



## 3. 객체와 클래스(1)

방중교육 진행자: 김준성

# TO-DO LIST:

☒ 클래스와 객체

☒ 생성자

☒ this

☒ 객체배열

# 클래스와 객체

클래스: 객체를 만들어 내기 위한 틀(설명서)

객체: 클래스를 바탕으로 만들어진 인스턴스(실체)

*public class* Human {      **클래스**

**접근지정자**

*public* String name;

*private* String ssn;

*public* int age;

*public* String getSsn() {

~

}

}

# 클래스와 객체

객체도 레퍼런스 변수에 속하기에 new 연산자를 사용합니다.

클래스의 변수나 함수를 호출하기 위해서는  
{클래스의 레퍼런스 변수}.{멤버} 를 씁니다.

```
Human gildong = new Human();
```

 객체

```
gildong.name = "홍길동"
```

```
gildong.age = 100
```

```
String n = gildong.getSsn();
```

# 생성자

생성자는 객체가 생성될 때, 객체의 초기화를 위해 실행되는 메소드, 접근 지정자 + 클래스의 이름을 그대로 쓰면 생성자가 됩니다.

```
public class Human {  
  
    String name;  
    String ssn;  
    int age;  
  
    public Human() {  
        name = "개똥이";  
        ssn = "0000000-00000000";  
        age = 1;  
    }  
}
```

# 생성자

생성자에 매개변수를 입력하여 객체가 초기에 가져야 할 값들도 지정할 수 있습니다.

```
public class Human {
```

```
    String name;
```

```
    String ssn;
```

```
    int age;
```

참고로 이렇게 쓰면 컴파일 오류가 납니다.  
매개변수가 있는 생성자가 있으면,  
무조건 매개변수가 없는 생성자를 생성해줘야 합니다.

```
    public Human(String n, String s, int a) {
```

```
        name = n;
```

```
        ssn = s;
```

```
        age = a;
```

```
    }
```

```
}
```



**this는 객체가 자기 안에서 자기 자신을 지칭하는 말입니다.**

**이런식으로 변수를 바꿀 때 쓸 수도 있고,**

*this.age = 2*

*this.name = “우영우”*

**이런식으로 생성자를 호출할 수도 있습니다.**

*this*(“우영우”, “987654-7654321”)

**자세한 활용은 실습에서 배워봅시다.**

# 객체 배열

int와 같은 자료형 변수들과 같이, 객체도 배열로 만들 수 있습니다.

```
Human[] Attendance;
```

```
Attendance = new Human[20];
```

```
Attendance[0] = Human("홍길동", "123456-1234567", 20);
```