



변수와 타입



- 변수에 대한 개념들을 알고, 실제 자바에서 변수를 쓰는 규칙으로 사용할 수 있다.
 - 자바에서 활용되는 여러가지 데이터 형태를 알고 적절하게 사용할 수 있다.
 - 형 변환의 개념을 알고, 묵시적 형변환, 명시적 형변환의 의미를 이해하고 활용할 수 있다.
 - 전역변수, 지역변수, 상수라는 개념을 알고 이해할 수 있다.
-



생각해봅시다:

- 숫자형에서 문자형은 어떻게 만들어 지게 되었을까?
 - 문자형 한자가 만들어지는 과정은 어떻게 될까?
 - **0/1(2진수) → 56(10진수)**
 - 특정 숫자 코드값을 문자로 연결하자!!
 - **64 → 'A'**
- 문자와 문자열과의 관계는 어떻게 될까?
 - 문자열형 데이터 **type**의 도래는 언제부터 일까?
- 형변환이 왜 필요할까?
- 변수의 선언에 규칙이 없다면 어떤 문제점이 발생하게 될까?
 - 객체가 선언규칙, 변수 선언규칙 마다 사용되는 의미를 살펴보자!!



변수란? :



변수란? :

■ Variable

- 프로그램 작업을 처리하기 위해 하나의 값을 저장할 수 있는 메모리 공간을 말한다.
- 자바에서는 다양한 타입을 저장할 수 없고, 한가지 타입만 값으로 저장 될 수 있다.

- `int num01 = "안녕하세요";(X);`

- `int num02 = 25;(O)`

■ 식별자(Identifier)

- 자바코드에서 변수, 클래스명, 메서드명으로 입력 시킨 이름을 의미한다.

- `String name01="안녕하세요";`

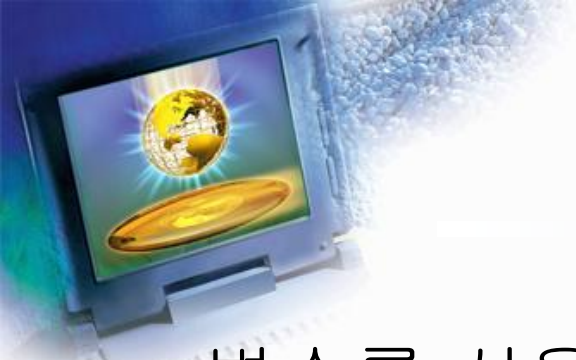
- `Class Hello{}`

- `public void setName(){}`



식별자 작성규칙 :

- 첫 문자는 반드시 영문자! 단, `_`(언더스코어), `$`(달러기호)는 특수문자이지만 사용가능.
- 대소문자를 구별하여 사용한다.
- 길이제한 없고, 공백 문자는 사용할 수 없다.
 - `userName`,
 - `$number(O)`, `#age(X)`
 - `_introName(O)`, `intro Age(X)`
- 예약어
 - 기본 데이터 type: `boolean`, `byte`, `short`, `int`, `long`, `float`, `double`
 - 접근제어자 : `private`, `protected`, `public`
 - 클래스관련내용 : `class`, `abstract`, `interface`, `implements`, `extends`
 - 객체 관련 내용 : `new`, `instanceof`, `this`, `super`, `null`
- 예약어 모르는 경우? - `@@@[01]` ex) `this01`, `null01`



변수 사용 :

- 변수를 사용하는 것은 변수를 선언하고, 값을 저장하는 것을 말한다. 정수형
- 변수 선언
 - 데이터 type 식별자;
 - `int age;` // 정수형 데이터로 `age`라는 식별자를 선언합니다. `age`는 정수형 데이터만 할당할 수 있음. (크기와 형태)
- 변수 할당
 - 선언된 식별자 = 할당할 데이터;
 - `age = 25;` ex) `age = "하이맨";` (X)
`age = 27.2;` (x)

[age]

25(할당할 data)



변수 사용 예제 :

```
package javaexp.a02_variable;

public class A01_declare_allot {

    public static void main(String[] args) {
        // 자바에서 변수 선언;
        // type 식별자
        int age;
        age = 25; // 데이터 할당..
        // 선언 + 할당(저장)
        double weight = 65.7;
        // 변수값 읽어서, 새로운 데이터에 할당.
        int after10Year=age+10;
        // 데이터 형태에 맞는 데이터를 할당해야 에러가 발생하지 않는다.
        // 데이터 형태에 맞지 않는 할당..
        //age = 25.7; 정수형 type에 실수형의 데이터를 할당시 에러발생..
        //age = "나이25세"; 정수형 type에 문자형의 데이터를 할당시 에러발생..
        System.out.println("나이는 "+age+"살이며,");
        System.out.println("몸무게는 "+weight+"kg입니다!!");
    }
}
```




데이터 type :

- 변수는 데이터 타입이 있고, 타입에 따라서 저장할 수 있는 값의 종류와 크기가 정해진다.
- 기본 타입(primitive data type)
 - 정수, 실수, 문자, 논리형을 직접 저장하는 타입
 - 정수 (소숫점이하가 없는)
 - byte(1byte), char(2byte), short(2byte), int(4byte), long(8byte)
 - 실수 (소숫점이하표현)
 - float(4byte), double(8byte)
 - 논리 (true/false)
 - boolean(1byte)
- 이스케이프(escape) 문자
 - \n (새로운라인), \t(탭 추가), \'(따옴표), \"(쌍따옴표),
 - \\(역슬래쉬), \r(리턴기능), \b(백스페이스)



데이터 **type** 예제 :

```
package javaexp.a02_variable;

public class A02_dataType {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        //정수형 data type
        //byte(1byte) -128 ~ 127
        byte age= 25; // byte데이터로 25를 할당.
        //age = 129; byte데이터 범위를 넘어서는 할당 불가.
        //short, [int], long, char
        char codeVal=66; //char 2byte로 데이터 할당..0~65535
        System.out.println("나이는 "+age);
        System.out.println("코드값"+codeVal);
        codeVal='한'; // char는 "로 한 자만 할당
        System.out.println((int)codeVal);
    }
}
```



데이터 **type** 예제 :

```
// 실수형 date type
//float(4byte) : 부호화비트(1bit) + 지수(8bits) + 가수(23bits) =32bits
//[double(8byte)] : 부호(1bit) + 지수(11bits) + 가수(52bits) = 64bits
float degree = 25.2F; // float 변수 = 데이터F;
double height= 175.2; // 실수형의 default값이기에 할당 가능
System.out.println("온도는 "+degree+"도");
System.out.println("키는 "+height+"cm");
//논리형 date type
//boolean 1byte(8bits) 표현.. true/false
boolean isGo = true;
if( isGo ){
System.out.println(" go!! go!!!");
}else{
System.out.println(" stop!! it's enough!!");
}
```



확인예제 :

- 다음과 같은 정보들을 변수를 선언하고, 아래 형식으로 출력하세요.. (@@@ 변수선언)
 - 하루 지출비용 list
 - 교통비 : @@@ 원
 - 식비 : @@@ 원
 - 다과비 : @@@ 원
 - 품목별 평균 비용 : @@@ /3.0
 - 총비용 : @@@ 원
 - 목표 비용 초과 여부 : true/false (10000 이하시)













자바란? :

- 개발 플랫폼에 따른 3가지 형태의 **java**
 - **java SE(Standard Edition) :**
 - 데스크탑에서 실행되는 응용 프로그램 개발 사용
 - **java EE(Enterprise Edition)**
 - 대규모 엔터프라이즈 기반 애플리케이션 개발
 - 서버와 클라이언트 분산에서 개발 시 활용
 - **java ME(Micro Edition)**
 - 모바일 및 임베디드(특정한 기기에 포함된 프로그램) 기반 애플리케이션 개발환경에서 활용



자바 개발 환경 구축 :

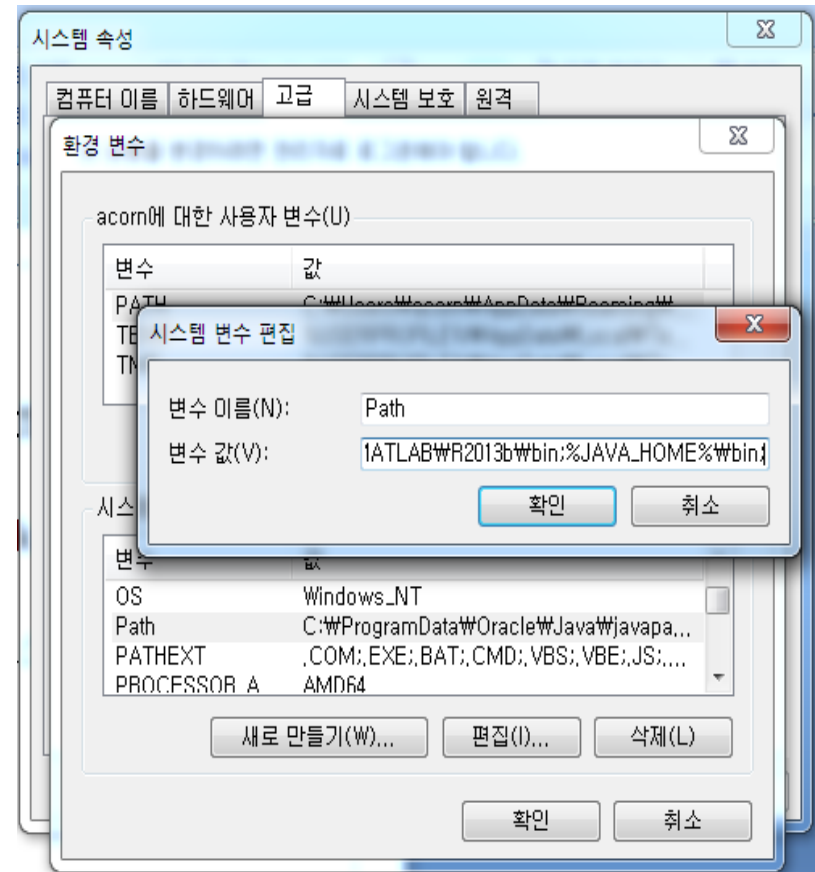
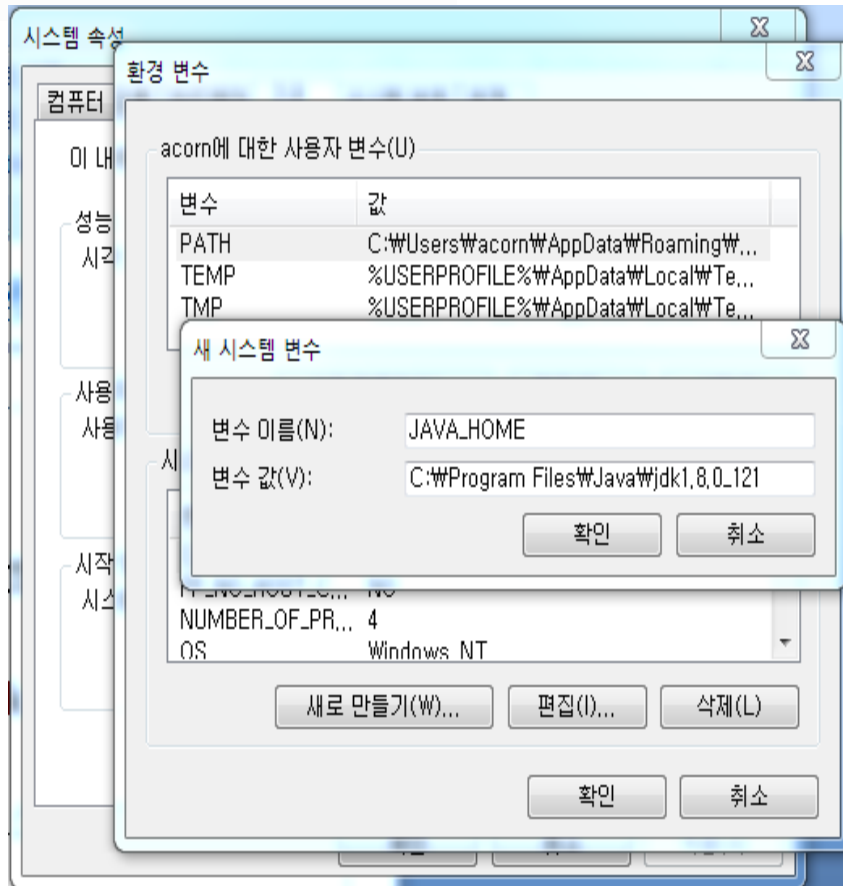
- 자바 개발 도구(JDK) 설치 : JRE+개발을 위한 도구
 - java development kit
 - jvm(자바가상 기계), 라이브러리 API(손쉽게 프로그램을 할 수 있는 클래스모음), 컴파일러 등 개발 도구가 포함.
- 자바개발 환경(JRE)
 - java runtime environment
 - jvm + 표준 클래스 라이브러리.
- 오라클(<http://www.oracle.com>) 설치 사이트
 - 운영체제에 맞는 jdk를 다운로드(64/32 CPU)



실행 환경 setting :

- java 설치
 - cmd 창에서 java 실행.
 - JAVA_HOME 환경변수로 setting하기.
 - java 설치된 곳 확인..
 - C:\Program Files\Java\jdk1.8.0_121
 - 시스템변수에 등록.
 - 윈도우 시작 → 제어판 → 시스템 및 보안 → 시스템 → 고급시스템 설정
 - 시스템 변수 Path에서 추가
 - ;%JAVA_HOME%\bin;
-

환경변수 setting



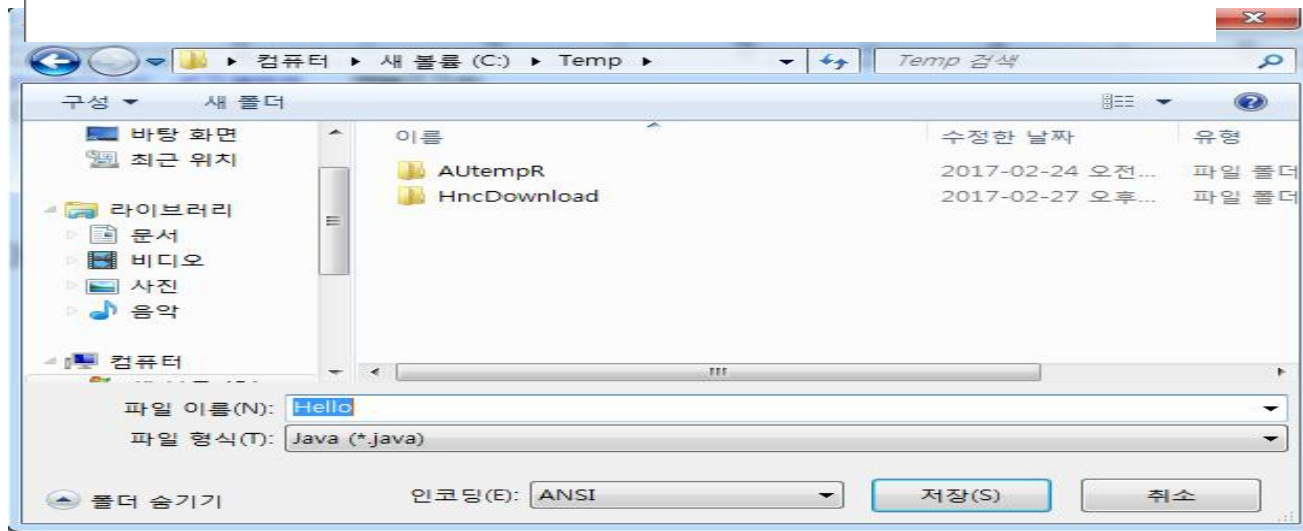


자바프로그램 개발 순서 :

- 소스 작성에서 실행
 - java 소스 파일 작성
 - Hello.java : main안에
System.out.println("안녕하세요");
 - 컴파일러(javac.exe)로 바이트 코드(.class-
기계어)생성
 - javac Hello.java → Hello.class
 - JVM 구동 명령어(java.exe)로 실행
 - java Hello

소스 작성 :

```
1 public class Hello
2 {
3     public static void main(String[] args)
4     {
5         System.out.println("Hello World!");
6     }
7 }
8
```





정리 및 남은 과제 :

- jdk와 jre이 어떤 부분에 대해서 차이가 있는가?
 - 자바에서 말하는 플랫폼 독립적이라는 말은 무엇을 의미하는가?
 - 가베지 컬렉션을 통해서 처리되는 것은 무엇인가?
 - 프로그램 설치에서 32/64는 무엇을 의미하고, jdk와 eclipse 버전에 어떤 영향을 미치는가?
 - eclipse에서 workspace는 어떤 역할을 하는가?
 - java프로그램의 실행 단계별 내용을 기술하세요!!
-



감사합니다 !
