

# 생성자(Constructor)

## ▶ 생성자 (Constructor)

### ✓ 생성자란 ?

객체가 new 연산자를 통해 Heap 메모리 영역에 할당될 때  
객체 안에서 만들어지는 **필드 초기화 + 생성 시 필요한 기능 수행**

생성자는 **일종의 메소드**로 전달된 초기값을 받아서 객체의 필드에 기록

### ✓ 생성자 규칙

생성자의 선언은 메소드 선언과 유사하나 반환 값이 없으며  
**생성자명을 클래스명과 똑같이 지정해주어야 함**

# ▶ 생성자 (Constructor)

## ✓ 생성자 표현식

```
[접근제한자] [예약어] class 클래스명 {  
    [접근제한자] 클래스명() { }  
    [접근제한자] 클래스명(매개변수) { (this.)필드명 = 매개변수; }  
}  
  
public class Academy {  
    private int studentNo;  
    private String name;  
  
    public Academy() {} // 기본 생성자  
    public Academy(int studentNo, String name) { //매개변수 있는 생성자  
        this.studentNo = studentNo;  
        this.name = name;  
    }  
}
```

## ▶ 생성자 (Constructor)

### ✓ 기본 생성자

작성하지 않은 경우, 클래스 사용 시 **컴파일러가 자동으로 기본 생성자 생성**

### ✓ 매개변수 생성자

- 객체 생성 시 전달받은 값으로 객체를 초기화 하기 위해 사용
- **매개변수 생성자 작성 시 컴파일러가 기본 생성자를 자동으로 생성해주지 않음**
- 상속에서 사용 시 반드시 기본 생성자를 작성
- 오버로딩을 이용하여 작성

## ▶ 오버로딩

### ✓ 오버로딩이란 ?

한 클래스 내에 동일한 이름의 메소드를 여러 개 작성하는 기법

### ✓ 오버로딩 조건

- 같은 메소드 이름
- 다른 매개변수의 **개수** 또는 다른 매개변수 **타입, 순서**

# ▶ this

## ✓ this란 ?

모든 인스턴스의 메소드에 숨겨진 채 존재하는 레퍼런스로, 할당된 객체를 가리킴  
함수 실행 시 전달되는 객체의 주소를 자동으로 받음

## ✓ this 사용 예시

```
public class Academy{  
    private String name;  
    public Academy() { }  
    public Academy(String name) { this.name = name; }  
}
```

\* 위와 같이 매개변수를 가지는 생성자에서 **매개변수 명이 필드명과 같은 경우**  
매개변수의 변수명이 우선이므로 this 객체를 이용하여 대입되는 변수가 필드라는 것을 구분해줌

## ▶ this()

### ✓ this()란 ?

생성자, 같은 클래스의 다른 생성자를 호출할 때 사용, 반드시 첫 번째 줄에 선언해야 함

### ✓ this() 사용 예시

```
public class Academy{  
    private int age;  
    private String name;  
    public Academy() { this(20, "김철수"); }  
    public Academy(int age, String name) {  
        this.age = age;    this.name = name;  
    }  
}
```