



나는 정말 JAVA^를 공부한 적이 있나?

Chapter 03. 상수와 형 변환



03-1.

자료형을 기반으로 표현이 되는 상수

■ 상수를 언제 사용했었지?

✓ 표현되는 데이터는 상수 아니면 변수

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

✓ 상수와 변수의 비교

- 변수와 마찬가지로 상수도 메모리 공간에 저장이 된다.
- 다만 이름이 존재하지 않으니 값의 변경이 불가능하다.
- 상수는 존재 의미가 없어지면 바로 소멸된다.

■ 상수도 자료형을 기반으로 저장이 됩니다.

✓ 상수의 저장방식의 근거는 자료형

- int, double과 같은 자료형은 데이터 표현의 기준이다.
- 따라서 변수뿐만 아니라 상수의 데이터 저장 및 참조의 기준이다.

✓ 정수형 상수와 실수형 상수의 표현 자료형

- 정수형 상수 int형으로 표현
- 실수형 상수 double형으로 표현

■ 접미사 이야기

✓ 다음 세 문장에서 컴파일 오류가 발생하는 이유는?

```
int num1=100000000000;    // num에 저장 불가!
long num2=100000000000;   // 상수의 표현이 먼저이므로!
float num3=12.45;         // 12.45는 double형 상수
```

✓ 접미사를 이용한 상수표현방식의 변경

```
long num1=1000000000000L; // 접미사 L은 long형 상수표현을 의미
float num2=12.45F;        // 접미사 F는 float형 상수표현을 의미
```



03-2. 자료형의 변환

■ 자료형의 변환이 의미하는 것은?

✓ 자료형의 변환은 표현방법의 변환

```
int main(String[] args)
{
    short num1=10;
    short num2=20;
    short result = num1 + num2;
    . . . .
}
```

num1(10) → 00000000 00001010

num2(20) → 00000000 00010100

↓ short to int

int형 정수 10 → 00000000 00000000 00000000 00001010

int형 정수 20 → 00000000 00000000 00000000 00010100

int형 정수 1 → 00000000 00000000 00000000 00000001

↓ short to int

float형 실수 1.0 → 00111111 10000000 00000000 00000000

■ 자료형을 일치시켜야 하는 이유?

✓ 사람도 못하는데 컴퓨터가 할 수 있겠는가?

	00000000	00000000	00000000	00000001
+	00000000	00000000	00000000	00000010
<hr/>				
	00000000	00000000	00000000	00000011

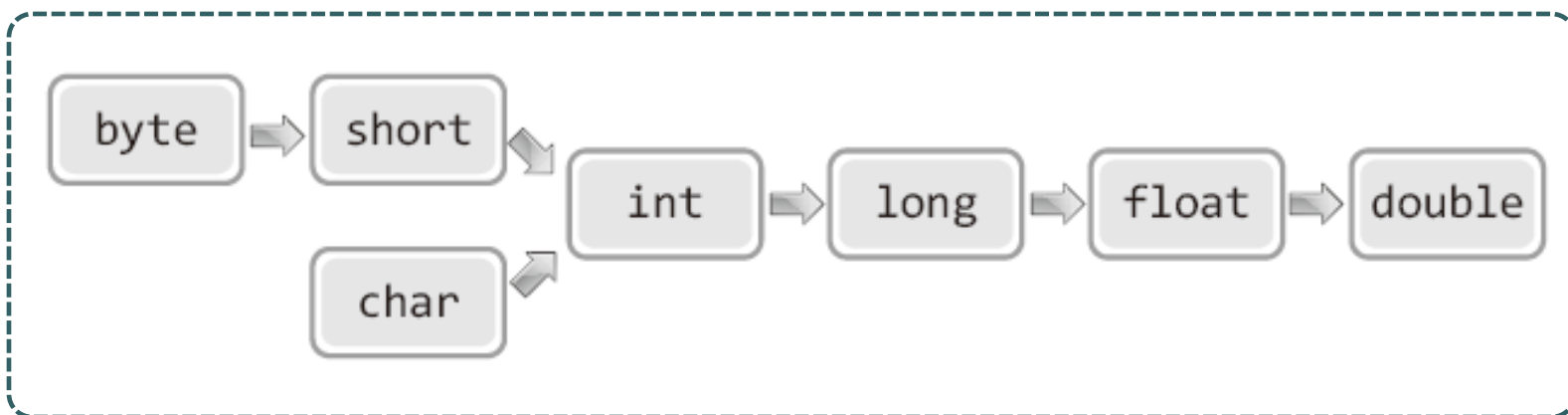
[그림 3-1 : 1 더하기 2는 3]

	00000000	00000000	00000000	00000001
+	00111111	10000000	00000000	00000000
<hr/>				
	????????	????????	????????	????????

[그림 3-2 : 1 더하기 1.0은?]

■ 자동 형 변환(Implicit Conversion)

✓ 자동 형 변환 규칙



`double num2=3.5f+12;` 12가 12f로 자동 형 변환

■ 명시적 형 변환

✓ 명시적 형 변환을 하는 이유

- 자동 형 변환 발생지점의 표시를 위해서 case 1
- 자동 형 변환의 규칙에 위배되지만 변환이 필요한 상황 case 2

case 1

```
long num1 = 2147483648L;
int num2 = (int)num1;
```

case 2

```
int num3 = 100;
long num4 = (long)num3;
```

