

Day12

HTML5

- 웹 표준 기술
- 플랫폼, 브라우저 어디서든 동일한 웹화면을 구현하기 위한 약속
- HTML5로 문서를 작성하고 CSS3로 화면을 디자인함
- 가장 기초적이면서도 가장 중요한 주제

WEB

- World Wide Web , 1989년 팀 버너스리에 의해 탄생
- 인터넷에 연결된 사용자들이 서로의 정보를 공유하는 공간
- 인터넷과 동일한 의미로 사용되고 있으나 인터넷 서비스중 하나로 볼 수 있음

1세대 웹

- 정적 Static 웹, 단순히 HTML로 구성된 페이지

2세대 웹

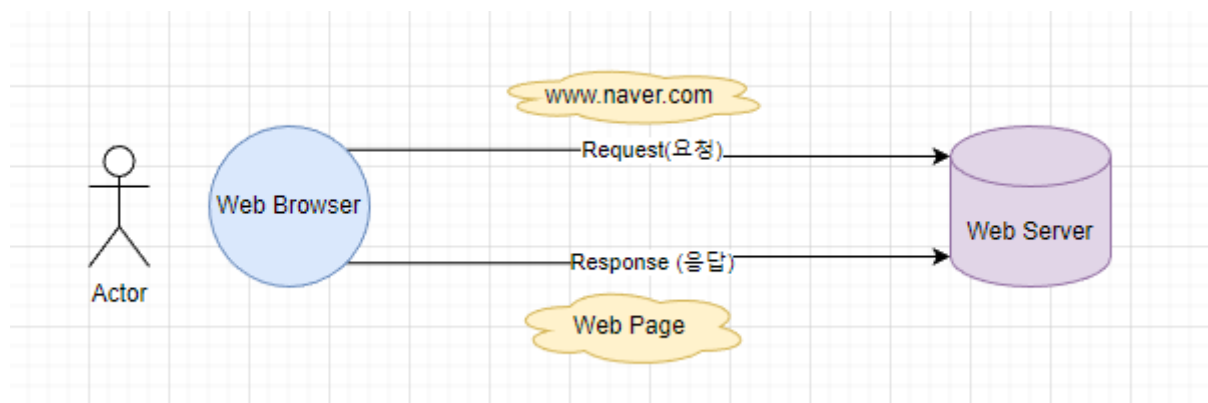
- HTML+CSS+Java Script 를 활용한 동적 Dynamic 인 페이지

3세대 웹

- 웹페이지에서 연관 제품 추천하여 사용자가 주문하거나 유튜브에서 연관 동영상을 추천해주는 등의 기능

WWW의 4요소

- 웹 브라우저
- 웹서버
- HTTP Hyper Transfer Protocol
- HTML



1. 사용자가 웹 브라우저를 이용하여 웹서버와 통신
2. 웹 브라우저가 특정 웹 주소를 웹 서버에 요청
3. 웹 서버는 해당 페이지를 찾아 웹 브라우저에 응답
4. 웹 브라우저는 HTML을 해석하여 사용자의 브라우저에 띄워준다

HTML5

- Hyper Text Markup Language 5 (2014년 10월) 기존의 HTML4는 다루지 못했던 오디오, 비디오, 그래픽 처리등의 다양한 기능들이 추가되어 웹페이지 자체에서 처리 할 수 있도록 향상됨
- 기존 버전에서는 플러그인 이라는 이름으로 별도의 프로그램을 붙여서 사용하여 범용성과 호환성이 떨어졌으나 HTML5부터는 자체 지원하여 유연하게 처리할 수 있게 됨

CSS3

- Cascading Style Sheet의 약자로 웹에서 모양을 미리 정해둔 스타일 시트
- HTML 기본은 디자인에 상당한 제약이 있어 이를 보완하기 위해 만들어진 표준

기본구조

```
<html >
<head>

    웹페이지의 설정

</head>
<body>

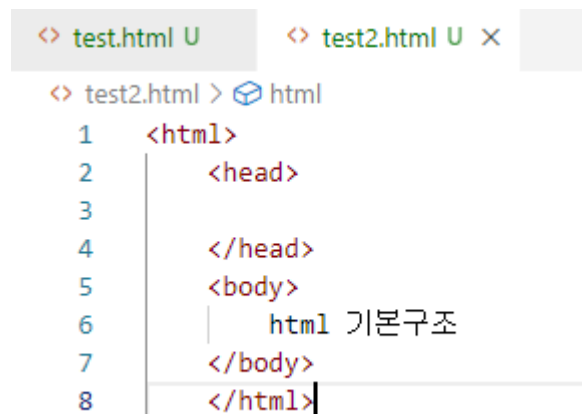
    웹페이지 코드

</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



```
<> test.html U    <> test2.html U X
test2.html > html
1  <html>
2    <head>
3
4    </head>
5    <body>
6      |   html 기본구조
7    </body>
8    </html>|
```

태그 기본

- <시작 태그></끝 태그>

대부분의 태그는 시작과 끝으로 구성되는데 간혹 시작만 있는 태그도 있음

- 중첩태그는 겹쳐서 또는 포개서 사용되는 태그

예를 들면 위에서 <HTML>시작과 끝사이에 <HEAD><BODY> 가 중첩되어 사용

자주 사용되는 태그 정리

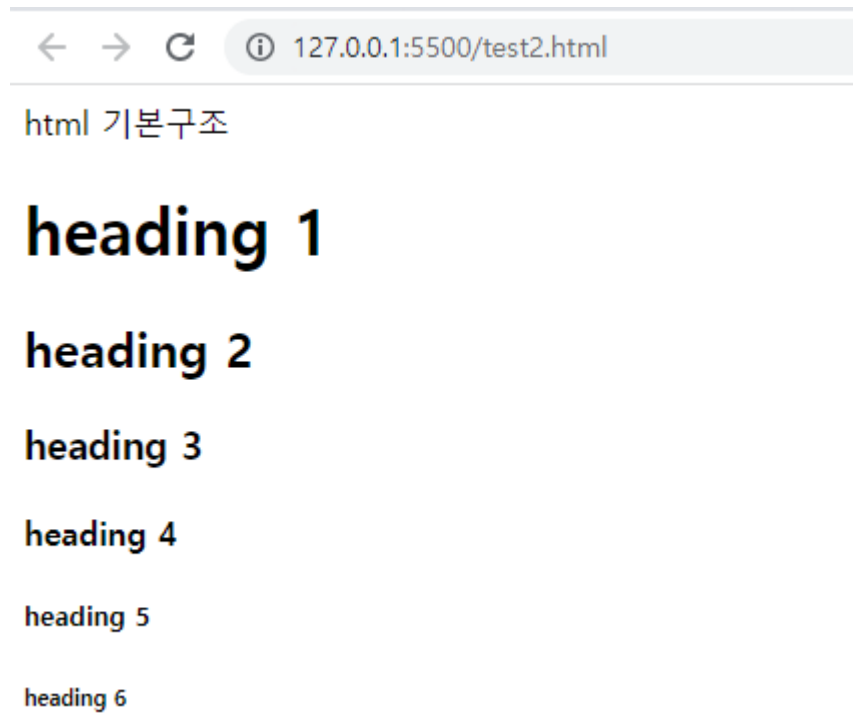
1. heading

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

h는 헤딩의 약자로 하나의 제목을 만들때 사용됨

중요도에 따라 6단계로 구분하여 h1이 가장 중요

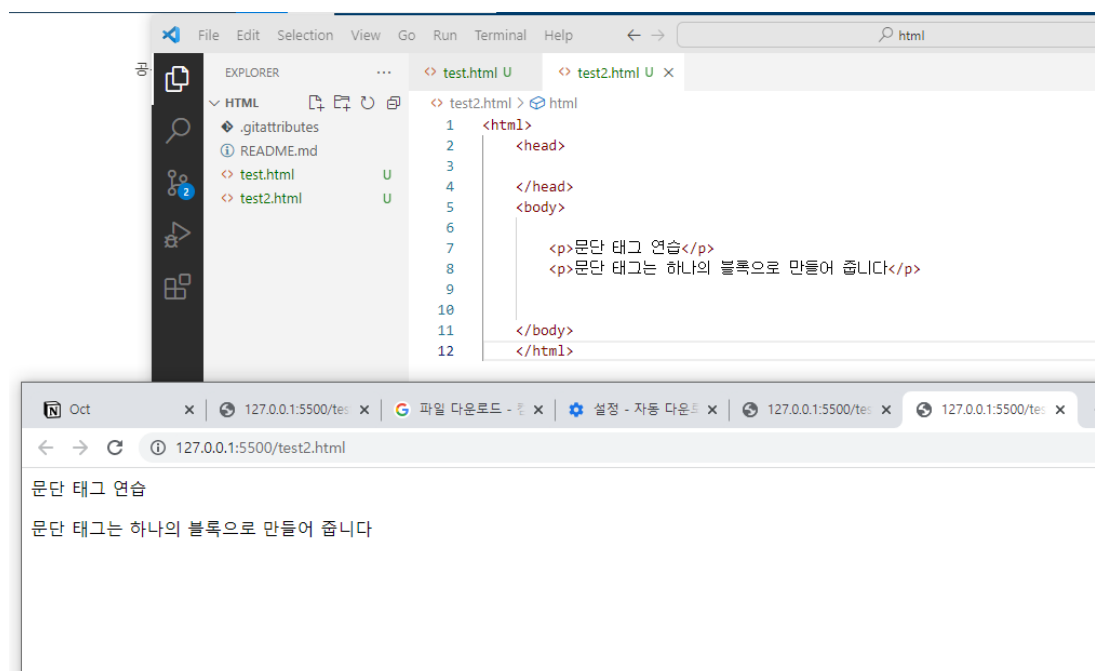
중요도 뿐이 아니라 크기로도 사용됨 h1이 가장 큼



2. Paragraphs 태그

`<p></p>`

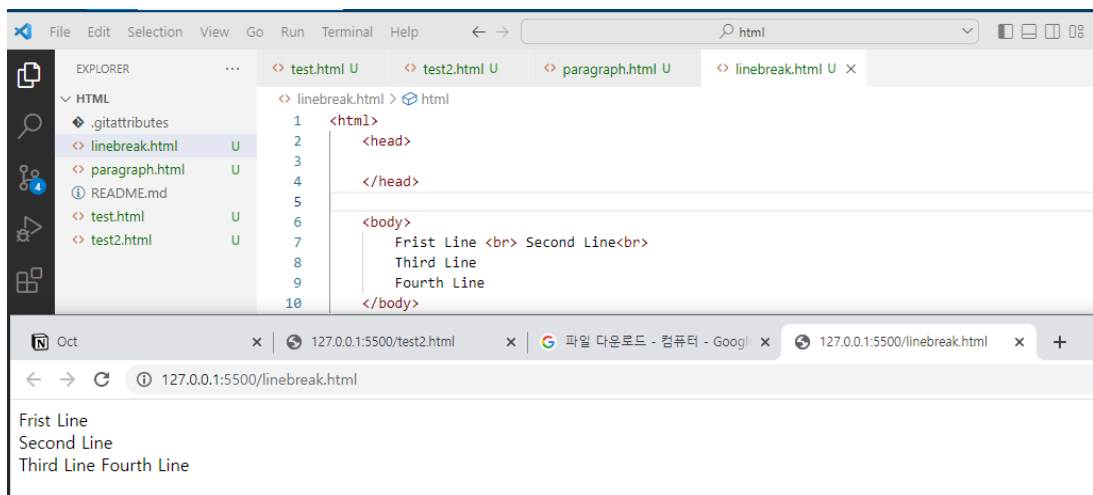
문단의 약자 하나의 문단을 만들때 사용



<p></p> 를 기준으로 위아래로 여분의 공백을 함께 표시

<p></p> 마다 새로운 줄에서 시작하며 사이에는 약간의 공백 존재함

3. Line Break



 태그는 break를 뜻하며 정확히는 Line Break

html 상에서는 줄을 바꾸어도 실제로 줄바꿈이 일어나지 않으며

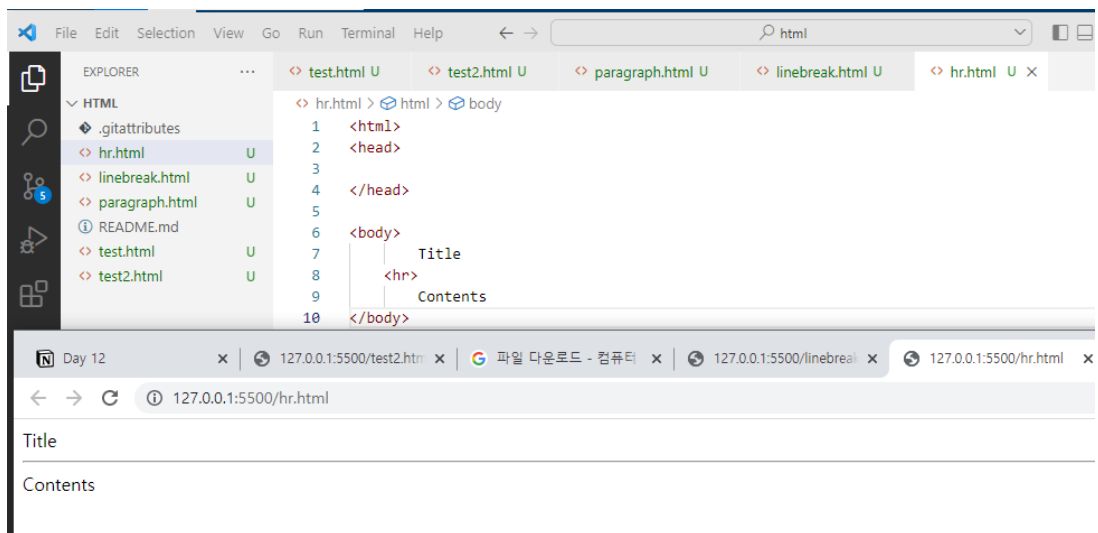
 태그를 사용하여 줄을 바꿀수 있음

4. hr 태그 Horizontal Rules

<hr> 태그는
태그와 같이 시작만으로 사용하는 empty 태그

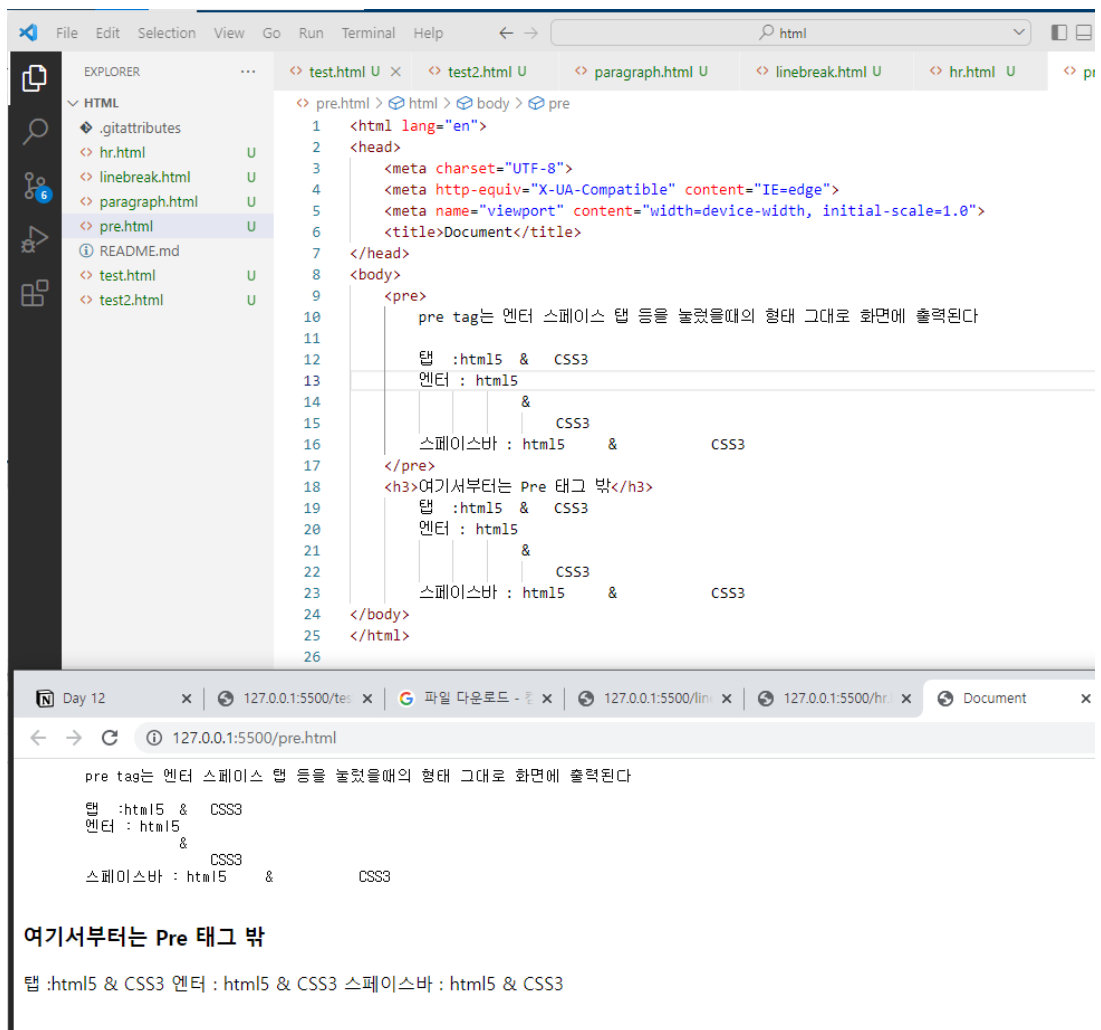
기능은 단순히 웹페이지에 수평선을 그려줌

웹페이지에서 내용을 의미적으로 분리할때 주로 사용



5. preformatted Text 태그

<pre></pre> 태그로 감싼 문장은 입력한 형태 그대로 웹페이지에 표현됨



6. Text Formatting 태그

`` bold : 글자를 굵게

`` 글자를 굵게 + 중요하다

`<i>` italic 기울임꼴

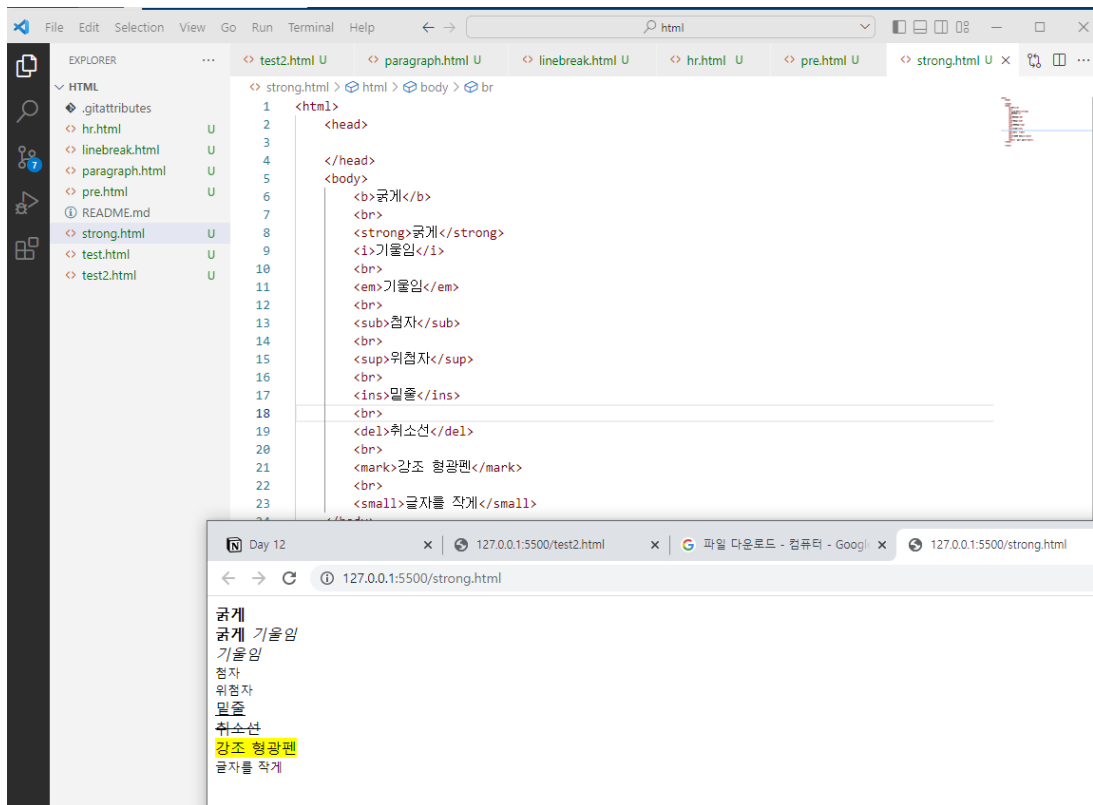
``기울임꼴 + 중요하다

`<ins>` 밑줄

`` 취소선

`<mark>` 강조 형광펜

`<small>` 글자를 작게



인용(Quotation)

`<q></q>` 짧은 문장을 인용표시할때 쓰이는 태그

브라우저에서는 큰따옴표로 나타남

`<blockquote></blockquote>` 긴 문장을 인용표시할때 쓰이는 태그

브라우저에서는 들여쓰기로 나타남

`<abbr></abbr>` abbreviation 태그는 마우스를 올리면 풍선 도움말처럼 표시 해줌

7. Comment 주석

- 브라우저에는 나타나지 않는 개발자만 보는 기능

```

16
17     <p>abbr tag</p>
18     <abbr title="제목">HTML</abbr>
19     입니다
20
21
22     <!-- 주석 단축키는 Ctrl + / -->
23 </body>
24 </html>

```

8. 링크

하이퍼링크(HyperLink 글자위로 마우스를 올리면 커서의 모양이 바뀌고 클릭을 하면 해당 링크 문서로 이동하게됨

 표시할 글자

Option

_self : 기본값, 디폴트값 현재 창에서 바로 보여줌

_blank : 가장 많이 사용되는 옵션 현재창은 놔두고 새로운 창을 열어 거기서 보여줌

_parent, _top 등의 옵션도 있음

```

<> Link.html > html > head > meta
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="https://www.naver.com" target="_self">네이버 페이지로 이동</a>
10     <br><br><br>
11     <a href="https://www.daum.net" target="_blank">다음 페이지로 이동</a>
12 </body>
13 </html>

```

상속 Inheritance

- 특정 클래스를 상속 받게 되면 그 클래스의 자산(매서드, 변수)을 그대로 가져다 사용 할 수 있음
- 다중상속은 지원하지 않음
- 일반적인 경우 자식 클래스는 부모클래스의 모든 자산에 자신의 자산을 가지게 되므로 더 많은 기능을 가지게 됨

Animal 클래스 생성 (부모)

```
public class Animal {
```

```
    String name;  
  
    void setName(String name) {  
        this.name = name;  
    }  
}
```

```
}
```

Dog 클래스 생성 (자식)

```
public class Dog extends Animal {
```

```
    void sleep () {  
        System.out.println(this.name+"ZZzzz");  
    }  
}
```

```
}
```

메인 클래스

```
public class Sample {
```

```
    public static void main(String[] args) {
```

```

Dog dog = new Dog();
dog.setName("poppy");

System.out.println(dog.name);
dog.sleep();
}

```

```

}

```

The screenshot shows an IDE with several tabs: MethodTypeEx..., module-info..., Animal.java, Dog.java, Sample.java, and Housedog.java. The Sample.java tab is active, showing the following code:

```

1 package inExam;
2
3 public class Sample {
4
5     public static void main(String[] args) {
6
7         Dog dog = new Dog();
8         dog.setName("poppy");
9
10        System.out.println(dog.name);
11        dog.sleep();
12
13    }
14 }
15
16 }
17

```

Below the code editor, the console window is visible, showing the output of the program:

```

<terminated> Sample (1) [Java Application] C:\Program Files\Java\jdk-17.0.4\bin\javaw.exe (2022. 10. 17. 오후 3:03:10 - 오후 3:03:10)
poppy
poppyZZzzz

```

Dog 클래스는 Animal 클래스를 상속 했기에 Dog은 Animal 의 하위 개념이라고 볼수 있음

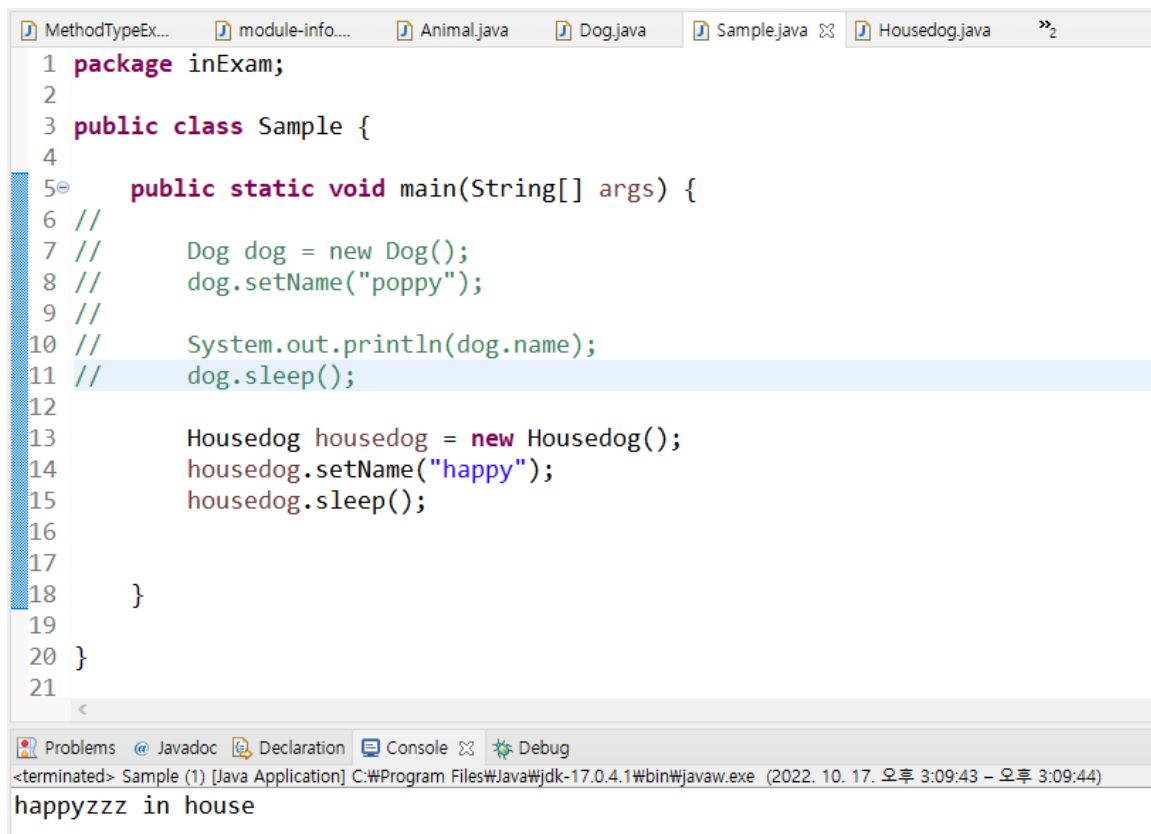
이 경우 Dog은 Animal에 포함되기 때문에 '개는 동물이다.'라고 표현가능하지만

“동물은 개다” 는 표현할수 없음

```
public class Housedog (손자) extends Dog{
```

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

```
}
```



```
1 package inExam;  
2  
3 public class Sample {  
4  
5     public static void main(String[] args) {  
6         //  
7         Dog dog = new Dog();  
8         dog.setName("poppy");  
9         //  
10        System.out.println(dog.name);  
11        dog.sleep();  
12  
13        Housedog housedog = new Housedog();  
14        housedog.setName("happy");  
15        housedog.sleep();  
16  
17    }  
18 }  
19  
20 }  
21
```

Problems @ Javadoc Declaration Console Debug
<terminated> Sample (1) [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (2022. 10. 17. 오후 3:09:43 - 오후 3:09:44)
happyzzz in house

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

```
public class Housedog extends Dog{
```

```

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");

}

}

```

The screenshot shows an IDE window with the file `*Housedog.java` open. The code defines a package `inExam` and a public class `Housedog` that extends `Dog`. Inside the class, two `sleep` methods are implemented: a no-argument `sleep()` method and a `sleep(int hour)` method. Both methods use `System.out.println` to print the dog's name and sleep status. The IDE interface includes a tab bar at the top with files like `MethodTypeEx...`, `module-info...`, `Animal.java`, `Dog.java`, `Sample.java`, and `*Housedog.java`. A line number margin on the left shows lines 1 through 19.

```

1 package inExam;
2
3 public class Housedog extends Dog{
4
5     void sleep(){
6
7         System.out.println(this.name+"zzz in house");
8
9     }
10
11
12     void sleep(int hour){
13
14         System.out.println(this.name+" zzz in house for "+hour+"hours");
15
16     }
17
18 }
19

```

- 동일한 이름의 매서드 가 여러개 있을때 인수의 종류 또는 갯수로 실행되는 내용을 바꾸고자 할때는 매서드 오버로딩을 이용함 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
```

<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
happy zzz in house for 3hours

생성자 Constructor

- 생성자는 클래스명과 동일해야 함
- 생성자는 리턴을 정의하지 않음

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

<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
happy zzz in house for 3hours

기본 생성자(Default Constructor)

사실은 클래스에는 반드시 생성자가 존재해야만한다.

개발자가 클래스를 만들때 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
```

<terminated> Sample (1) [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (2022)

happy
yorkshire

```
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        }
```

Quiz

[출력]

- 소나타 모델명 :

- 소나타 사양:
- 소나타 마일리지 : 마일리지는 10 Mile입니다.

자동차 클래스 (부모)

public class Car {

```
private String name;
private int width;
private int height;
private int length;
private double x;
private double y;
private double fuel;

public Car(String name, int width, int height, int length, double x, double y, double fuel) {
    super();
    this.name = name;
    this.width = width;
    this.height = height;
    this.length = length;
    this.x = x;
    this.y = y;
    this.fuel = fuel;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public int getWidth() {
    return width;
}

public void setWidth(int width) {
    this.width = width;
}

public int getHeight() {
    return height;
}

public void setHeight(int height) {
    this.height = height;
}
```

```

    public int getLength() {
        return length;
    }

    public void setLength(int length) {
        this.length = length;
    }

    public double getX() {
        return x;
    }

    public void setX(double x) {
        this.x = x;
    }

    public double getY() {
        return y;
    }

    public void setY(double y) {
        this.y = y;
    }

    public double getFuel() {
        return fuel;
    }

    public void setFuel(double fuel) {
        this.fuel = fuel;
    }

    public void PutSpec() {
        System.out.println("이름 : "+name);
        System.out.println("전폭 : "+width);
        System.out.println("전고 : "+height);
        System.out.println("전장 : "+length);
    }
}

```

소나타 클래스(자식)

```
public class Sonata extends Car {
```

```

    private double totalMileage;

    public Sonata(String name, int width, int height, int length, double x, double y,
        double fuel) {
        super(name, width, height, length, x, y, fuel);
    }
}

```

```

        this.totalMileage=0.0;
    }

    public double getTotalMileage() {
        return totalMileage;
    }

    public void setTotalMileage(double totalMileage) {
        this.totalMileage=totalMileage;
    }
}

```

public class Quiz {

```

    public static void main(String[] args) {

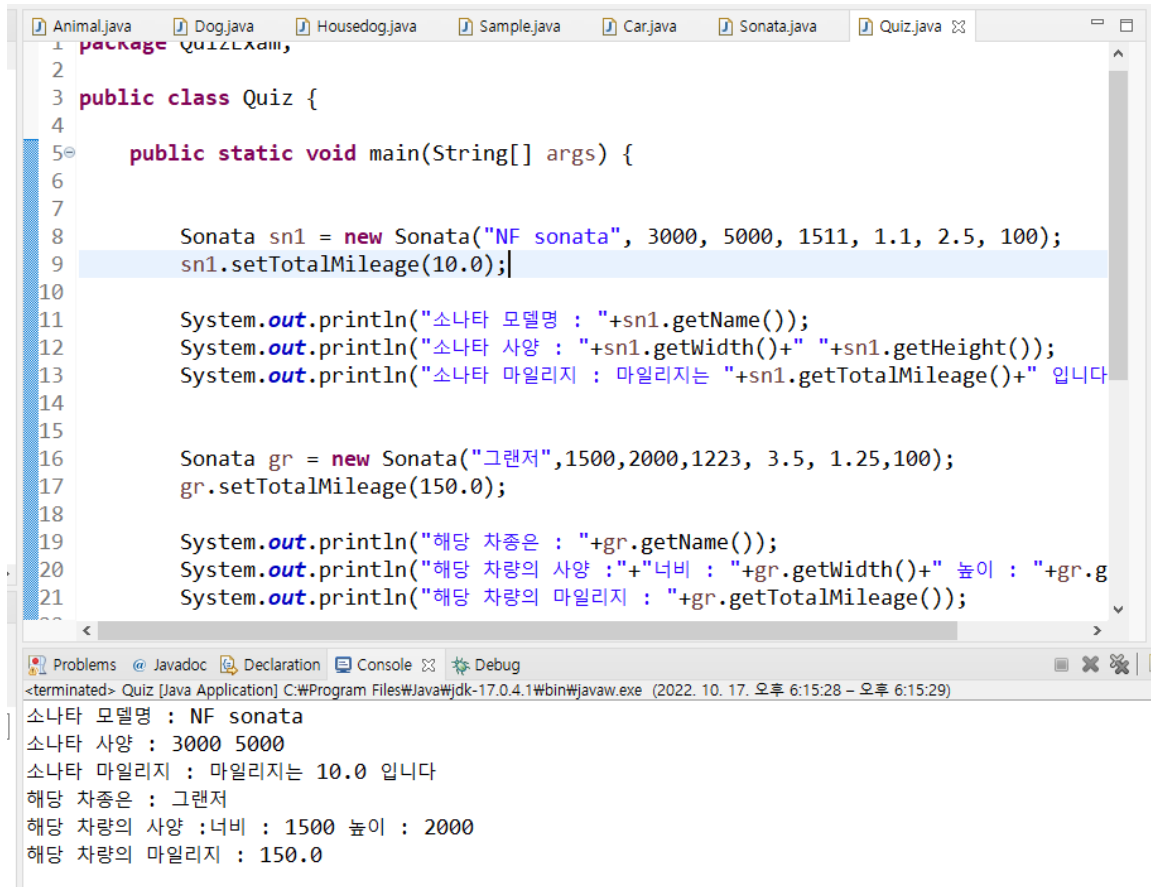
        Sonata sn1 = new Sonata("NF sonata", 3000, 5000, 1511, 1.1, 2.5, 100);
        sn1.setTotalMileage(10.0);

        System.out.println("소나타 모델명 : "+sn1.getName());
        System.out.println("소나타 사양 : "+sn1.getWidth()+" "+sn1.getHeight());
        System.out.println("소나타 마일리지 : 마일리지는 "+sn1.getTotalMileage()+" 입니다");

        Sonata gr = new Sonata("그랜저",1500,2000,1223, 3.5, 1.25,100);
        gr.setTotalMileage(150.0);

        System.out.println("해당 차종은 : "+gr.getName());
        System.out.println("해당 차량의 사양 :"+"너비 : "+gr.getWidth()+" 높이 : "+gr.getHeight());
        System.out.println("해당 차량의 마일리지 : "+gr.getTotalMileage());
    }
}

```



```
1 package QuizLab;
2
3 public class Quiz {
4
5     public static void main(String[] args) {
6
7         Sonata sn1 = new Sonata("NF sonata", 3000, 5000, 1511, 1.1, 2.5, 100);
8         sn1.setTotalMileage(10.0);
9
10
11         System.out.println("소나타 모델명 : "+sn1.getName());
12         System.out.println("소나타 사양 : "+sn1.getWidth()+" "+sn1.getHeight());
13         System.out.println("소나타 마일리지 : 마일리지는 "+sn1.getTotalMileage()+" 입니다");
14
15
16         Sonata gr = new Sonata("그랜저",1500,2000,1223, 3.5, 1.25,100);
17         gr.setTotalMileage(150.0);
18
19         System.out.println("해당 차종은 : "+gr.getName());
20         System.out.println("해당 차량의 사양 : "+gr.getWidth()+" 너비 : "+gr.getHeight());
21         System.out.println("해당 차량의 마일리지 : "+gr.getTotalMileage());
22     }
23 }
```

<terminated> Quiz [Java Application] C:\Program Files\Java\jdk-17.0.4\bin\javaw.exe (2022. 10. 17. 오후 6:15:28 - 오후 6:15:29)

소나타 모델명 : NF sonata
소나타 사양 : 3000 5000
소나타 마일리지 : 마일리지는 10.0 입니다
해당 차종은 : 그랜저
해당 차량의 사양 ;너비 : 1500 높이 : 2000
해당 차량의 마일리지 : 150.0