
2. 웹 브라우저가 특정 웹 주소를 웹 서버에 요청한다.
3. 웹 서버는 해당 페이지를 찾아 웹 브라우저에 제공(응답)한다.
4. 웹 브라우저는 HTML을 해석하여 사용자의 브라우저에 띄워 준다.

HTML5

Hyper Text Markup Language 5 (2014년 10월) 기존의 html은 다루지 못했던 오디오, 비디오, 그래픽 처리 등의 다양한 기능들이 추가되려 웹페이지 자체에서 처리 할 수 있도록 향상 되었다.

기존 버전에서는 플러그인 이라는 이름으로 별도의 프로그램을 붙여서 사용하여 범용성과 호환성이 떨어 졌으나 HTML5부터는 자체 지원하여 유연하게 처리 할 수 있게 되었다.

CSS3

Cascading Style Sheet 의 약자로 웹에서 모양을 미리 정해둔 스타일 시트이다. HTML 기본은 디자인에 상당한 제약이 있어 이를 보완하기 위해 만들어진 표준이 CSS이다.

기본 구조

```
<!DOCTYPE HTML>
<html>

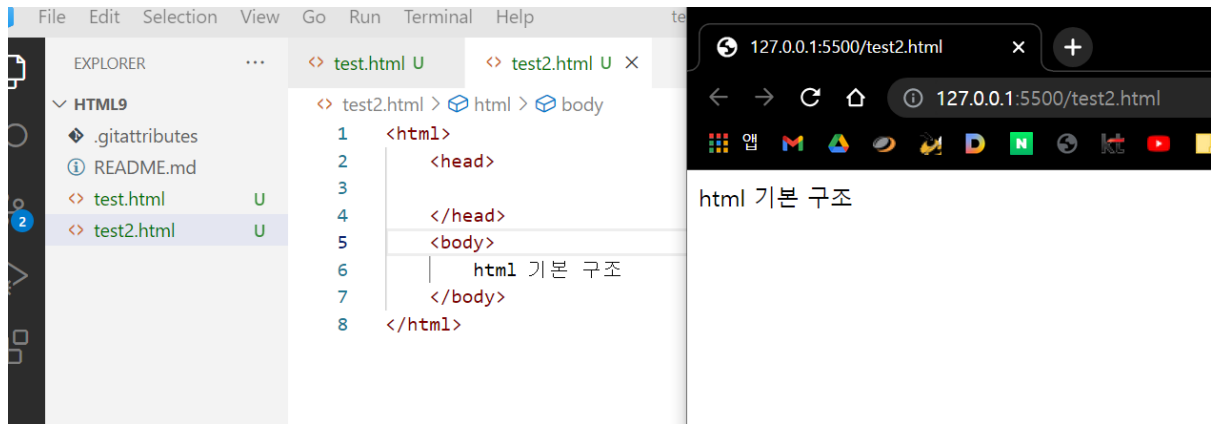
  <head>
    웹 페이지의 설정
  </head>
  <body>
    웹페이지 코드
  </body>

</html>
```

<html></html> 전체 html 문서를 감싸는 태그

<head></head> 문서 정보 설정 등을 나타내는 태그

<body></body> 실제 화면에 보여지는 부분을 나타내는 태그



태그 기본

<시작 태그></끝 태그> <START TAG> </END TAG>

대부분의 태그는 시작과 끝으로 구성되는데 간혹 시작만 있는 태그도 있다.

중첩 태그는 겹쳐서 또는 포개서 사용되는 태그들이다.

예를 들면 위에서 <HTML> 시작과 끝 사이에 <HEAD><BODY>를 중첩시켜서 사용되는 것이다.

자주 사용되는 태그 정리

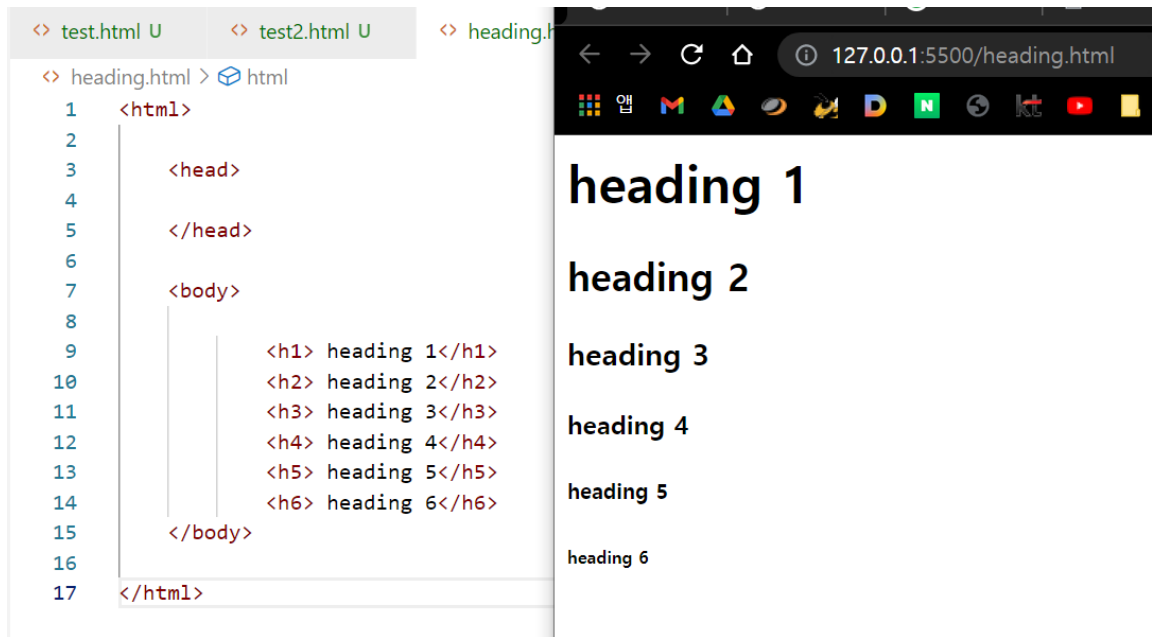
1. heading

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

h는 헤딩의 약자로 하나의 제목을 만들때 사용된다.

중요도에 따라 6단계로 구분하여 h1이 가장 중요 h6 이 가장 덜 중요한 헤딩으로 사용된다.

중요도 뿐이 아니라 크기로도 사용된다. h1이 가장 크고 h6이 가장 작다.

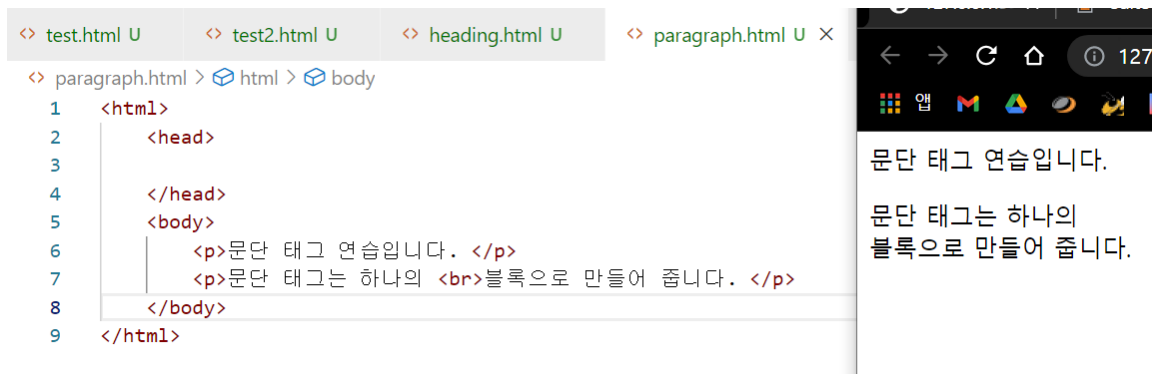


중요도에 따라 크기와 굵기가 다른것을 확인할 수 있다.

2. Paragraph 태그

`<p></p>` 문단의 약자로 하나의 문단을 만들 때 쓰인다.

문단은 항상 새로운 줄에서 시작하고 블록형태를 가진다.



`<p></p>`를 기준으로 앞뒤로 즉 위아래로 여분의 공백을 함께 표시한다.

`<p></p>`마다 새로운 줄에서 시작하며 사이에는 약간의 공백이 존재한다.

3. br 태그 Line Break

<> linebreak.html > html

```
1 <html>
2   <head>
3
4   </head>
5   <body>
6     First Line <br>Second Line
7   </body>
8 </html>
```



First Line
Second Line

 태그는 break를 뜻하며 정확히는 line break 이다.

<> linebreak.html > html > body

```
C:\WDevelop\html9\linebreak.html • Untracked
1 <html>
2   <head>
3
4   </head>
5   <body>
6     First Line <br>Second Line <br>
7     Third Line
8     Fourth Line
9   </body>
10 </html>
```

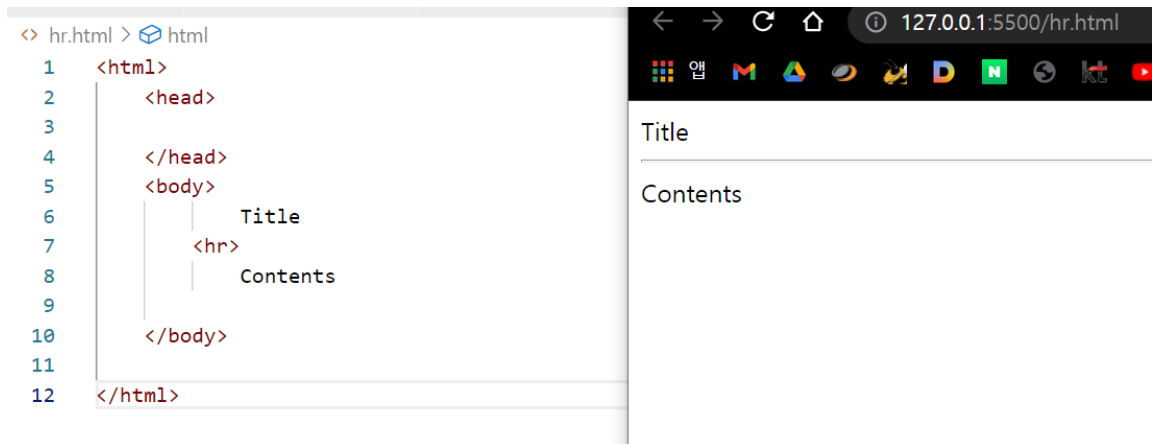


First Line
Second Line
Third Line Fourth Line

html 상에서는 줄을 바꾸어도 실제 줄 바꿈이 일어나지 않는다.
태그를 사용해야 줄을 바꿀수 있다.

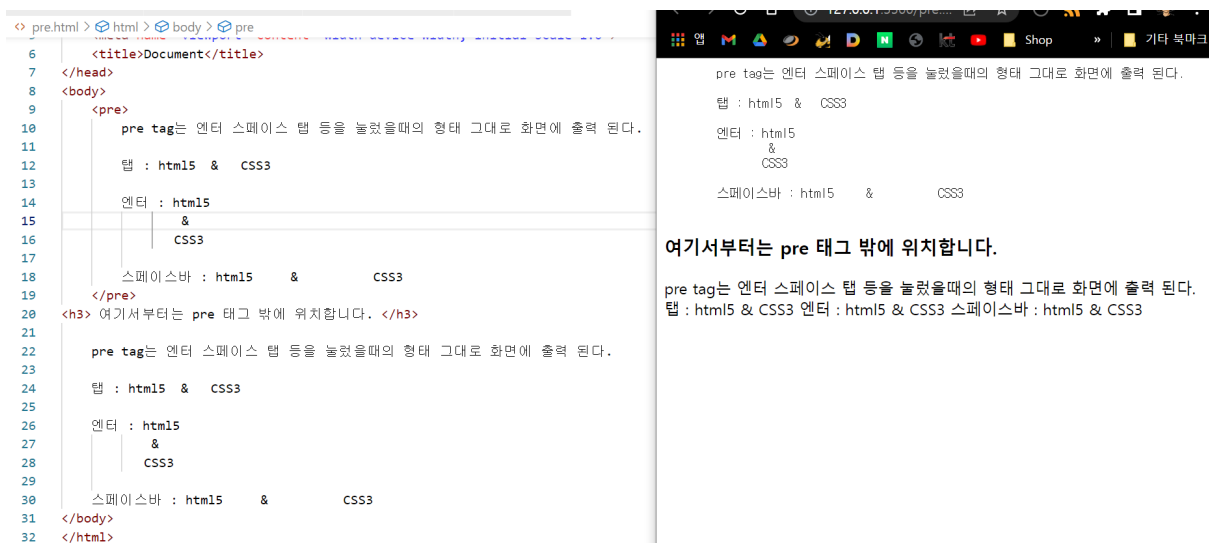
4. hr 태그 Horizontal Rules

<hr> 태그는
태그와 같이 시작만으로 사용하는 empty 태그이다. 기능은 단순히 웹 페이지에 수평선을 그려 준다. 웹페이지에서 내용을 의미적으로 분리할때 주로 사용된다.



5. formatted Text 태그

<pre></pre> 태그로 감싼 문장은 입력한 형태 그대로 웹페이지에 표현된다.



6. Text Formatting 태그

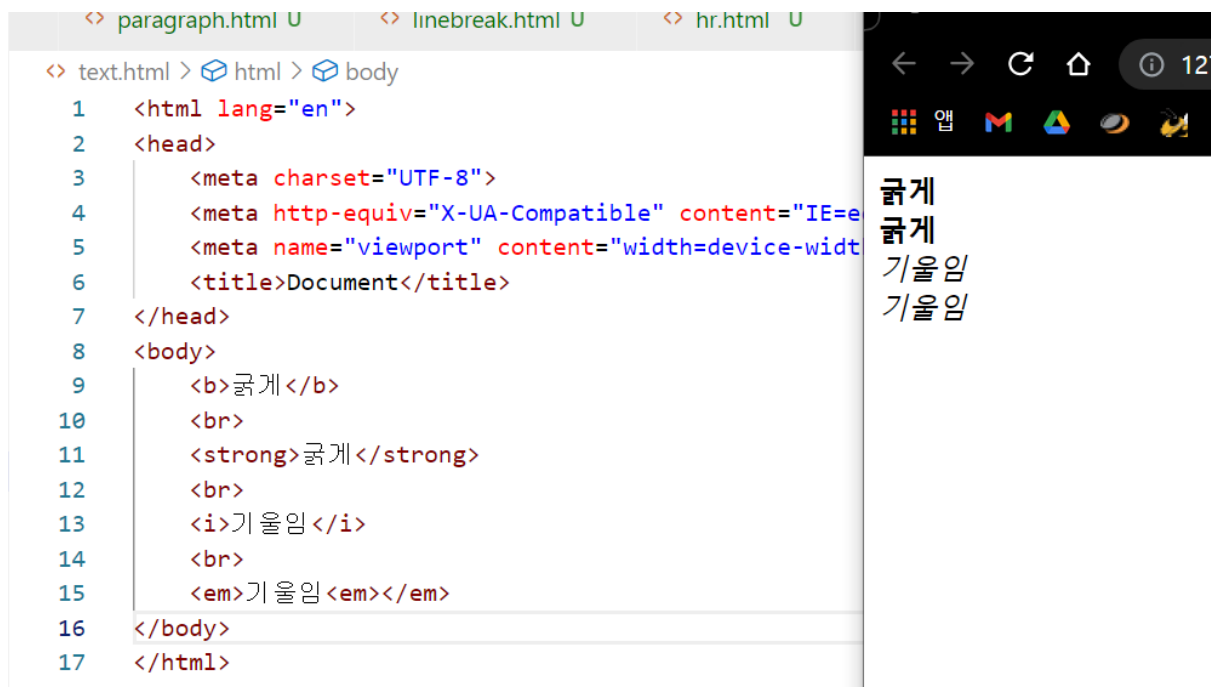
 bold 글자를 굵게

<steong> 글자를 굵게 + 중요하다



<i> italic, 기울임꼴

기울임꼴 + 중요하다.



<sub> 첨자

<sup> 위첨자

<ins> 밑줄

 취소선

<mark> 강조 형광펜

<small> 글자를 작게 표현

<> text.html > html > body > em > small

```
1 <html lang="en">
2 <head>
3   <meta charset="UTF-8">
4   <meta http-equiv="X-UA-Compatible" content="IE=edge">
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <title>Document</title>
7 </head>
8 <body>
9   <b>굵게</b>
10  <br>
11  <strong>굵게</strong>
12  <br>
13  <i>기울임</i>
14  <br>
15  <em>기울임</em>
16  <br>
17  <sub>첨자</sub>
18  <br>
19  <sup>위첨자</sup>
20  <br>
21  <ins>밑줄</ins>
22  <br>
23  <del>취소선</del>
24  <br>
25  <mark>강조 형광펜</mark>
26  <br>
27  <small>글자를 작게</small>
28 </body>
29 </html>
```

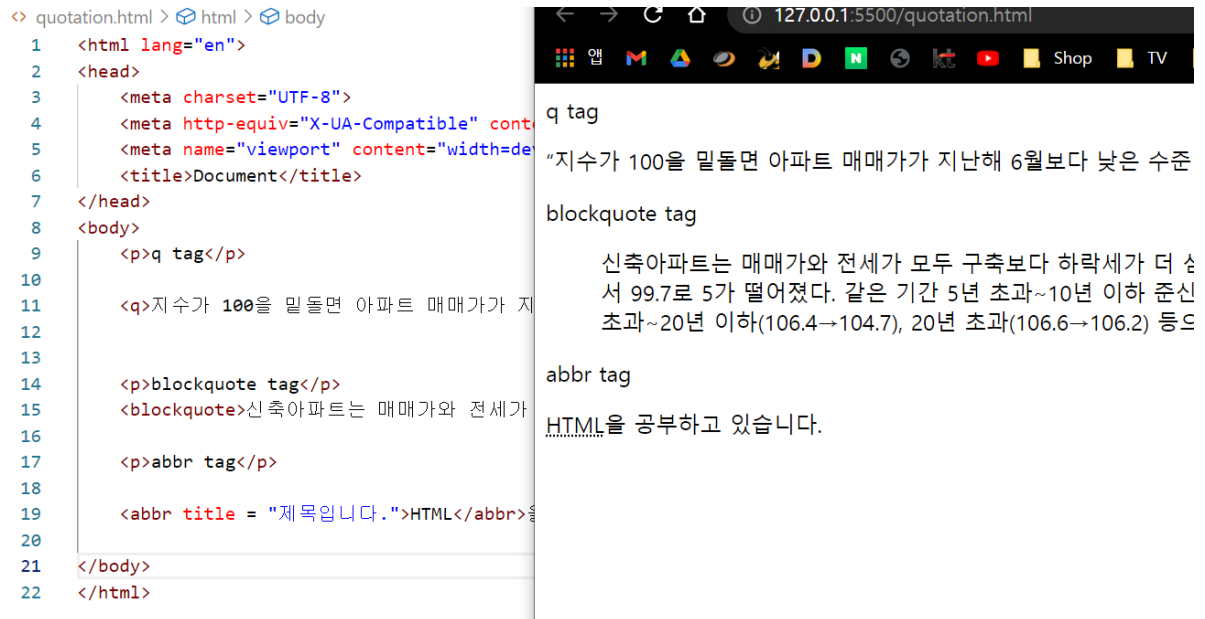
굵게
굵게
기울임
기울임
첨자
위첨자
밑줄
취소선
강조 형광펜
글자를 작게

7. 인용 태그 Quotation

<q></q> 짧은 문장을 인용 표시할때 쓰이는 태그이다. 브라우저에서 보여질 때는 큰 따옴표로 나타난다.

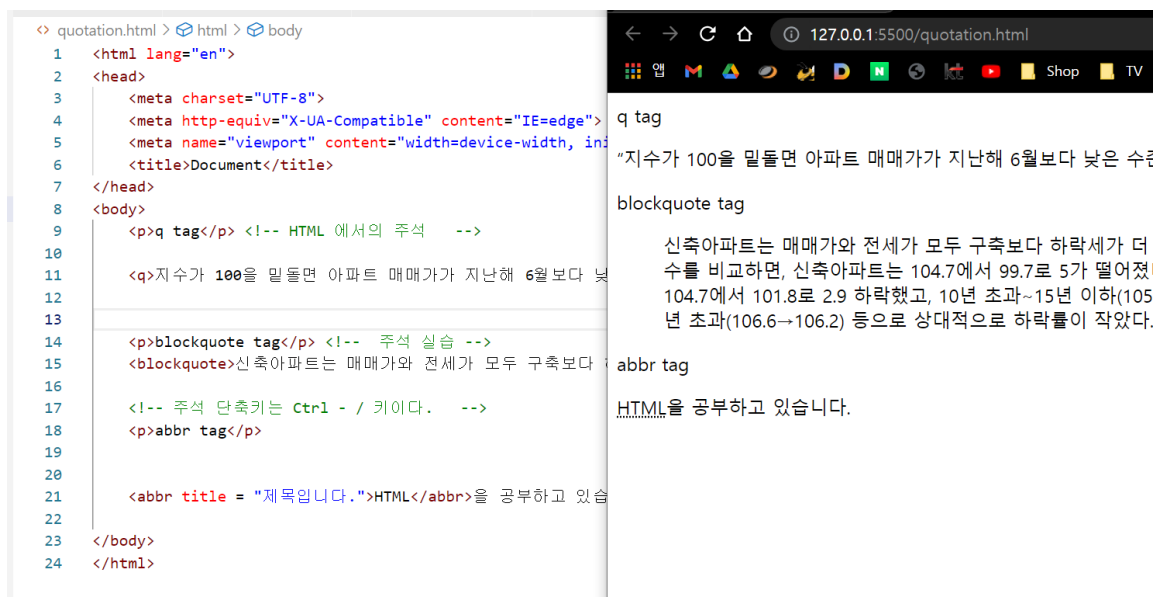
<blockquote></blockquote> 긴 문장을 인용표시 할 때 쓰이는 태그이다. 브라우저에서 보여질 때는 들여쓰기로 나타난다.

<abbr></abbr> abbreviation 태그는 마우스를 올리면 풍선 도움말 처럼 표시해준다.



8. Comment 주석

주석은 브라우저에는 나타나지 않는 개발자 또는 팀원들만 보는 기능

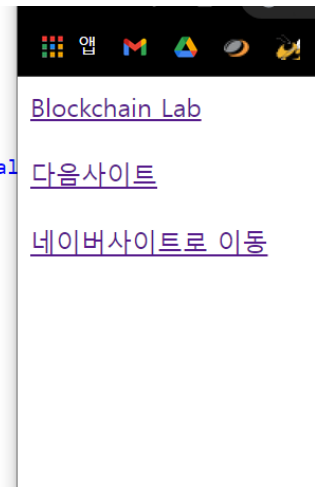


9. Link

하이퍼 링크 HyoerLink 글자 위로 마우스를 올리면 커서의 모양이 바뀌고 클릭을 하면 해당 링크 문서로 이동하게 된다.

표시할 글자

```
<> link.html > ...
1  <html lang="en">
2  <head>
3      <meta charset="UTF-8">
4      <meta http-equiv="X-UA-Compatible" content="IE=edge">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7  </head>
8  <body>
9      <a href="http://austin.sogang.ac.kr">Blockchain Lab</a>
10     <br><br>
11     <a href="http://daum.net">다음사이트</a>
12     <br><br>
13     <a href="http://naver.com">네이버 사이트로 이동</a>
14 </body>
15 </html>
```



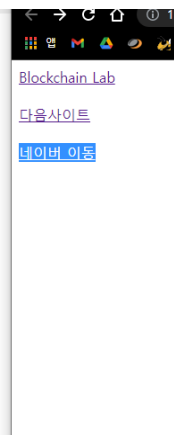
Link Option

_self : 기본값, 디폴트값 현재 창에서 바로 보여준다.

_blank : 가장 많이 사용되는 옵션 현재 창은 놔두고 새로운 창을 열고 거기에서 보여준다.

_parent, _top 등의 옵션도 있다.

```
<> link.html > ...
1  <html lang="en">
2  <head>
3      <meta charset="UTF-8">
4      <meta http-equiv="X-UA-Compatible" content="IE=edge">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7  </head>
8  <body>
9      <a href="http://austin.sogang.ac.kr">Blockchain Lab</a>
10     <br><br>
11     <a href="http://daum.net" target="_self">다음사이트</a>
12     <br><br>
13     <a href="http://naver.com" target="_blank">네이버 이동</a>
14 </body>
15 </html>
```



상속... Inheritance

상속은 실제 세계의 상속 처럼.. 즉 부모의 자산을 자식이 물려받아서 마치 자기 것 처럼 사용 할 수 있듯이 특정 클래스를 상속 받게 되면 그 클래스의 자산 (매서드, 변수,...)을 그대로 가져다가 사용할 수 있다. 다른 언어와는 달리 자바는 다중 상속을 지원하지 않는다.

일반적인 경우 자식 클래스는 부모 클래스의 모든 자산에 플러스 자신의 자산을 가지게 되므로 대부분 부모보다 자식 클래스가 더 많은 기능을 가지게 된다.

```
5 public static void main(String[] args) {
6
7     Dog dog = new Dog();
8     dog.setName("poppy");
9     System.out.println(dog.name);
10    dog.sleep();
11 }
12
13 }
14
```

poppy
poppyzzz

Dog 클래스는 Animal 클래스보 다 좀 더 많은 기능,,,여기서는 Sleep() 매서드를 가지게 된다.

보통 부모 클래스 보다 자식 클래스가 더 많은 기능을 가지게 되는게 일반적이다.

주의!!!

Dog 클래스는 Animal클래스를 상속했다 따라서 Dog는 Animal의 하위 개념이라고 볼수 있다.

이 경우 Dog 는 Animal에 포함되기 때문에 '개는 동물이다.' 라고 표현할 수 있다.

하지만 동물은 개만 될 수 는 없다. 따라서

Animal dog = new Dog(); //개는 동물이다. (O)

Animal dog = new Dog(); //동물은 개다. (X)

Dog dog = new Animal

개념적으로 생각해보면 동물로 만든 객체는 개 자료형이다. 로 읽을 수 있을 것이다. 하지만 말이 안된다. 호랑이 사자 말 등은 개가 될 수 없다. 성립할 수 없다.

```
Animal.java Dog.java Sample.java HouseDog.java
1 package inExam;
2
3 public class HouseDog extends Dog{
4
5
6     void sleep() {
7         System.out.println(this.name + " zzz in house");
8     }
9 }
```

```
13
14 HouseDog houseDog = new HouseDog();
15 houseDog.setName("happy"); //Animal 즉 할아버지의 기능
16 houseDog.sleep();
17 //Dog 즉 아버지의 sleep() 무시되고 자식이 가진 sleep() 이 호출된다.
18
19
20
21
22
```

Problems Javadoc Declaration Console

<terminated> Sample (1) [Java Application] C:\Program Files\Java\jdk-17.0.4\bin\javaw.exe (2022. 10. 17. 오후 3:00)

happy zzz in house

이렇게 부모 클래스가 가진 메서드를 자식 클래스에 동일한 이름의 메서드로 작성하면 자식이 가진 메서드가 호출된다. 이것을 메서드 오버 라이딩 이라고 한다. **method overriding**

```

    void sleep() {
        System.out.println(this.name + " zzz in house");
    }

    void sleep(int hour) {
        System.out.println(this.name + " zzz in house for " + hour + "hours");
    }
}


```

동일한 이름의 매서드가 여러 개 있을 때 인수의 종류 또는 갯수로 실행되는 내용을 바꾸고
자 할 때는 매서드 오버로딩을 이용한다. method overloading

```

14      HouseDog houseDog = new HouseDog();
15      houseDog.setName("happy"); //Animal 즉 할아버지의 기능
16      houseDog.sleep();
17      houseDog.sleep(3);
18
19
20
21
22
23

```

Problems @ Javadoc Declaration Console 
 <terminated> Sample (1) [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (2022. 10. 17. 오후 3:00)
 happy zzz in house
 happy zzz in house for 3hours

생성자 Constructor

생성자는 클래스명과 동일해야한다.

생성자는 리턴을 정의하지 않는다. void 도 사용하지 않는다

```
19
20     HouseDog houseDog = new HouseDog("happy");
21     // houseDog.setName("happy");
22     houseDog.sleep();
23     houseDog.sleep(3);
24
25
26
27
```

Problems Javadoc Declaration Console

<terminated> Sample (1) [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (202

happy zzz in house
happy zzz in house for 3hours

기본 생성자 (default constructor)

사실은 클래스에는 반드시 생성자가 존재해야한다.

개발자가 클래스를 만들 때 생성자를 만들지 않으면 자바가 자동으로 생성자를 생성해준다.

하나라도 생성자를 개발자가 직접 만들게 되면 자바는 기본 생성자를 추가로 생성하지 않는다.

```

5      // 생성자
6      HouseDog(String name){
7          this.setName(name);
8      }
9
10     HouseDog() {
11
12     }
13
14     HouseDog(int type) {
15         if (type == 1) {
16             this.setName("yorkshire");
17         } else if (type == 2)
18             this.setName("bulldog");
19     }

```

```

18
19
20     HouseDog houseDog1 = new HouseDog("happy");
21     HouseDog houseDog2 = new HouseDog(1);
22
23     System.out.println(houseDog1.name);
24     System.out.println(houseDog2.name);
25
26
27
28

```

Problems Javadoc Declaration Console
 <terminated> Sample (1) [Java Application] C:\Program Files\Java\jdk-17.0.4\bin\javaw.exe (2022
 happy
 yorkshire

위에서 HouseDog클래스는 세개의 생성자가 있다. 하나는 String 하나는 int 하나는 인수가 없다.

차이는 입력 항목이다. 이렇게 인수의 개수나 자료형으로 구분하여 생성자를 여러개 만들수 있는 것을 생성자 오버로딩 이라고 한다.

