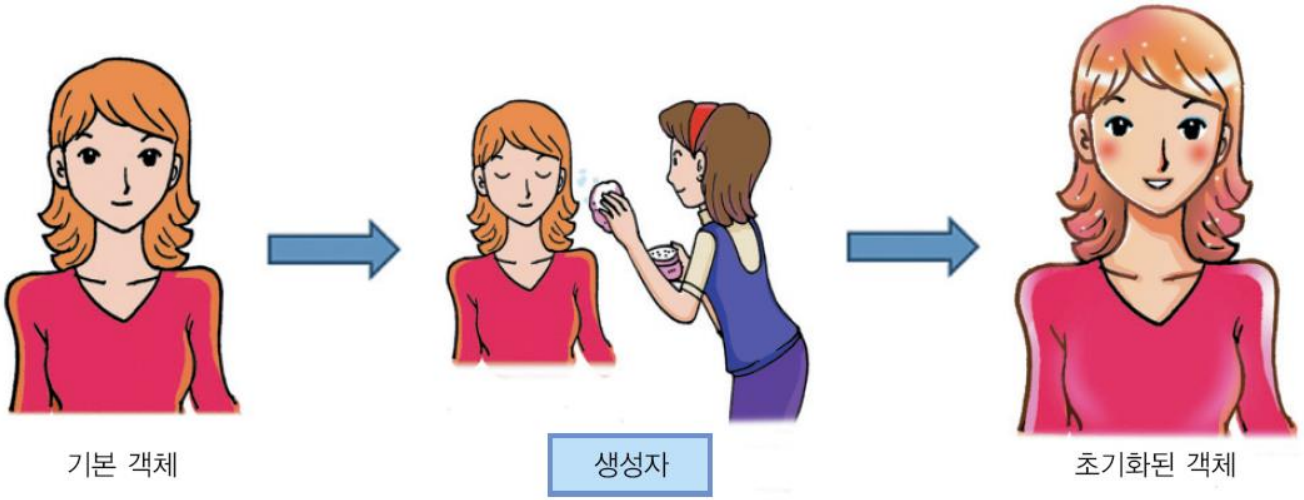
생성자 선언과 호출

▣ 생성자 선언과 호출

- new 연산자는 객체를 생성한 후, 연이어 생성자(Constructor)를 호출해서 객체를 초기화하는 역할을 한다.
- 객체 초기화란 필드 초기화를 하거나 메소드를 호출해서 객체를 사용할 준비를 하는 것을 말한다.
- 생성자가 성공적으로 실행이 끝나면 new 연산자는 객체의 주소를 리턴한다.
- 생성자는 new를 통해 객체를 생성할 때만 호출됨

```
클래스 변수 = new 클래스();
             생성자 호출
```



◎ 기본 생성자

- 모든 클래스는 생성자가 존재하며, 하나 이상을 가질 수 있다.

- 클래스에 생성자 선언이 없으면,
  컴파일러는 기본 생성자(Default Constructor)를 바이트코드 파일에 자동으로 추가시킨다.
- 클래스가 public class로 선언되면 기본 생성자도 public이 붙지만,
  클래스가 public 없이 class로만 선언되면 기본 생성자에도 public이 붙지 않는다.

- 개발자가 명시적으로 선언한 생성자가 있다면 컴파일러는 기본 생성자를 추가하지 않는다.
  개발자가 생성자를 선언하는 이유는 객체를 다양하게 초기화하기 위해서이다.

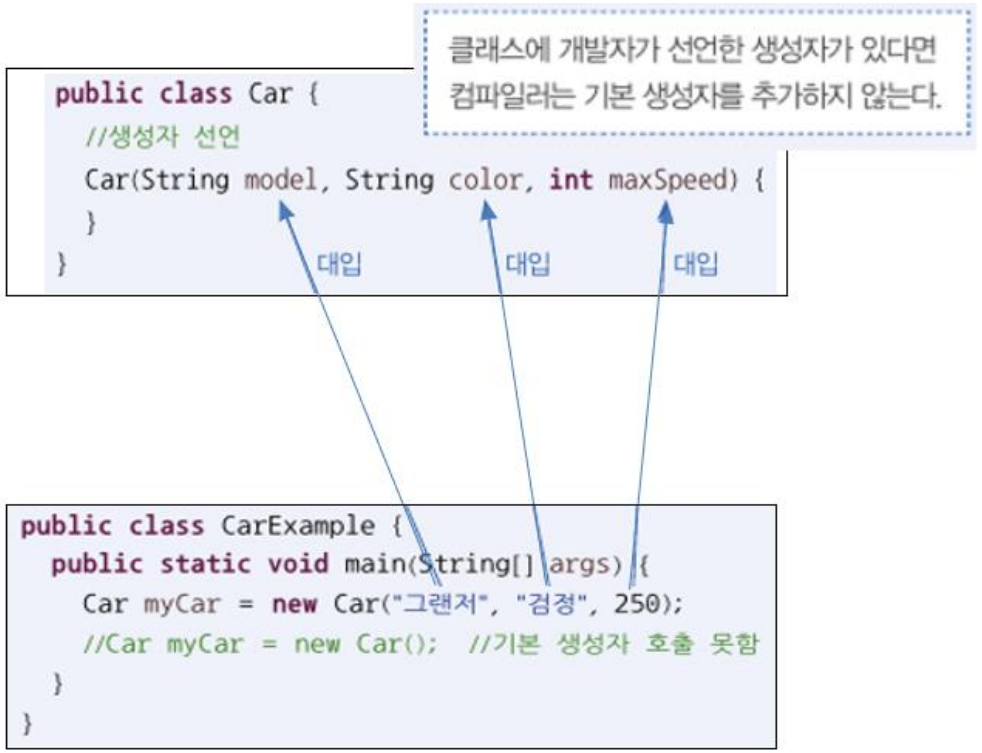

```
Car myCar = new Car();
            기본 생성자 호출
```

◎ 생성자 선언

- 생성자는 메소드와 비슷한 모양을 가지고 있으나, 리턴 타입이 없고 클래스 이름과 동일하다.

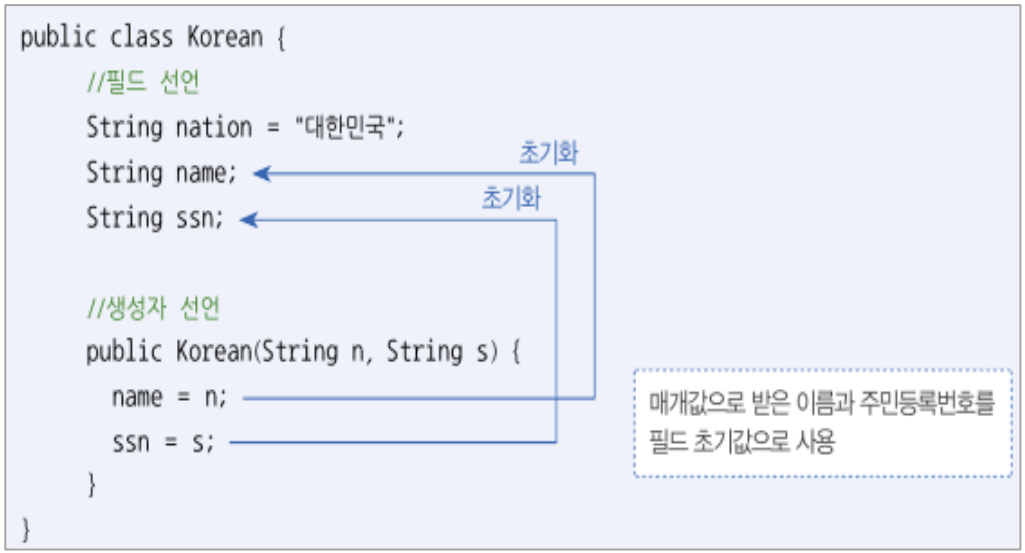

```
public class Car {
  //생성자 선언
  Car(String model, String color, int maxSpeed) { … }
}
```

```
Car myCar = new Car("그랜저", "검정", 300);
```


클래스에 개발자가 선언한 생성자가 있다면
컴파일러는 기본 생성자를 추가하지 않는다.
public class Car {
//생성자 선언
Car(String model, String color, int maxSpeed) {
}
}
대입
대입
대입
public class CarExample {
public static void main(String[] args) {
Car myCar = new Car("그랜저", "검정", 250);
//Car myCar = new Car(); //기본 생성자 호출 못함
}
}

◎ 필드 초기화

- 객체마다 동일한 값을 갖고 있다면 필드 선언 시 초기값을 대입하는 것이 좋고,
- 객체마다 다른 값을 가져야 한다면 생성자에서 필드를 초기화하는 것이 좋다.




```
Korean k1 = new Korean("박자바", "011225-1234567");
Korean k2 = new Korean("김자바", "930525-0654321");
```

◎ 생성자 오버로딩

- 생성자 오버로딩(Overloading)란 매개변수를 달리하는 생성자를 여러 개 선언하는 것을 말한다.

```
Car(String model, String color) { … }
Car(String color, String model) { … }    //오버로딩이 아님, 컴파일 에러 발생
```

```
public class Car {
  Car() { … }
  Car(String model) { … }
  Car(String model, String color) { … }
  Car(String model, String color, int maxSpeed) { … }
}
```

[생성자 오버로딩]
매개변수의 타입, 개수, 순서가 다르게 여러 개의 생성자 선언

```
Car car1 = new Car();                    → Car() {…}
Car car2 = new Car("그랜저");             → Car(String model) {…}
Car car3 = new Car("그랜저", "흰색");      → Car(String model, String color) {…}
Car car4 = new Car("그랜저", "흰색", 300); → Car(String model, String color, int maxSpeed) {…}
```

◎ 클래스내에서 또 다른 생성자 호출

- this(매개값, ...)는 생성자의 첫 줄에 작성되며, 다른 생성자를 호출하는 역할을 한다.

```
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;
}
```

중복 코드

중복 코드

중복 코드

```
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;
}
```

호출

호출

공통 초기화 코드

```
Car(String model) {
  this(model, "은색", 250);
  //추가적인 실행문
}

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