

<p>1. 자바의 특징</p> <p>가. 자바는 1995년 5월 23일에 미국 선 마이크로 시스템사에서 제임스 고슬링과 아서 벤 호프와 같은 썬 개발자에 의해 개발된 언어이다.</p> <p>나. 자바는 <u>운영체제에 독립적인 컴파일 언어</u>이다.</p> <p>다. 최초 작성한 <u>자바 언어는 파일명.java로 저장</u>한다. 이 언어를 <u>원시 언어</u>라 한다.</p> <p>이 원시언어를 자바 컴파일 명령어인 <u>javac.exe</u> 에 의해서 다음과 같이 컴파일 한다.</p> <p style="padding-left: 40px;">javac 파일명.java</p> <p>로 컴파일 하면 바이트 코드 클래스 파일이 만들어 진다. <u>파일명.class</u></p> <p>라. 컴파일 된 클래스 파일에서 .class 확장자를 떼 버리고, <u>java 클래스명</u> 만 입력하면 <u>자바 가상 머신(J V M)에 로드</u> 되어 저서 실행된다. 즉 운영 체제와 상관없이 JDK에 클래스 파일이 실행되는 것이 자바의 가장 큰 특징이다.</p> <p>마.2009년 4월 20일, 썬 마이크로시스템즈는 자사가 오라클에 인수된다는 것을 발표 했다. 이듬해 2010년 1월 27일 썬 마이크로시스템즈는 오라클에 공식 합병되었다. 썬 마이크로시스템즈가 오라클과 공식 인수 합병됨에 따라 자바에 대한 권리 및 유지보수 또한 오라클로 넘어가게 되었다.</p>

<p>2. 자바 개발 환경 설정</p> <p>가. 먼저 www.oracle.com 회사 사이트로 접속해서 자바 가상 머신 jdk를 다운 받고,운영체제에 설치한다.</p> <p>나. 그리고 jdk를 설치 한 다음 요즘은 거의 개발 툴을 <u>이클립스를 사용하기 때문</u> <u>에 과거와 같은 path 운영체제 환경설정과 CLASSPATH 자바 클래스 환경설정을 각각</u> <u>따로 설정할 필요가 없다.</u></p> <p>다. 자바 개발 툴 이클립스를 www.eclipse.org/home/index.php 로 접속해서 이클립스 압축파일을 다운받아 적절한 폴더에 압축을 풀어준다. 그러면 eclipse라는 폴더가 생기고,이 폴더 안에 eclipse.exe 이클립스 개발툴 실행 파일이 있 다.</p> <p>라. 이 실행 파일을 실행해서 워크스페이스 폴더 경로를 만든다. 이 폴더안에 자바 프로젝트를 만들고 자바 클래스를 만들어 실습하면 된다. 워크스페이스 폴더는</p>
--

이클립스가 알아서 만들어 준다.

마. 이클립스 실행전 반드시 jdk를 먼저 설치해야 한다. 그렇지 않으면 이클립스가 실행되다 에러를 발생하고 이클립스를 실행할 수 없다.

이유는 이클립스가 실행될때 jdk에 포함된 JRE를 함께 읽어들이면서 실행되기 때문이다.

3. 자바 주석문 기호

가. 주석문이란 소스에 대한 설명문으로 실행되지 않는다.

나. // 한줄 주석문 기호.

/*

한줄 이상 주석문 기호

*/

4. 자바 기본 자료형(Data Type)과 참조형 즉 레퍼런스 타입

가. 정수형 숫자형: byte(크기:1바이트), short(2바이트), int(4바이트), long(8바이트)

나. 실수형 숫자형: float(크기:4바이트), double(8바이트)

다. 논리형(true or false) : boolean(1바이트)

라. 단일 문자 : char(2바이트)

자바 기본 자료형 8개를 제외한 자료형을 레퍼런스 자료형 즉 참조형이라 한다.

가. 레퍼런스형 종류: 클래스형, 배열형, 인터페이스 형

참고)

기본 타입은 값을 저장하고, 참조 타입은 객체의 주소를 저장한다.

5. 변수와 상수

가. 변수란, 단 하나의 값을 저장할 수 있는 기억공간

나. 변수 선언 방법

타입 변수명;

예를 들면, int age;

다. 변수 초기화란 변수 선언후 변수에 최초의 값을 저장하는 것을 말한다.

예를 들면, int age=26;

라. 변수 명명 규칙

첫째, 대소문자를 구분하며, 길이에 제한이 없다.

둘째, 예약어를 변수명으로 사용할 수 없다.

셋째, 숫자로 시작해서는 안된다.
넷째, 특수문자 _와 \$만을 허용한다.

마. 상수란 변수와 마찬가지로 값을 저장할 수 있는 공간이지만 변수와 달리 **한번값을 저장하면 다른 값으로 변경할 수 없다.**

바. 변수의 타입 앞에 키워드 `final`을 붙이면 상수가 된다.
예를 들면, `final int MAX=100;`

사. 상수는 선언과 동시에 반드시 초기화를 해야 한다. 그리고 상수명은 영문 대문자로 만드는 것으로 관례화 됨.

6. Java10에서 추가된 var키워드 특징

Java에서 `var` 키워드는 Java 10부터 사용 가능해졌다. 이 키워드는 지역 변수의 타입을 컴파일러가 추론하도록 할 수 있게 해주며, 코드의 가독성을 높이는 데 도움을 준다.

java 10에서 추가된 `var`키워드는 생성자나 메서드내에서 지역(local)변수를 선언할 때 사용한다. Java의 `var` 키워드는 지역 변수 선언에만 사용할 수 있으며, 클래스나 인터페이스의 멤버 변수를 선언할 때는 사용할 수 없다. 명시적 타입을 선언하는 대신 `var`을 사용해서 컴파일시에 타입을 추론할 수 있다. 이 기능의 도입으로 복잡한 코드가 간결해진다.

자바스크립트의 `var`와는 다르게, Java의 `var`은 정적 타입 언어의 특성상 런타임이 아닌 컴파일 타임에 타입이 결정된다. 따라서 `var`로 선언된 변수는 한 번 타입이 결정되면 이후에는 변경될 수 없다.

예를 들어:

```
var number = 10; // int로 추론. 타입이 int로 결정
```

```
number = "Hello"; // 오류 발생: 타입 불일치 => 타입 변경 불가능하다.
```