

**생성자**

---

# 생성자

---

- 생성자(Constructor)는 new 연산자로 클래스로부터 객체를 생성할 때 호출되어 객체의 초기화를 담당.
- 객체의 초기화란 필드를 초기화하거나 메소드를 호출해서 객체를 사용할 준비를 하는 것!
- 객체가 성공적으로 생성되면 힙(heap) 영역에 객체가 생성되고 객체의 번지가 리턴되며, 리턴된 번지는 클래스 변수에 저장



# 기본 생성자

- 모든 클래스는 생성자가 반드시 존재
- 생성자를 하나 이상 가질 수 있음!
- 클래스에 생성자 선언을 생략한 경우 기본생성자를 자동 추가!
- 클래스의 접근제한자를 따른다 (자동 추가된 생성자일 경우)
- 명시적으로 생성한 생성자가 1개라도 있으면 컴파일러는 기본생성자를 추가하지 않음!

# 생성자 선언

- 메소드와 비슷하지만, 리턴타입이 없고 클래스 이름과 동일
- 생성자 {}블록 내부에 초기화 코드가 작성
- 초기화는 일반적으로 멤버변수의 초기화
- 매개변수는 생략 또는 여러 개를 선언
- new 연산자로 생성자 호출 시 외부에 값을 생성자 내부로 전달

# 멤버 변수 초기화

- 멤버 변수 선언시 초기화
- 생성자 {} 블록에서 초기화

```
class Korean {  
    String nation = "대한민국";  
    String name;  
    int age;  
  
    public Korean(String n, int a) {  
        name = n;  
        age = a;  
    }  
}  
  
public class Main {  
    public static void main(String args[]) {  
        Korean k1 = new Korean("차등원", 40);  
        Korean k2 = new Korean("김윤진", 30);  
    }  
}
```



# 멤버 변수 초기화(this)

- **멤버변수명**과 **매개변수** 이름을 동일하게 설정 하는 것이 국룰!
- {}에서 가까운 매개변수 우선순위가 높기 때문에 **this** 키워드로 구별 해준다!

```
class Korean {  
    String nation = "대한민국";  
    String name;  
    int age;  
  
    public Korean(String name, int age) {  
        this.name = name;  
        this.age = age;  
    }  
}
```

# 생성자 오버로딩(Overloading)

- 다양한 방법으로 객체를 생성할 수 있도록 자바는 생성자 오버로딩(Overloading)을 제공한다.
- 생성자 오버로딩이란 **매개 변수를 달리하는 생성자를 여러 개 선언** 하는 것을 의미한다.

```
Korean(){}  
Korean(String nation){}  
Korean(String nation, String name){}  
Korean(String nation, String name, int age){}
```

# 다른 생성자 호출: `this()`

- 생성자 오버로딩이 많아질 경우 생성자간 **중복 코드 발생!**
- 초기화는 하나의 생성자에만 집중적으로 작성한 후 나머지 생성자에서는 초기화 내용이 있는 생성자를 호출
- 생성자에서 다른 생성자를 호출할 때 **`this()`**사용



# 다른 생성자 호출: **this()**

```
class Car {  
    String model;  
    String color;  
    int maxSpeed;  
  
    Car(String model){  
        this.model = model;  
        this.color = "은색";  
        this.maxSpeed = 250;  
    }  
  
    Car(String model, String color){  
        this.model = model;  
        this.color = color;  
        this.maxSpeed = 250;  
    }  
  
    Car(String model, String color, int maxSpeed){  
        this.model = model;  
        this.color = color;  
        this.maxSpeed = maxSpeed;  
    }  
}
```

```
class Car {  
    String model;  
    String color;  
    int maxSpeed;  
  
    Car(String model){  
        this(model, "은색", 250);  
    }  
  
    Car(String model, String color){  
        this(model, color, 250);  
    }  
  
    Car(String model, String color, int maxSpeed){  
        this.model = model;  
        this.color = color;  
        this.maxSpeed = maxSpeed;  
    }  
}
```

```
graph LR  
    subgraph Left [Left Code Block]  
        L1[Car(String model){  
            this.model = model;  
            this.color = "은색";  
            this.maxSpeed = 250;  
        }]  
        L2[Car(String model, String color){  
            this.model = model;  
            this.color = color;  
            this.maxSpeed = 250;  
        }]  
        L3[Car(String model, String color, int maxSpeed){  
            this.model = model;  
            this.color = color;  
            this.maxSpeed = maxSpeed;  
        }]  
    end  
    subgraph Right [Right Code Block]  
        R1[Car(String model){  
            this(model, "은색", 250);  
        }]  
        R2[Car(String model, String color){  
            this(model, color, 250);  
        }]  
        R3[Car(String model, String color, int maxSpeed){  
            this.model = model;  
            this.color = color;  
            this.maxSpeed = maxSpeed;  
        }]  
    end  
    L1 --> R1  
    L2 --> R2  
    L3 --> R3
```