



JAVA

GU, DA HAE

컴 퓨 터

변

자 료

식 별

상

자 료 형

구 조

수

형

자

수

변 환

J A V A

변

수

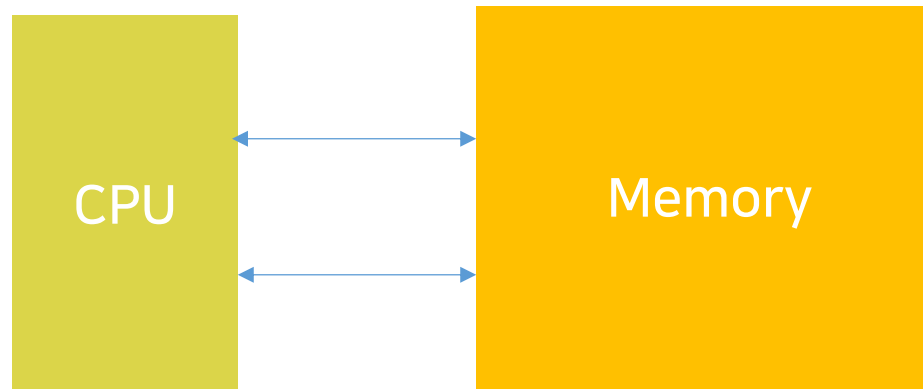
컴퓨터 구조

| CPU, 메모리

컴퓨터 구조에서 가장 중요한 부분은 **cpu(Central Process Unit)**와 **Memory(메모리)**다.

CPU는 컴퓨터에서 모든 **연산**(프로그램 실행)을 담당한다. (뇌)

메모리는 CPU에서 사용하는 값을 **저장**한다. (신경세포)



변수

| 프로그램에서 사용하는 값의 저장

프로그램에서 필요한 값을 **메모리에 저장**하여 **사용**하려면 변수가 필요하다. 변수는 다음과 같이 만들 수 있다.

```
자료형 식별자;  
자료형 식별자, 식별자;
```

```
자료형 식별자 = 값;  
자료형 식별자 = 값, 식별자 = 값;
```

이 과정을 '**변수를 선언한다**'라고 표현한다. 선언 후에는 항상 **;**을 찍어야 한다.

자료형은 사용할 메모리 **공간의 크기**, 저장할 **값의 종류**에 따라 다르다.

식별자는 변수를 구분하기 위한 **이름**이다. '**변수명**'이라고도 한다.

값은 변수에 **저장할 값**이다.

자료형

| 자바의 기본 자료형

자료형	데이터	메모리 크기(byte)	표현 가능 범위
boolean	논리값	1	true, false
char	문자	2	Unicode 문자
byte	정수	1	-128 ~ 127
short		2	-32,768 ~ 32,767
int		4	-2,147,483,648 ~ 2,147,483,647
long		8	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807
float	실수	4	$\pm(1.40 \times 10^{-45} \sim 3.40 \times 10^{38})$
double		8	$\pm(4.94 \times 10^{-324} \sim 1.79 \times 10^{308})$

int	, 가
가	double
char	, 2
boolean	(true, false)

식별자

| identifier

사용할 메모리 공간에 붙이는 ID, 혹은 이름이다.

식별자는 영어 대·소문자와 숫자, `_`, `$`로만 구성해야 한다.

식별자는 첫 번째 문자로 숫자가 올 수 없다.

식별자는 대·소문자를 구분한다.

키워드는 식별자로 사용 할 수 없다.

J A V A

예제

```
public class Ex01 {  
    public static void main(String[] args) {  
        double n1, n2, result;  
  
        n1 = 1.0000001;  
        n2 = 2.0000001;  
        result = n1 + n2;  
  
        System.out.println(result);  
    }  
}
```

예제

```
public class Ex02 {  
    public static void main(String[] args) {  
        // 어떤 코드가 더 좋은 코드일까?  
        // 1. short로 구성한 경우  
        short n1 = 11, n2 = 22;  
        short result = n1 + n2;  
  
        // 2. int로 구성한 경우  
        int m1 = 11, m2 = 22;  
        int resultM = m1 + m2;  
    }  
}
```


예제

```
public class Ex03 {  
    public static void main(String[] args) {  
        double e1 = 1.2e-3; // 지수 표기법  
        double e2 = 1.2e+3; // 지수 표기법  
  
        int n1 = 0xA0E; // 16진 정수  
        int n2 = 0752; // 8진 정수  
  
        System.out.println(e1);  
        System.out.println(e2);  
        System.out.println(n1);  
        System.out.println(n2);  
    }  
}
```

예제

```
public class Ex04 {  
    public static void main(String[] args) {  
        // java는 Unicode로 문자를 저장한다. Unicode는 문자 하나를 2바이트로 표현하는 문자체계이다.  
        char c1 = 'A'; // 10진수 : 10, 16진수 : 0x41  
        char c2 = '한'; // 10진수 : 54620, 16진수 : 0xD55C  
        char c3 = 0x3091;  
        char c4 = 0x3092;  
        System.out.println(c1);  
        System.out.println(c2);  
        System.out.println(c3);  
        System.out.println(c4);  
    }  
}
```

J A V A

예제

```
public class Ex05 {  
    public static void main(String[] args) {  
        boolean b1 = true;  
        boolean b2 = false;  
  
        System.out.println(b1);  
        System.out.println(b2);  
        System.out.println(3 < 4);  
        System.out.println(3 > 4);  
    }  
}
```

컴 퓨 터

변

자 료

식 별

상

자 료 형

구 조

수

형

자

수

변 환

J A V A

변

수

상수

프로그램에서 사용하는 값을 상수(≡ 리터럴; literal)라 부른다.

```
int n = 1 + 5;  
System.out.println(2.4 + 7.5);
```

상수도 메모리 공간에 저장된다.

상수는 이름이 없다.

이름이 없으므로 메모리에 저장된 상수는 값을 변경할 수 없다.

예제

```
public class Ex06 {  
    public static void main(String[] args) {  
        double e1 = 7.125;  
        double e2 = 7.125f; // 접미사 f : 상수를 double이 아닌 float형으로 표현할 때 사용  
  
        long n1 = 10000000000L; // 접미사 L : 상수를 int가 아닌 long형으로 표현할 때 사용  
        long n2 = 150;  
  
        System.out.println(e1);  
        System.out.println(n1);  
    }  
}
```

자료형 변환

자료형 변환은 값의 표현 방식을 바꾸는 것이다. 자바의 형 변환은 2가지 종류가 있다.

1. 자동 형 변환(묵시적 형 변환)

~~연산을 할 때 피연산자의 자료형이 일치하지 않아서 발생하는 형 변환~~

가

2. 명시적 형 변환

형 변환 연산자를 이용한 형 변환

가

자동 형 변환

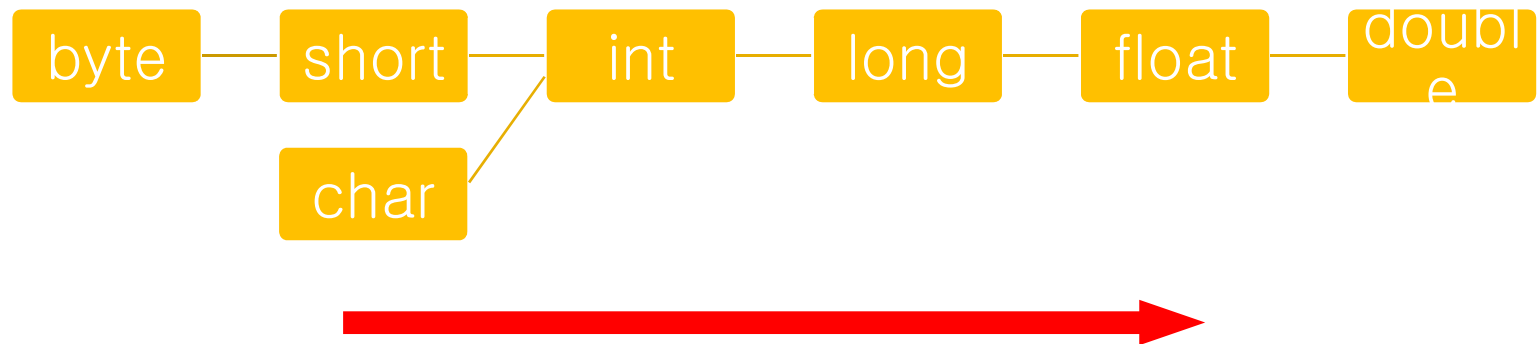
CPU는 자료형이 다른 피연산자를 연산할 수 없다.

```
00000000 00000000 00000000 00000001  
00111111 10000000 00000000 00000000
```

int형 1과 float형 1.0을 어떻게 더할까? 이처럼 자료형이 다른 두 피연산자를 연산할 때는 자동 형 변환을 통해 피연산자의 자료형을 일치시킨다.

J A V A

자동 형 변환 규칙



왼쪽에서 오른쪽 방향으로서는 자동으로 형 변환이 발생한다.

명시적 형 변환

자동 형 변환 규칙에 위배되는 상황임에도 형 변환이 필요할 경우에 사용하는 형 변환 방법이다.

```
int n = (int)3.15;
```

```
long n2 = 2147483648L;  
int n3 = (int)n2;
```

실수형 데이터를 정수형 데이터로 형 변환하는 경우에는 소수점 이하가 잘린다.
두 정수형 데이터 사이에서 형 변환 하는 경우 사위 바이트가 잘린다.

J A V A

예제

```
public class Ex07 {  
    public static void main(String[] args) {  
        char c1 = 'A';  
        char c2 = 'Z';  
  
        int n1 = c1;  
        int n2 = (int)c2;  
  
        System.out.println("문자 A의 유니코드 값 : " + n1);  
        System.out.println("문자 Z의 유니코드 값 : " + n2);  
    }  
}
```