Day13

Links (Hyper Link)

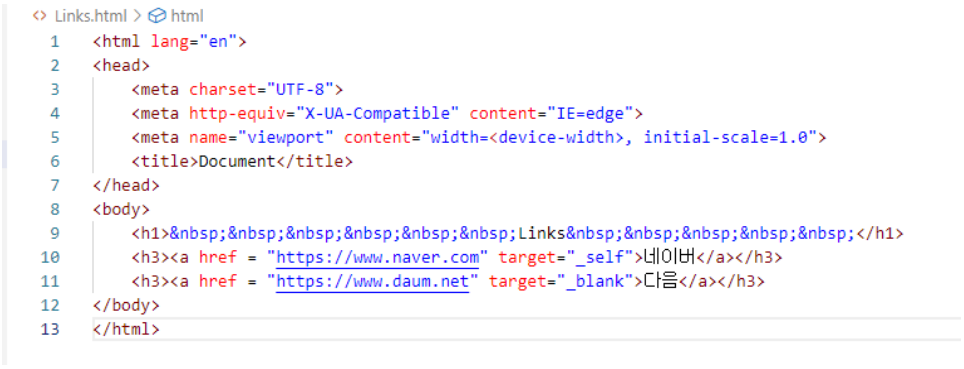
마우스를 글자나 그림에 올려 놓으면 마우스의 모양이 바뀌고 클릭을 하면 특정 주소 또는 페이지의 특정 위치로 이동하는 역할

```
<a href =”주소 or 위치> 글자 </a>
```

중요 속성

<a href=”주소 or 위치” target=”_self”글자 : 자신의 창에서 바로 이동

<a href=”주소 or 위치” target=”_blank”글자 : 새로운 탭 or 창을 열어서 이동



- 특정 위치로 이동

링크를 클릭하여 다른 페이지가 아닌 현재 페이지의 특정 위치로 이동 시킬때 사용

#이름표 를 사용하여 특정위치 name=”이름표”의 위치로 이동시키게 함

```
<h5><a href = "#article">화면 아래로 이동</a></h5>

<pre>

부모자식 사이에도 큰 돈이 오가게 되면, 차용증을 쓰고 이자를 주는 것이 좋다.

보통의 사인간 채무는 구두상의 계약만으로도 그 채무관계가 인정되지만, 직계존비속 간의 채무는 증빙이 없는 경

    이자를 지급하는 사람이 이자의 27.5%를 떼고 지급하고, 지급한 다음달 10일까지 세무서에 신고납부해야 한다.

    </pre>

    <a name = article">확인</a>
```

Image

웹 페이지에는 텍스트 뿐만이 아니라 많은 이미지를 활용하게 됨 이미지를 삽입하려면 <img> 태그를 사용

```
<img src = “이미지 파일” alt = “대체 텍스트”>
```

src = 뒤에는 이미지 파일의 이름이 오게 되는데 그 사진이 다른 위치에 있을때에는 주소값 까지 포함할 수 있음

위치 표현 2가지 방법

- 절대경로

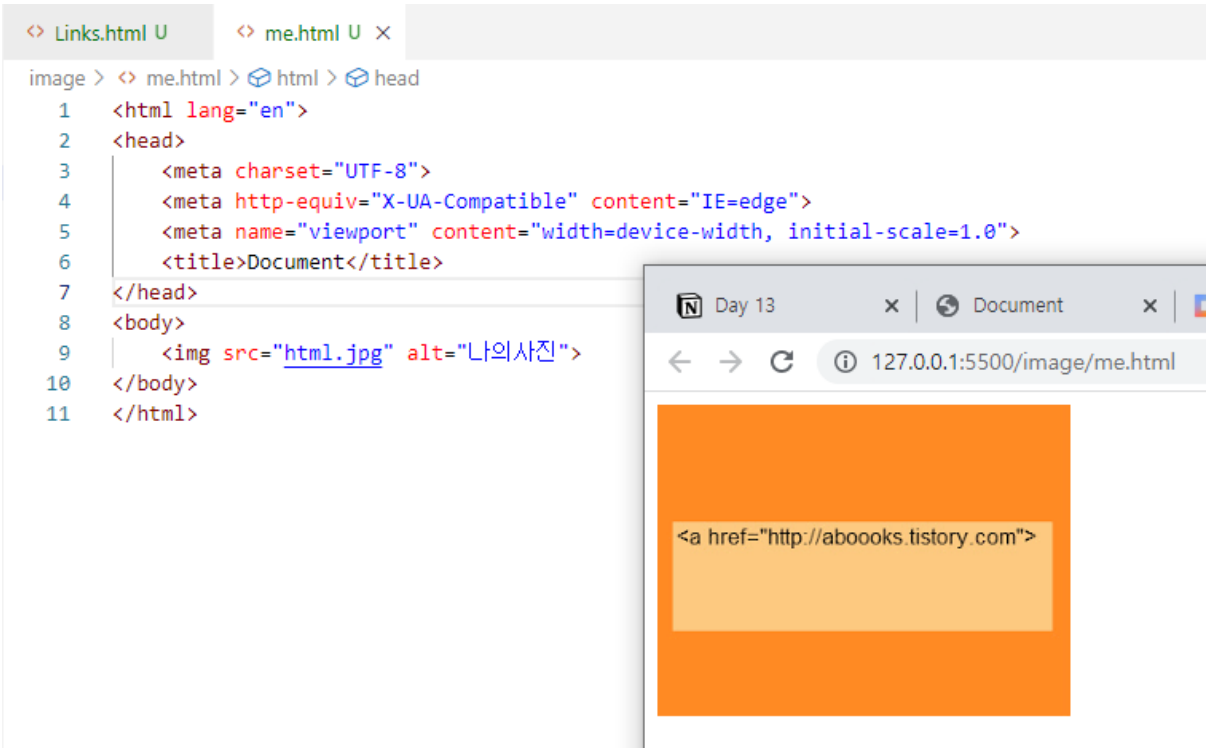
http://www.naver.com/image/me.jpg

- 상대경로

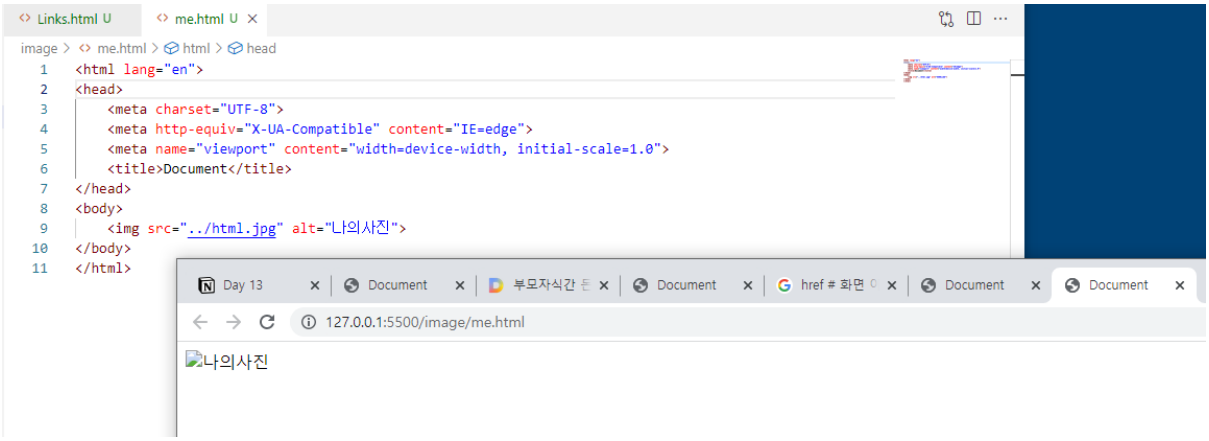
/image/me.jpg 루트 경로에서부터 내려옴

./image/me.jpg 현재 경로에서 찾음

../image/me.jpg 현재보다 상위 디렉토리에서 찾게 됨



img 태그에 이미지 파일 경로 입력하여 출력



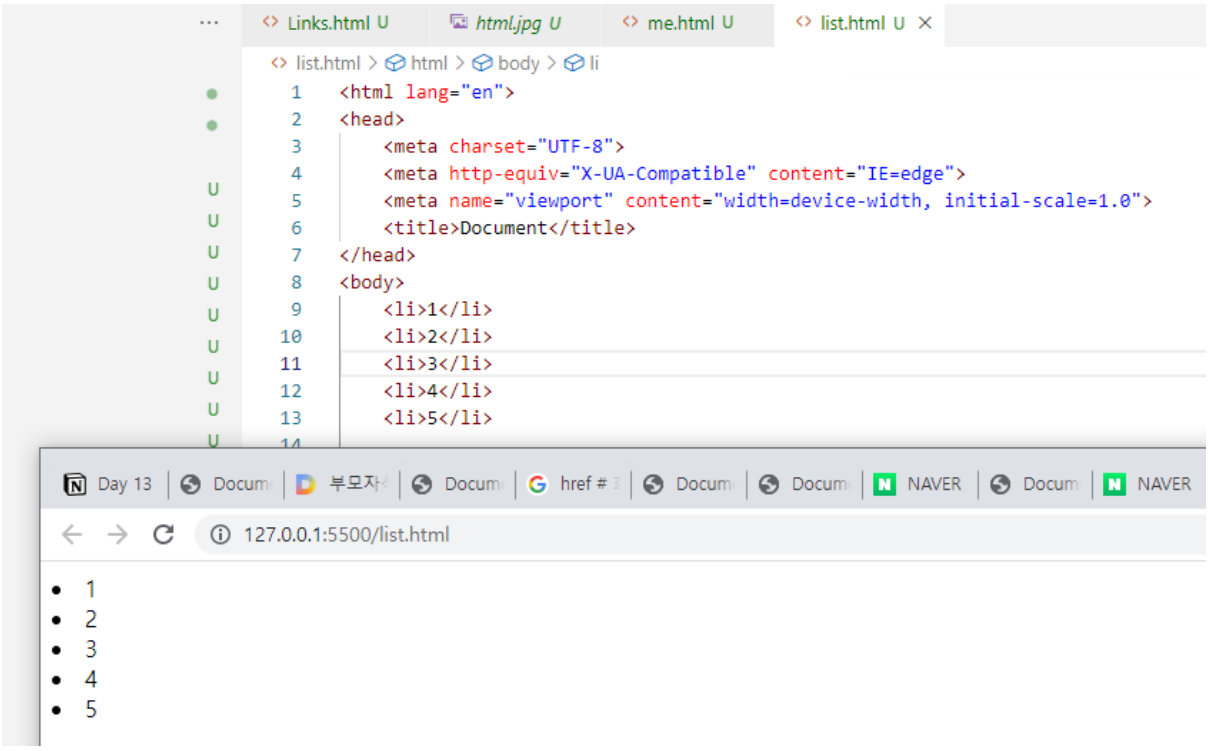
경로가 잘못될 경우 alt 문구가 나옴

List

- 순서가 없는 리스트 Unordered List UL
- 순서가 있는 리스트 Ordered List OL
- 정의 리스트 Definition List DL

1. 순서가 없는 리스트

항목을 나열할때 순서 없이 나열



중첩리스트

중첩리스트는 리스트 안에 또다른

2. 순서가 있는 리스트

type 옵션을 사용하면 다양한 모양으로 순서를 표현할수 있음

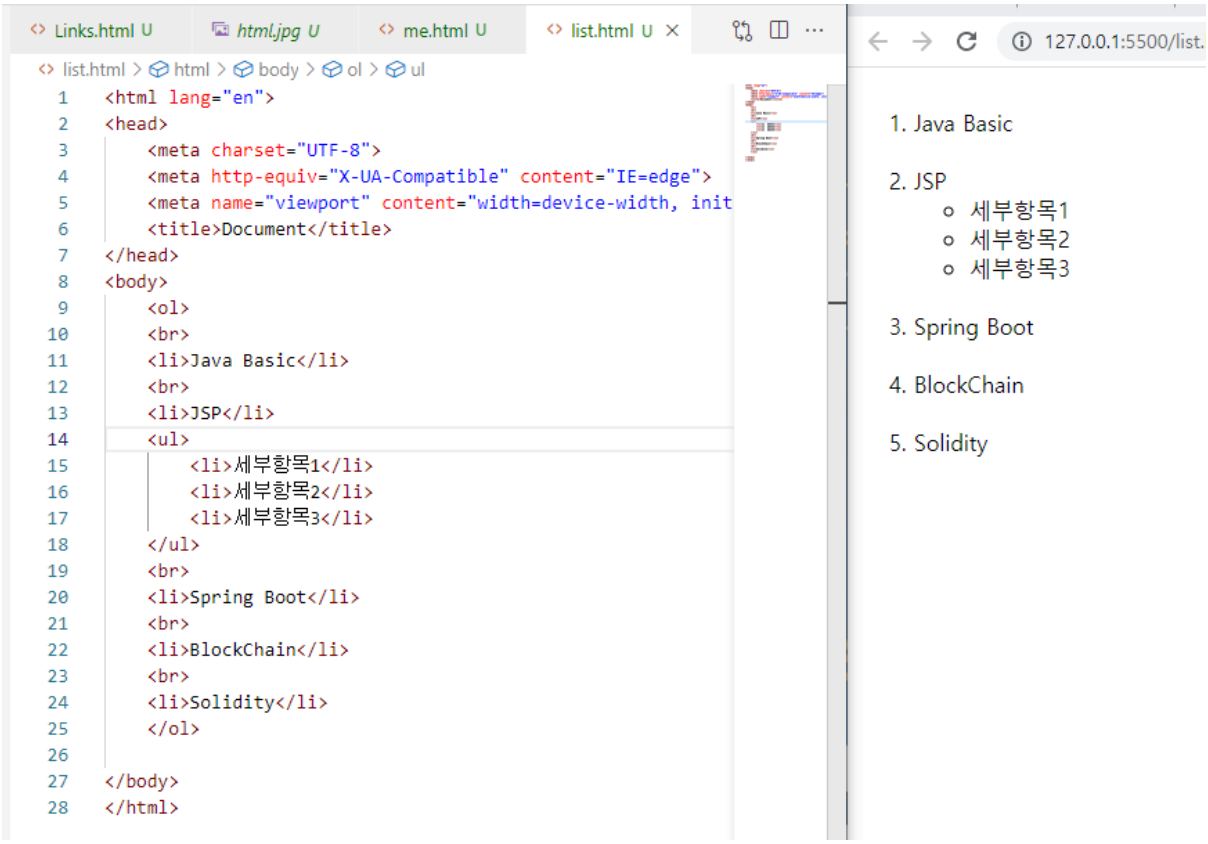
1 : 디폴트 값 아라비아 숫자

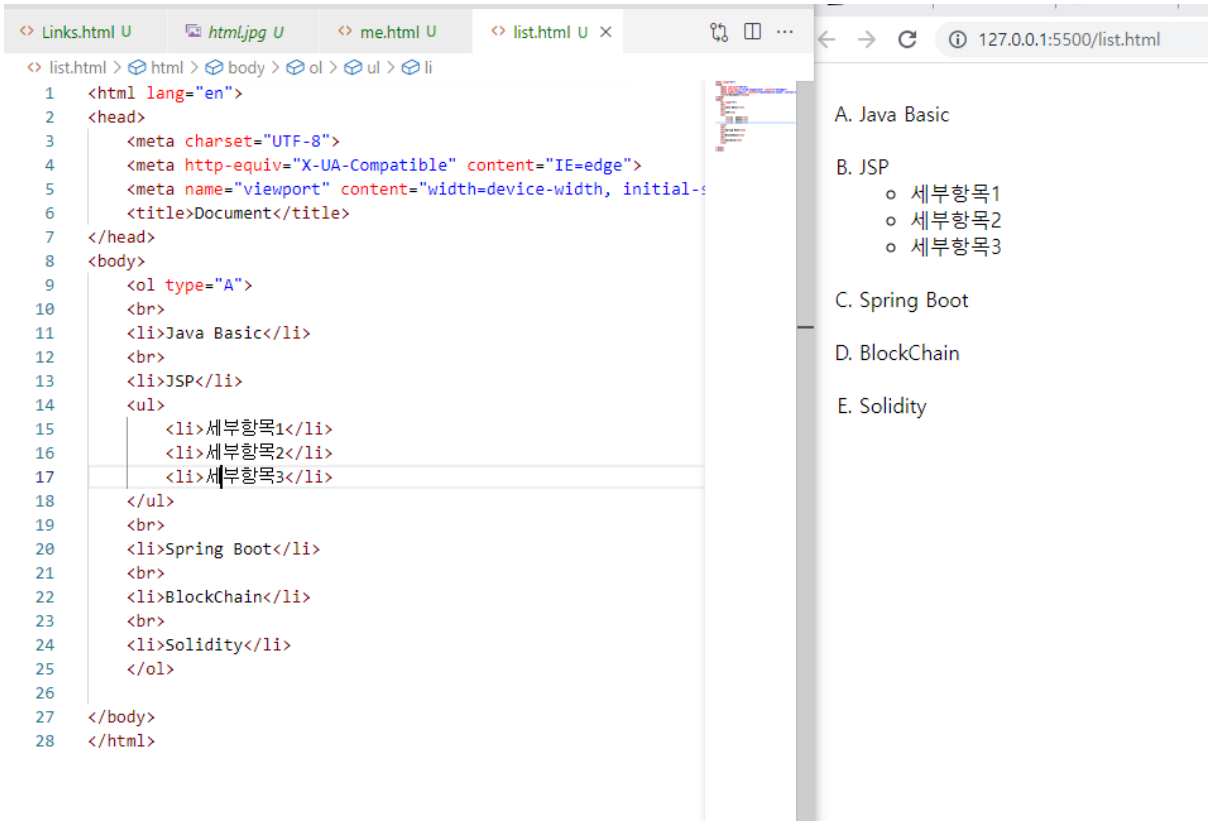
A : 영어 대문자 순서

a : 영어 소문자 순서

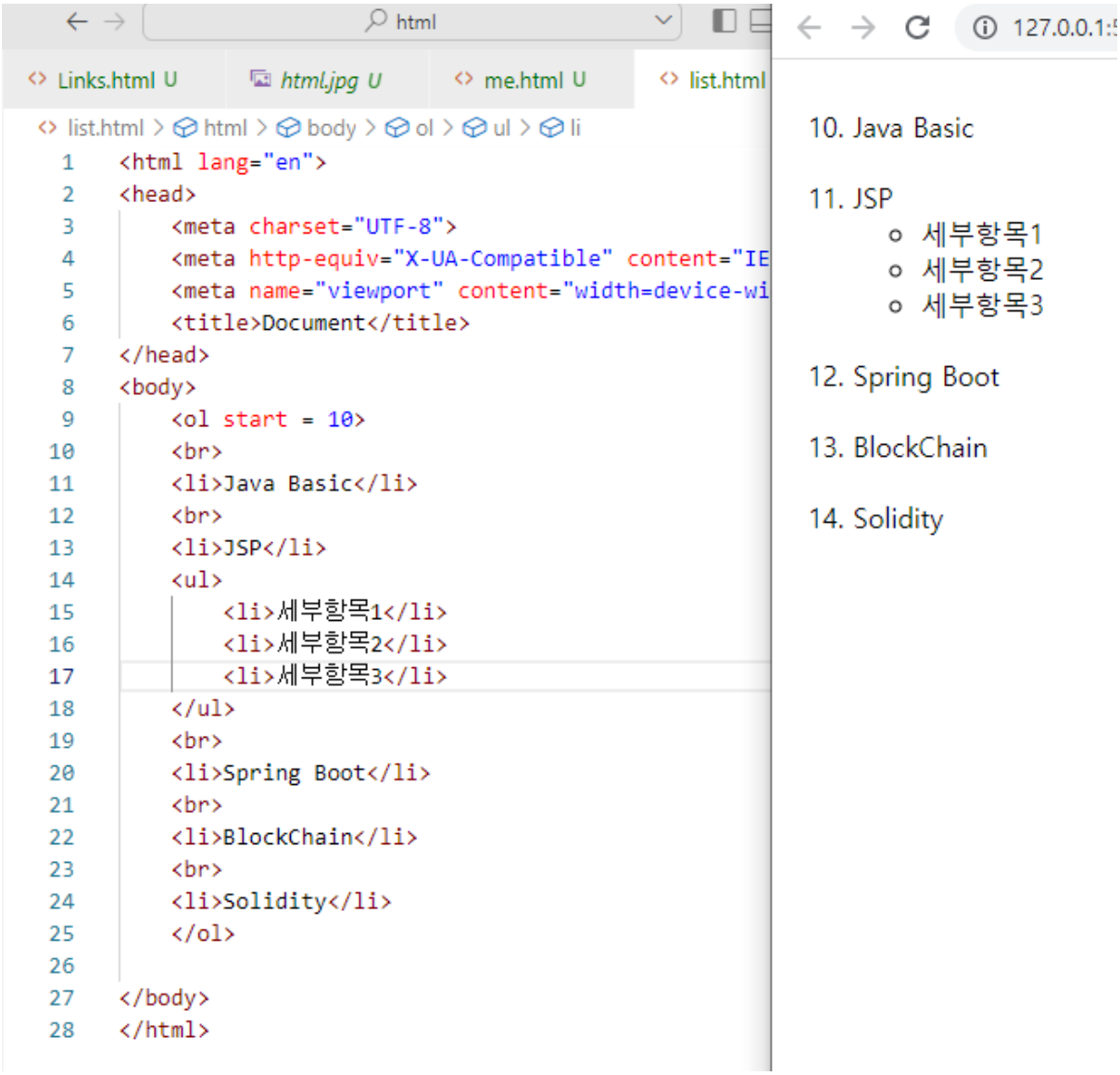
I : 대문자 로마 숫자

i : 소문자 로마 숫자

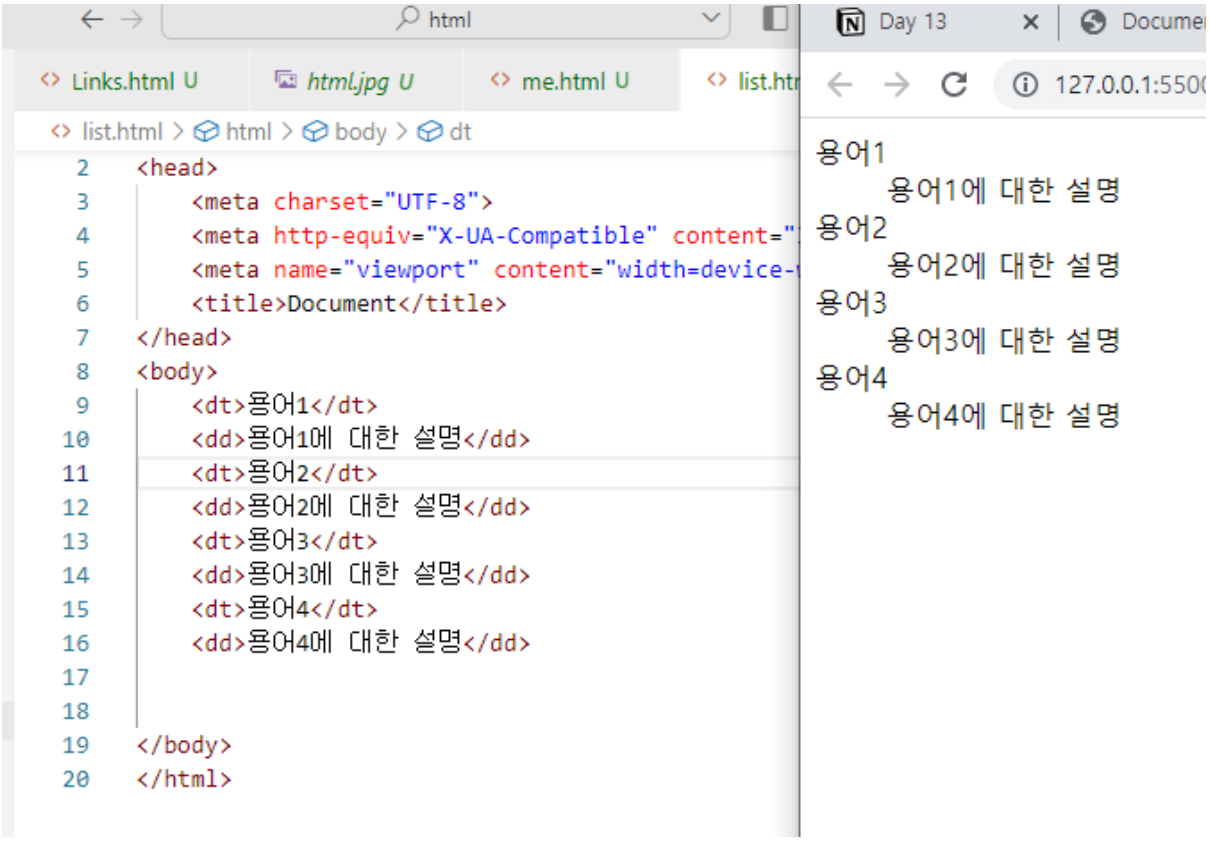




start 옵션으로 시작하는 숫자 정의할수 있음

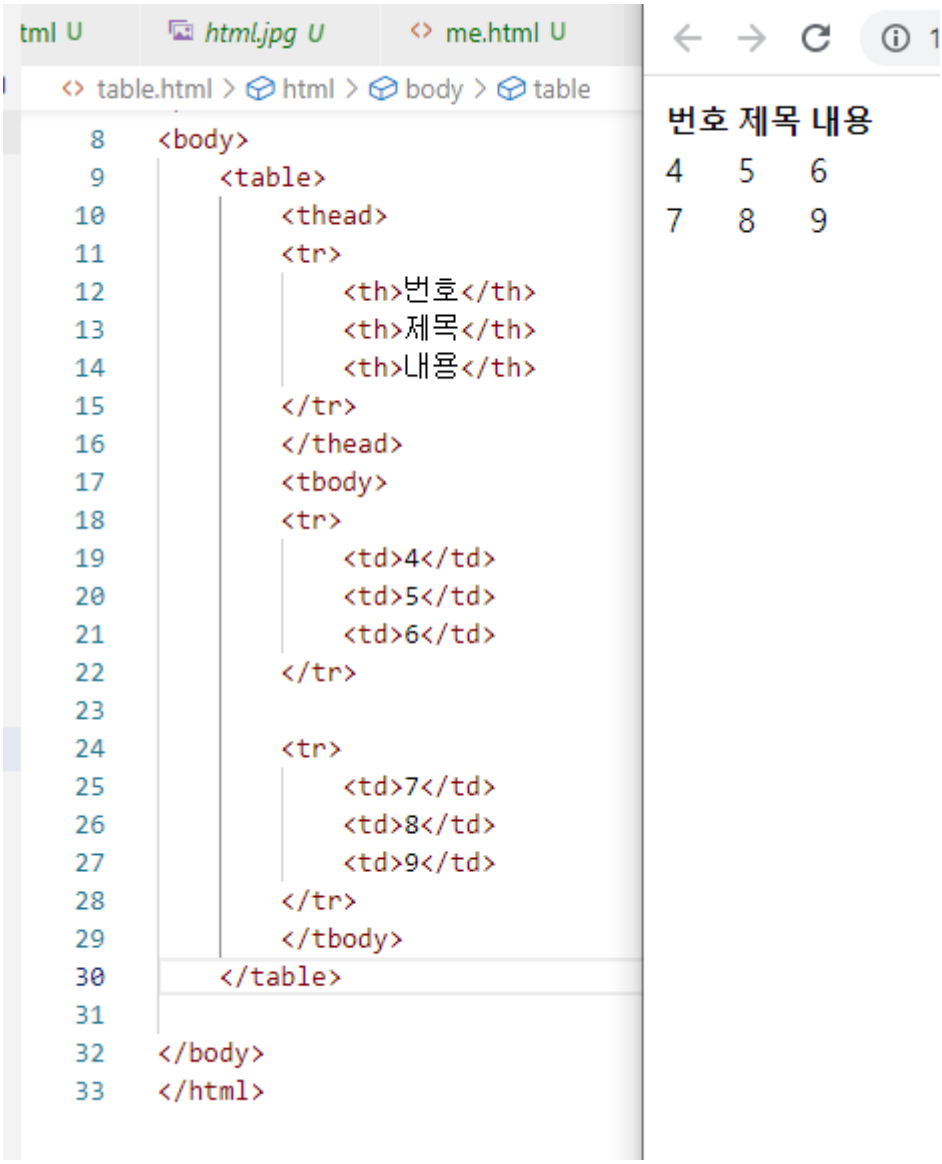


3. Definition List (정의 리스트)




```
dl : 사용자 정의 목록 선언
dt : 용어의 제목
dd : 용어의 정의
```

Table




```
tr 한줄   td 한칸thead : 테이블에서 헤더 영역을 의미
tbody : 테이블에서 내용 영역을 의미
th : 테이블에서 하나의 헤더칸을 의미
tr : 테이블에서 하나의 줄을 의미
td : 테이블에서 하나의 칸을 의미
```

Block & Inline

html 모든 태그를 Block 형태와 Inline 형태로 나눌수 있음

1. Block 형태

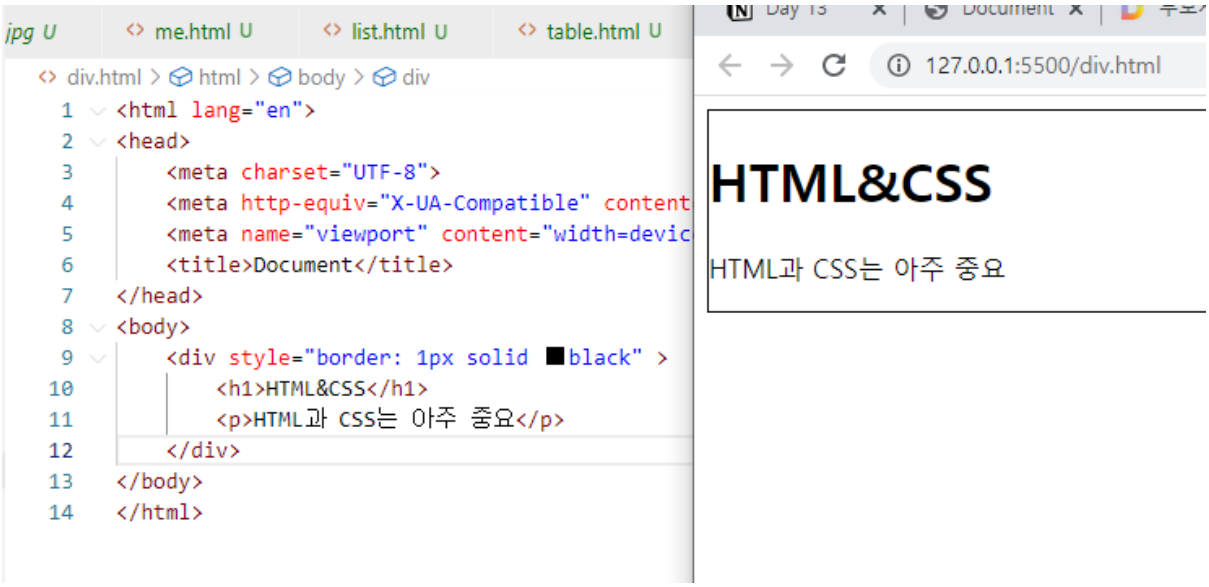
항상 새로운 줄에서 시작되고 그 줄 전체를 차지함

```
예)

<div> : 블록 형태의 가장 대표적인 태그

- html 태그들을 하나의 덩어리로 묶는데 많이 사용됨
- 하나로 묶은 html 태그들을
  한번에 CSS 스타일 적용하는데 많이 사용
- 특히 bootstrap에서 많이 사용

<p>,<li>,<ul>,<ol>,<h1> 등등
```



div 태그 블록에 다른 html 태그들을 넣어 하나로 묶을수 있음

위에 결과에서도 확인할수 있듯이 하나의 블록 형태를 가짐

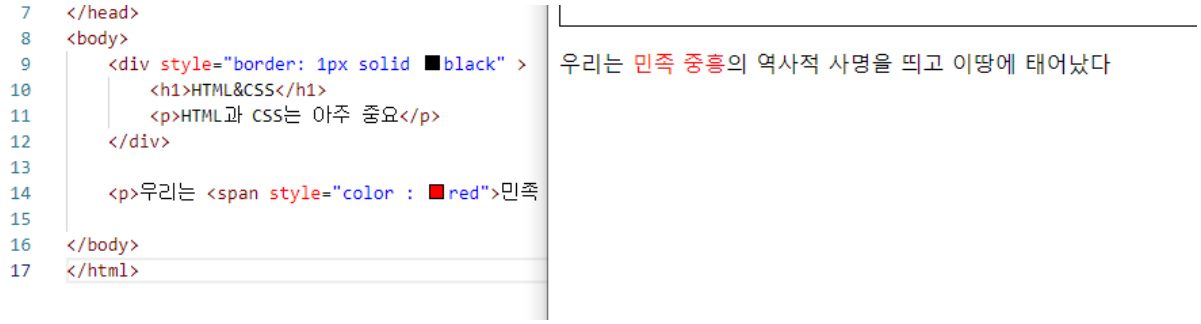
2. Inline 형태

새로운 줄에서 시작하지 않고 그 내용만큼만 너비를 가지게 된다

```
<span> : 인라인 속성을 가진 가장 대표적인 태그

- 텍스트의 특정 or 일부분을 강조하는데 주로 사용
- 텍스트의 일부분에만 css 스타일을 적용

<a>,<q>,<b>,<em>,<small>,<img>,<br> 등등
```

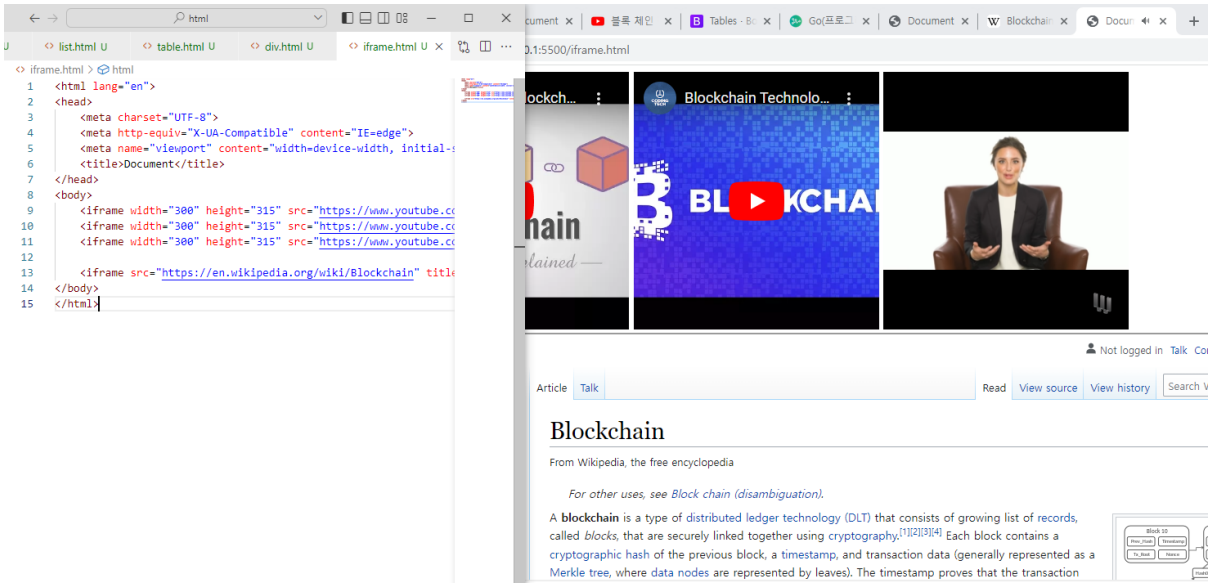


- 텍스트 일부분에만 스타일 적용 가능

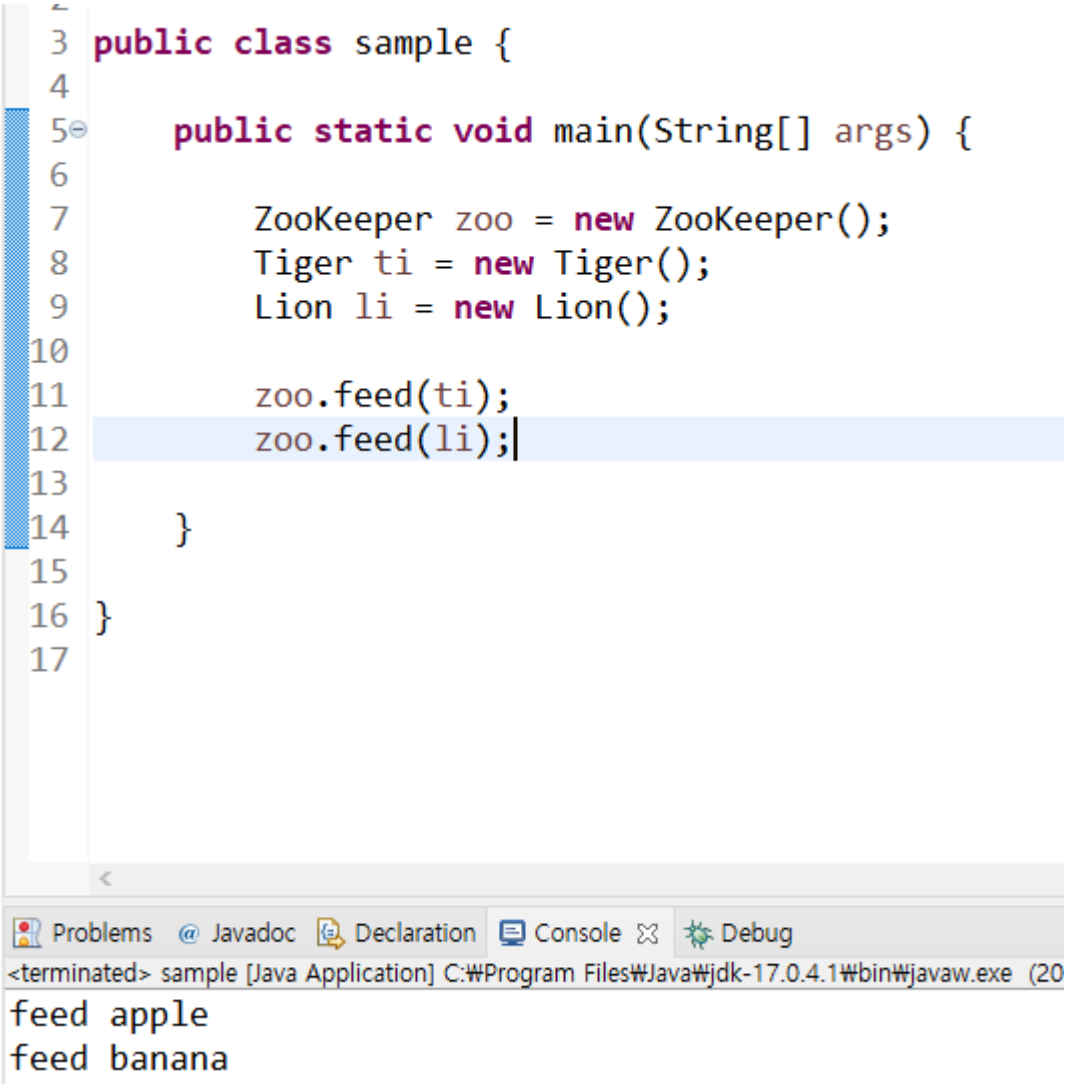
iframe

Inline Frame 의 약자로 웹페이지 안에 다른 웹페이지를 삽입하는 것을 의미

```
<iframe src = "삽입하는 웹페이지" title="제목">
</iframe>
```


iframe.html U
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-
<title>Document</title>
</head>
<body>
<iframe width="300" height="315" src="https://www.youtube.co
<iframe width="300" height="315" src="https://www.youtube.co
<iframe width="300" height="315" src="https://www.youtube.co
<iframe src="https://en.wikipedia.org/wiki/Blockchain" title
</body>
</html>
Blockchain Technolo...
Article
Talk
Read
View source
View history
Blockchain
From Wikipedia, the free encyclopedia
For other uses, see Block chain (disambiguation).
A blockchain is a type of distributed ledger technology (DLT) that consists of growing list of records, called blocks, that are securely linked together using cryptography.[1][2][3][4] Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data (generally represented as a Merkle tree, where data nodes are represented by leaves). The timestamp proves that the transaction

Interface



- zookeeper 객체는 각기 다른 매서드 오버로딩을 이용하여 호출해서 각각 다른 값을 출력 받을수 있음

- 이 방법은 동물이 추가될시 추가되는 동물마다 새로운 매서드를 계속 작성해서 추가해주어야하는 단점이 발생

```java
public interface Predator {

}
```

```java
public class Tiger extends Animal implements Predator{

}
```

```java
public class ZooKeeper {

//  void feed(Tiger tiger) {
//
//      System.out.println("feed apple");
//
//  }
//
//  void feed (Lion lion) {
//
//      System.out.println("feed banana");
//
//  }

    void feed (Predator pr) {
    System.out.println("feed " + pr.getFood());


    }

}
```

- 각각의 동물 별로 각각의 매서드를 작성해야 했지만

하나의 매서드 만으로 줄 수 있게 됨

- Tiger, Lion 객체가 각각 클래스의 자료형이기도 하지만

Predator 인터페이스를 구현하고 있기 때문에 Predator 인터페이스의 자료형이기도 함

```java
package InterfaceExam;

public interface Predator {

    String getFood();

}
```

- 동물의 종류만큼 feed 매서드가 필요했었는데 인터페이스를 이용하면 하나의 feed 매서드만으로도 구현이 가능

- 매서드의 개수가 줄었다는 것도 좋지만 ZooKeeper 클래스에게 의존적이였던 각각의 동물들 클래스가 독립적인 클래스가 된 점이 핵심

- 비교

실 세계	자바 세계
컴퓨터	ZooKeeper
USB 허브	인터페이스
프린터, 스캐너, 디스크	Tiger, Lion

다형성 (Polymorphism)

```java
public class sample {

    public static void main(String[] args) {

        ZooKeeper zoo = new ZooKeeper();
        Tiger ti = new Tiger();
        Lion li = new Lion();
        Bouncer bou = new Bouncer();

        bou.barkAnimal(ti);
        bou.barkAnimal(li);



    }

}
```

Problems Javadoc Declaration Console Debug

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

어흥

으르렁

- Bouncer 클래스는 동물을 짖게 만드는 역할
- Bouncer 클래스의 barkAnimal 매서드는 입력받은 객체가

  Tiger로 생성한 객체인 경우는 '어흥'을 출력하고

  Lion으로 생성한 객체인경우는 '으르렁'을 출력함

(어느 클래스로 생성한 인스턴스인지를 확인하는 명령어는 instanceof 임)

```
11 //  }
12     void barkAnimal(Barkable ba) {
13         ba.bark();
14     }
15
```

```
public interface Barkable {

        void bark();
}
```

```
public class Tiger extends Animal implements Predator,Barkable{

    public String getFood () {

    return "apple";
    }

    public void bark() {
        System.out.println("어흥");
    }
}
```

- 위에서처럼 여러 인터페이스를 동시에 다중 구현 할 수 있음
- lion과 Tiger 클래스는 predator와 Barkable 을 동시에 구현하고 있음
- predator의 자료형임과 동시에 Barkable의 자료형 이기도함
- 이렇게 하나의 객체가 여러개의 자료형 타입을 가질수 있는 것을 객체지향에서는 다형성이라고 함

Tiger 클래스

Tiger tiger = new Tiger();

Animal tiger = new Tiger();

Predator tiger = new Tiger();

Barkable tiger = new Tiger();

- Predator로 선언된 tiger 객체와

Barkable로 선언된 tiger 객체는 매서드가 서로 다름

- BarkablePredator 인스턴스를 만들어

Barkable 인스턴스, Predator 인스턴스를 상속 처리하여

매서드를 동일하게 처리할수 있음

그러나 매서드가 모두 존재해야함