Data Type

데이터의 단위 bit And byte

1 bit	0, 1 둘 중 하나를 담을 수 있는 데이터의 최소단위
1 byte	8bit
1 KB	1024 byte
1 MB	1024 KB
1 GB	1024 MB
1 TB	1024 GB

Data Type (자료형)

Java가 Data를 표현하는 방법이다.

정수 4의 int형 표현 : 00000000 00000000 00000000 00000100

Java Primitive Data Type (원시자료형)

정수	byte	1 byte	-128 ~ 127
	short	2 byte	-32,768 ~ 32,767
	int	4 byte	-2,147,483,648~2,147,483,647
	long	8 byte	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807
실수	float	4 byte	±(1.40129846432481707e-45 ~ 3.40282346638528860e+38)
	double	8 byte	±(4.94065645841246544e-324d ~ 1.79769313486231570e+308d)
문자	char	2 byte	모든 유니코드 문자
참,거짓	boolean	1 byte	true, false

```
class VariableDec1 {
	public static void main(String[] args) {
	double num1, num2 result;
	num1 = 1.0000001;
	num2 = 2.0000001;
	result = num1 + num2;

	System.out.println(result); // 3.0000001999999997 부동소수점오차 발생!
	}
}
```

정수 자료형

- Byte, short, int, long
- 정수를 표현하는 방법은 같으나 바이트크기에 따라 구분

? 정수 12를 어떤 자료형을 근거로 선언할까

- 일반적인 정수의 자료형은 int이다.
  - CPU 연산은 특정 Byte 연산에 포커스가 되어있다. CPU의 구조단순화로 속도가 올라간다.
    (32bit → int, 64bit → long)
  - short + short 연산상황이 되면 int + int 로 자동 형변환한다. (시간낭비)
  - 변환의 과정을 생략하기 위해서 int로 표현한다.

- **short와 bite 사용이유?**
  - Mp3 player같은 디바이스는 연산능력보다 메모리 저장공간이 중요하다.
  - 연산능력보다는 메모리의 효율성이 중요하게 여겨질 때 표현.
  - **데이터의 성격이 강하면 short와 byte로 표현**

실수 자료형

- float, double
- **float는 소수점 이하 6자리까지, double은 12자리까지 정밀도 보장**
  (그러나 누적연산 시 보장받을 수 없다.)

- 두 자료형은 모두 충분히 넓은 범위를 표현할 수 있으므로, **정밀도에 따른 자료형의 선택이 필요하다.**
  0.00001+0.00001이 0.0000000001+0.0000000001보다 **누적 연산**을 하면 소수점에 가까워지기 때문에 오차가 더 커진다.

- **일반적인 실수의 자료형은 double이다.**

- **실수 연산시에는 Java의 BigDecimal API사용**
- **실수는 == 연산을 할수 없다 -> 부동소수점오차 때문!**

실수의 e 표기법과 8, 16진수의 표현

```
class ENotation {
	public static void main(String[] args) {
	// 지수기반 표기법 e는 밑 10, e+5 → 10^5
	double e1 = 1.2e-3; // 1.2 x 10^-3
	double e2 = 1.2e+3; // 1.2 x 10^+3

	// 0x : 16진수, 0 : 8진수
	int num1 = 0xA0E; // 16진수 A0E
	int num2 = 0x752; // 16진수 752
	int num3 = 0752; // 8진수 752

	System.out.println(e1);
	System.out.println(e2);
	System.out.println(num1); // 저장되고 나면 16진수가 아니게 된다.
	System.out.println(num2);
	System.out.println(num3);
	}
}
```

문자 자료형 char

- 문자가 컴퓨터에 저장될 때에는 0과 1을 조합해서 저장한다. (인코딩과 디코딩)
- **Java에서는 2 byte 유니코드 문자Set 기반으로 문자를 표현한다.**
- 유니코드는 전세계의 문자를 표현할 수 있다. (2^16개)
- char a = 1; (X), char a ='1'; (O)

Uni Code Character, Value

문자가 저장될 때에는 **유니코드문자 → 유니코드 값**으로 변환

- char ch1 = 'A'; → char ch1 = 65; // 0x41
- char ch2 = '한'; → char ch2 = 54620; // 0xD55C

참, 거짓을 표현하기 위한 자료형 boolean

true	참이라는 현상
false	거짓이라는 현상

- 정수, 실수, 문자도 아니다.
- Java는 참, 거짓이라는 현상을 true, false 키워드를 별도의 자료형으로 인지한다.
- true, false는 독립된 하나의 데이터로 인식한다. (숫자가 아니다!)
- **boolean형은 true, false의 저장을 위한 자료형!**

```
class Boolean {
     public static void main(String[] args) {
     boolean b1 = true;
     boolean b2 = false;

     System.out.println(b1);  // true
     System.out.println(b2);  // false
     System.out.println(3<4); // true
     System.out.println(3>4); // false
     }
}
```
