

Initialize And Constructor

Initialize (초기화)

어떤 일을 시작하기 전에 준비를 하게 되는데 이것을 다른 말로 초기화라고 한다.

```
public static void main(String[] args) {  
    Calculator c1 = new Calculator();  
    c1.setOprands(10, 20);  
    c1.sum();  
    c1.avg();  
}
```

sum, avg 메소드를 사용하려면 setOprand 메소드를 이용해 인스턴스멤버 변수를 세팅해 주어야한다.

Constructor (생성자)

- 인스턴스를 생성할 때 객체를 초기화를 할 수 있는 기능, **Instance를 생성할 수 있는 방법**
- **인스턴스 생성 과정의 일부로써 자동으로 클래스와 똑같은 이름을 가진 생성자가 먼저 실행되도록 약속되어 있다.**

Constructor (생성자) 특징

- i. 클래스와 이름이 동일한 메소드
- ii. 반환형이 선언되어 있지 않으면서, 반환하지 않는 메소드
- iii. **final 멤버 상수를 초기화 할 수 있다. (생성자는 인스턴스 생성 시 딱 한번만 호출되므로)**

```
class Calculator {  
    final int left;  
    final int right;
```

//Constructor Method

```
public Calculator(int left, int right) {  
    this.left = left;  
    this.right = right;  
    System.out.println("생성자 호출!");  
}
```

```
public void sum() {  
    System.out.println(this.left + this.right);  
}
```

```
public void avg() {  
    System.out.println((this.left + this.right) / 2);  
}
```

```
}
```

```
class CalculatorDemo{  
    public static void main(String[] args) {  
        Calculator c1 = new Calculator(10, 20);
```

```
new Calculator(10, 20)  
→ new Calculator + Calculator(10,20);  
→ 인스턴스 생성 + 생성자 호출을 동시에 중복해서 합친 표현
```

```
c1.sum();  
c1.avg();
```

```
}
```

```
}
```

Default Constructor (default 생성자)

생성자가 없어도 인스턴스 생성이 가능한 이유는 클래스에 생성자가 정의되어 있지 않아도

Java가 자동으로 default 생성자를 만들어 주기 때문이다.

```
Calculator c1 = new Calculator();
```

```
public Calculator() {  
    // 텅 비어있다.  
}
```
