



# 자바 프로그래밍 기초

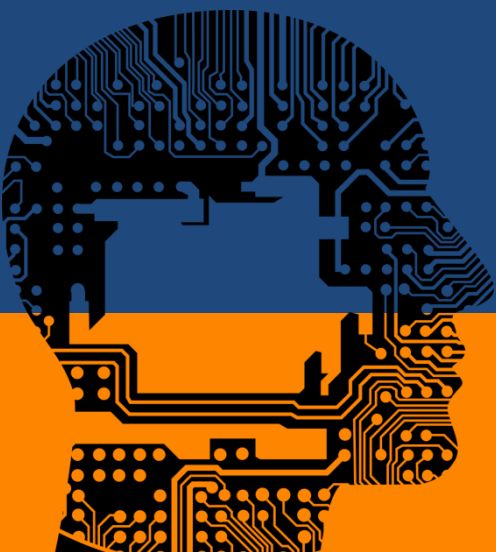
## 자바언어의 기본구조

By SoonGu Hong(Kokono)



한국IT진흥부설

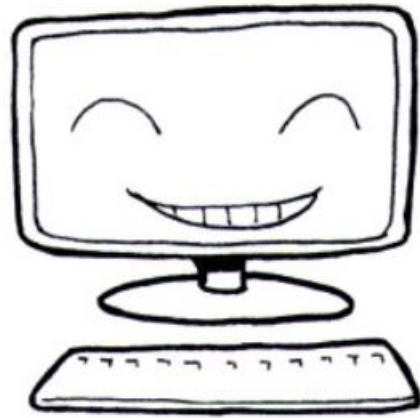
정보보호교육학원 아이섹



# 자바 프로그래밍 기초

## 1. 주석문과 기본규칙

# 1. 주석문(Comment)



public class  
~~~~~  
System.out.print()



**주석**은 코드의 이해를 높이기 위해 사용하는 **컴퓨터**가 아닌 **사람**만이 이해할 수 있는 설명문입니다.

## 1-1. 주석문(Comment)

### \* 자바에서 사용하는 주석의 종류

// 한 줄 주석

/\*

한 줄 또는 여러 줄 주석

\*/

/\*\*

document 생성용 주석

javadoc 명령으로 API 문서를 자동 생성시킬 때 사용

\*/

## 1-1. 주석문(Comment)

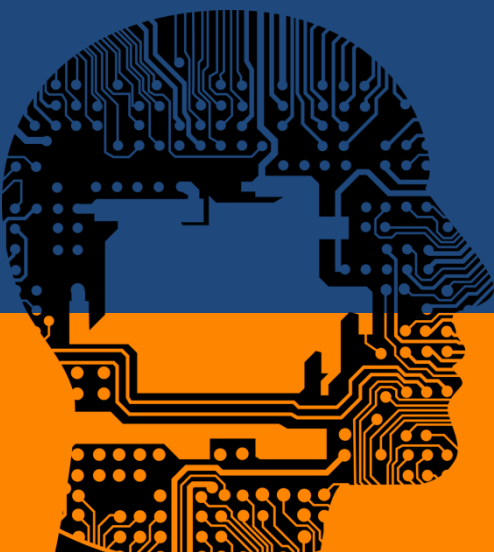
### \* 주석을 사용한 예시

```
/*  
# 클라이언트가 전송한 id값과 pw값을 가지고 DB에서 회원의 정보를  
  조회해서 불러온다음 값 비교를 통해  
  1. 아이디가 없을 경우 클라이언트측으로 문자열 "idFail" 전송  
  2. 비밀번호가 틀렸을 경우 문자열 "pwFail"전송  
  3. 로그인 성공시 문자열 "loginSuccess" 전송  
*/
```

```
System.out.println("/u:  
System.out.println("Pa  
//게시글 등록 기능  
void insert(BoardVO article);  
  
//게시글 상세 조회기능  
BoardVO getArticle(Integer boardNo);  
  
//게시물 조회수 상승 처리  
void updateViewCnt(Integer boardNo);
```

## 1-2. 기본규칙 - 세미콜론(;), 블록({}), 공백(whitespace)

- 논리적인 한 문장이 끝날 때 반드시 세미콜론(;)을 붙여 줌
- 블록은 중괄호(curly brace), 즉 "{" 와 "}"로 묶여진 부분을 말하며, 복합문을 구성하는 요소
- 종속된 문장이 다수 개 일 때 "{"로 시작하고 "}" 닫아줌
- 공백(Whitespace)은 빈칸, 탭, 빈 줄 등이 있으며, 이러한 공백은 소스 코드를 보기 좋게 하고 알아보기 쉽게 하는데 사용됨

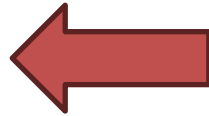


# 자바 프로그래밍 기초

## 2. 변수(variable)

## 2-1. 변수

100



우리는 현재 비행기 게임을 만들고 있습니다. 현재 게이머의 점수는 100점이지만 게임을 진행할 수록 점수는 올라가겠죠?

어떻게 하면 메모리에 있는 100이라는 숫자를 지속적으로 변경할 수 있을까요??

```
int point;  
point = 100;  
  
Enemy enemy = new Enemy();  
  
if(enemy.hp <= 0) {  
    point += 100;  
}
```



**변수를 활용!!!**



## 2-2. 변수의 선언과 초기화

### 1. 변수의 선언

- 변수를 만들기 위해서는 먼저 선언해야 합니다.

- 선언 방법

**<type> <name>;**  
ex) **int var;**



나는 메모리에 4바이트 정수를  
저장할 공간을 확보하고 그 이  
름을 var로 지정하겠어!

### 2. 변수의 초기화

- 변수를 사용하기 위해선 어떤 변수에 값을 초기화해야 합니다.

- 초기화란 변수에 값을 저장하는 것을 말합니다.(initialize)

- 초기화 방법

**<name> = <value>;**  
ex) **var = 100;**



나는 메모리에 만들어둔 var라  
는 공간에 100이라는 정수를  
저장할거야!

## 2-3. 변수의 특징

1. **변수**는 데이터가 저장되는 공간에 이름을 붙여놓은 것입니다.

2. 변수를 생성할 때는 변수에 저장되는 데이터의 **크기**와 **모양**에 따라 적당한 데이터 타입을 설정합니다.

ex) int: 4바이트 정수, String: 문자열(문장)

3. 변수는 동일한 이름으로 중복해서 선언할 수 없으며, **하나의 변수**에는 **하나의 데이터**만 저장할 수 있습니다.

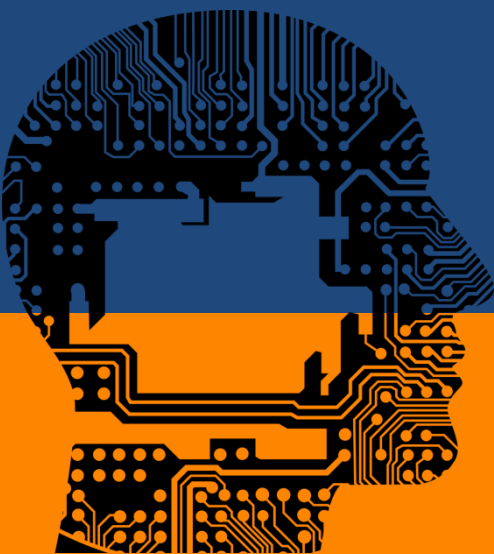
4. 변수에 저장된 값은 언제든지 **변경**할 수 있습니다.

## 2-4. 변수의 범위(Scope)

- 변수는 선언된 블록({})을 빠져나가는 순간 메모리에서 자동으로 제거됩니다.
- 따라서 선언된 블록 안에서만 변수를 참조할 수 있습니다.

```
int num1 = 10, num2 = 20;  
  
if(true) {  
    int num3 = num1 + num2;  
}  
  
System.out.println(num3);
```

- num3라는 변수는 if의 블록안에서 선언되었죠?
- 따라서 if블록이 닫히면 메모리에서 사라진답니다!
- 닫힌 이후에 num3를 프린트할 수 없는 것을 확인해보세요!



# 자바 프로그래밍 기초

## 3. 식별자(identifier)

## 3-1. 식별자란?

- **식별자**(identifier)란 개발자가 **직접 이름을 지어주는 것들을** 말합니다.
- 프로그래밍을 하면서 이미 언어에서 이름이 정해져 있는 것들을 **예약어(키워드)**라고 합니다. 이런 예약어들은 우리가 바꿀 수 없습니다.
- 변수, 함수, 메서드, 클래스, 인터페이스 등의 이름을 식별자라고 부릅니다.
- 이런 식별자들의 이름을 지어줄 때는 **몇가지 규칙**이 필요합니다.

## 3-2. 식별자 이름 규칙(필수 규칙)

1. 식별자 이름은 **중복**을 허용하지 않고 대/소문자를 구분!
2. 식별자 이름은 숫자만으로 구성하거나 **숫자로 시작**하면 안됨!
3. 식별자 이름에 **공백**을 포함할 수 없음!
4. 식별자 이름에는 **특수문자**를 사용하면 안됨(예외: \_, \$)
5. **키워드(예약어)**는 식별자 이름으로 사용 불가능!

### 3-3. 식별자 이름 규칙(관례: convention)

#### # naming convention(관례): 개발자들의 약속

- camel case: phoneNumber (Java, JavaScript)
- snake case: phone\_number (Python, SQL)

#### 1. 클래스/인터페이스 이름

- 첫글자를 **대문자**로 Pascal case 적용  
ex) MyCalculatorForHim

#### 2. 변수/ 메서드(함수) 이름

- 첫글자를 **소문자**로 Camel case 적용  
ex) String userName; // void insertBoardArticle();

## 3-4. 식별자 규칙 적용 예시

다음은 잘못된 식별자의 예로서 컴파일 시 오류가 발생합니다.

| 잘못된 예                              | 설 명                                                                                                                       |
|------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| 3test<br>@test<br>this<br>num data | 식별자는 숫자로 시작하면 안 됩니다.<br>식별자의 첫 문자에 올 수 있는 특수문자는 "_"와 "\$"뿐 입니다.<br>this는 키워드입니다. 키워드는 식별자로 사용할 수 없습니다.<br>공백을 포함할 수 없습니다. |

다음은 올바른 사용 예를 보인 것입니다.

| 바른 예                              | 설 명                                                                                                                                        |
|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| test3<br>_test<br>This<br>numData | 숫자는 식별자의 첫 문자만 아니면 어느 곳이나 올 수 있습니다.<br>"_"는 식별자의 첫 문자로 사용할 수 있습니다.<br>This는 자바에서 사용되는 this와는 다릅니다.(대/소문자를 구분합니다.)<br>대/소문자를 같이 사용할 수 있습니다. |



### 3-5. 자바의 예약어(키워드)

|                       |                      |                   |                        |              |
|-----------------------|----------------------|-------------------|------------------------|--------------|
| abstract              | continue             | for               | new                    | switch       |
| assert <sup>***</sup> | default              | goto <sup>*</sup> | package                | synchronized |
| boolean               | do                   | if                | private                | this         |
| break                 | double               | implements        | protected              | throw        |
| byte                  | else                 | import            | public                 | throws       |
| case                  | enum <sup>****</sup> | instanceof        | return                 | transient    |
| catch                 | extends              | int               | short                  | try          |
| char                  | final                | interface         | static                 | void         |
| class                 | finally              | long              | strictfp <sup>**</sup> | volatile     |
| const <sup>*</sup>    | float                | native            | super                  | while        |

\* goto와 const는 키워드로 정의되어 있지만 사용하지 않습니다.

\*\* strictfp는 1.2 버전에 추가된 키워드입니다.

\*\*\* assert는 1.4 버전에 추가된 키워드입니다.

\*\*\*\* enum은 5.0 버전에 추가된 키워드입니다.

**감사합니다**  
**THANK YOU**