

'생성자(Constructor)'는 크게 두 가지 역할이 있습니다:

1. 인스턴스를 만들고,
2. 인스턴스의 속성(인스턴스 변수)들을 초기화시켜줍니다.

## 기본 생성자

생성자를 한 개도 정의 안 했을 경우에는 자바에서 자동으로 기본 생성자를 제공해줍니다. 우리도 사용했었죠?

```
Person p1 = new Person();
```

## 생성자 정의

생성자를 하나라도 정의하면 위의 기본 생성자는 사용할 수 없습니다.

```
public class Person {  
    String name;  
    int age;  
  
    public Person(String pName, int pAge) {  
        name = pName;  
        age = pAge;  
    }  
}
```

파라미터 두 개를 받는 생성자를 작성했습니다. 그러면 원래는 이렇게 썼던 것을:

```
public static void main(String[] args) {  
    Person p1 = new Person();  
    p1.setName("김신의");  
    p1.setAge(27);  
}
```

이렇게 바꿀 수 있습니다:

```
public static void main(String[] args) {  
    Person p1 = new Person("김신의", 27);  
}
```

그러면 **Person** 인스턴스가 생성되고, 그 인스턴스의 변수 **name** 에 "김신의" 가 지정되고, 변수 **age** 에 **27** 이 지정됩니다. 한번에 모든 인스턴스 변수의 초기값을 설정할 수 있게 된 것이죠!

## 오버로딩

파라미터의 구성을 다르게 하면 같은 이름의 메소드를 여러 개 정의할 수 있죠? 이 개념을 '메소드 오버로딩'이라고 했었는데요. 마찬가지로 생성자 오버로딩도 가능합니다.

```
public Person(String pName, int pAge) {
    name = pName;
    age = pAge;
}

public Person(String pName) {
    name = pName;
    age = 12;    // 12살을 기본 나이로 설정
}
```

파라미터 하나만 받는 생성자도 작성했습니다. 그러면 어떤 파라미터를 넘기느냐에 따라 어떤 생성자가 사용될지 정해집니다.

```
public static void main(String[] args) {
    Person p1 = new Person("김신의", 27);    // 첫 번째 생성자
    Person p2 = new Person("박윤석");        // 두 번째 생성자

    System.out.println(p1.name);
    System.out.println(p1.age);
    System.out.println(p2.name);
    System.out.println(p2.age);
}
```

```
김신의
27
박윤석
12
```



수업을 완료하셨으면 체크해주세요.



수강생 Q&A 보기



(/questions? 질문하기

assignment\_id=348&sort\_by=popular)  
(/questions/new?

assignment\_id=348&op1=%EA%B0%9D%EC%B2%B4+%EC%A7

< 이전 강의 (/assignments/337)  
생성자

다음 강의 > (/assignments/340)  
생성자 연습

