

1. 생성자 생성 시

(1) 클래스랑 동일한 이름

(2) 반환타입 없음

*접근제한자는 주로 public 많이 사용

ex) class Cat

```
{  
    public Cat(){  
    }  
}
```

2. 기본생성자

1) class Cat{

}

2) class Cat{

public Cat(){}

}

1)번 코드를 만들고 컴파일하면

2)번 코드와 같이 매개변수 없고, 내용이 없는 기본생성자를 컴파일러가 만들어준다.

(단, 프로그래머가 만든 생성자가 있으면 자동 생성 안됨)

1)번 코드만을 만들어도

Cat c = new Cat()

이 되는 이유...

3. 생성자의 오버로딩

```
class Cat{  
    private int weight;  
    public Cat( ){ this.weight = 2;}  
    public Cat(int w){ this.weight = w;}  
}
```

위와 같은 코드에서 아래 문장이 정상 동작

```
Cat c1 = new Cat(); //c1.weight == 2
```

```
Cat c2 = new Cat(5); //c2.weight == 5
```

* **this** -> **instance** 자기자신

* **this()** --> 자신의 생성자를 호출할 때 사용

4. 생성자의 자신의 다른 생성자 호출 1

(아래는 잘못된 코드)

```
class Cat{  
    private int weight;  
    private Cat(){ this.weight = 2;}  
    public Cat(int w){  
        if(w <=0) this();//나의 생성자  
        else this.weight = w;  
    }  
}
```

```
class Cat  
{  
    private int weight;  
    private Cat()  
    {  
        this.weight = 2;  
    }  
    public Cat(int w) : this()  
    {  
        if (w >= 0)  
            this.weight = w;  
    }  
}
```

Cat c1 = new Cat(-2); //c1.weight == 2

Cat c2 = new Cat(5); //c1.weight == 5

5. 생성자의 자신의 다른 생성자 호출 2

```
class Cat{  
    private string name; private int weight;  
  
    public Cat(string name){  
        this.name = name;  
    }  
    public void Cat(string name, int weight) : this(name)  
    {  
        this.weight = weight;  
    }  
}
```

```
Cat c1 = new Cat("나비"); // c1.name == "나비"  
                        // c1.weight == 0  
Cat c2 = new Cat("쫘",1); // c2.name == "쫘"  
                        c2.weight == 1
```

6. 생성자의 다른 메소드 호출

```
class Cat{  
    private string name;  
    private int    weight;  
  
    public Cat(string n, int weight){  
        this.SetName(n);  
        this.SetWeight(w);  
    }  
    private void SetName(n){  
        this.name = n;  
    }  
    private void SetWeight(w){  
        this.weight = w;  
    }  
}
```

생성자를 호출할 수 있는 것 : 생성자

메소드를 호출할 수 있는 것 : 생성자/메소드

* this-->this.변수 / this.메소드() / this()

