Chapter 02. 데이터

정수형

int

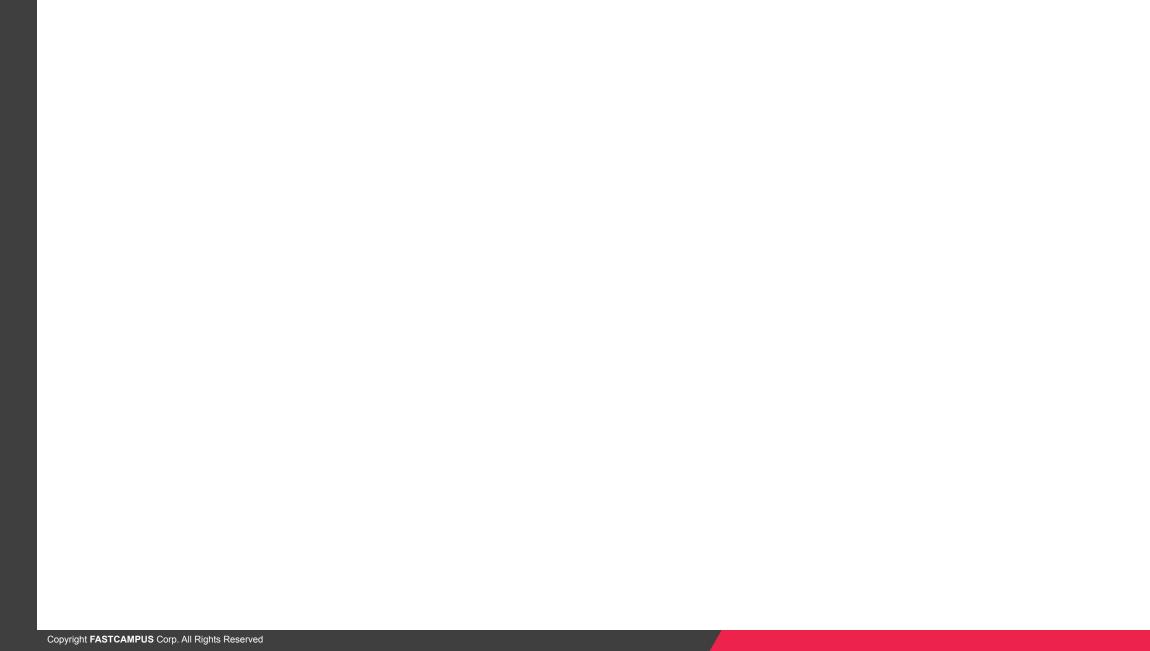
Type specifier	Equivalent type	Width	in bits	by data m	nodel	
		C++ standard	LP32	ILP32	LLP64	LP64
short	short int	at least 16	16	16	16	16
short int		10				
signed short						
signed short int						
unsigned short	unsigned short int	-				
unsigned short int						
int	int	at least 16	16	32	32	32
signed		16				
signed int						
unsigned	unsigned int					
unsigned int						



int

Type specifier	Equivalent type	Width	in bits	by data n	nodel	
		C++ standard	LP32	ILP32	LLP64	LP64
long	long int	at least 32	32	32	32	64
long int		32				
signed long						
signed long int						
unsigned long	unsigned long int					
unsigned long int						
long long	long long int (C++11)	at least 64	64	64	64	64
long long int	(С++11)	04				
signed long long						
signed long long int						
unsigned long long	unsigned long long int (C++11)					
unsigned long long int	(CTTII)					







int

32 bit systems	LP32	2/4/4	Win16 API
	ILP32	4/4/4	Win32 API Unix and Unix-like systems (Linux, Mac OS X)
64 bit systems	LLP64	4/4/8	Win64 API
	LP64	4/8/8	Unix and Unix-like systems (Linux, Mac OS X)



Binary(이진법)

$$0 = 0_{(2)}$$

$$1 = 1_{(2)}$$

$$3 = 11_{(2)}$$

$$4 = 100_{(2)}$$

$$8 = 1000_{(2)}$$

$$16 = 10000_{(2)}$$

$$32 = 100000_{(2)}$$

$$64 = 1000000_{(2)}$$



Binary(이진법)

Decima	əl	
10 ²	10 ¹	10 ⁰
1	0	0

$$100 = 1 * 10^2 + 0 * 10^1 + 0 * 10^0$$

Binary											
2 ⁶	2 ⁵	24	2 ³	2 ²	2 ¹	20					
1	1	0	0	1	0	0					

$$1100100_{(2)} = 1 * 2^6 + 1 * 2^5 + 0 * 2^4 + 0 * 2^3 + 1 * 2^2 + 0 * 2^1 + 0 * 2^0$$



unsigned(부호 없는)

uint8_t num0 = 1; // base 10

0	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---

uint16_t num1 = 020; // base 8 // 16 = 2 * 8¹ + 0 * 8⁰ // 16 = 1 * 2⁴ + 0 * 2³ + 0 * 2² + 0 * 2¹ + 0 * 2⁰

0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

uint32_t num2 = 0xff; // base 16

$$//$$
 255 = 15 * 16¹ + 15 * 16⁰,

$$//$$
 255 = 1 * 2⁷ + 1 * 2⁶ + 1 * 2⁵ + 1 * 2⁴ + 1 * 2³ + 1 * 2² + 1 + 2¹ + 1 + 2⁰ * 1

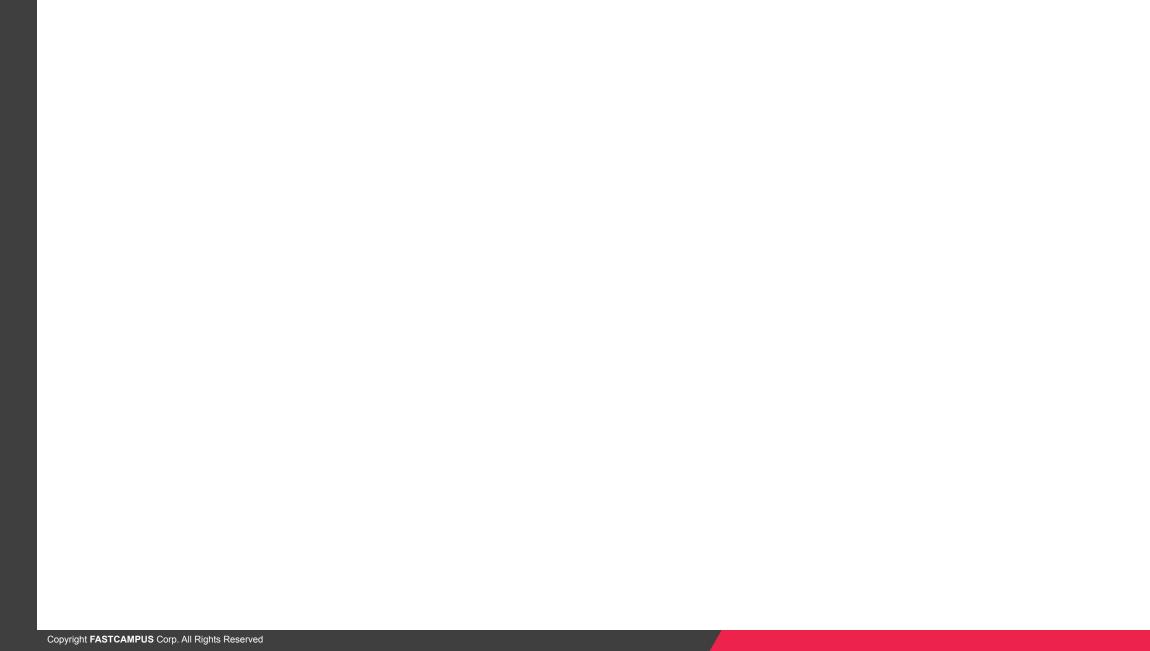
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

uint32_t num3 = 0b1010; // base 2

$$//9 = 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 0 * 2^0$$

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0







unsigned(부호 없는)

8 bit max value = 255 $255 = 2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0$

 $255 = 2^8 - 1$

1 1 1 1 1 1 1

16 bit max value = 65535

 $65535 = 2^{16} - 1$

1	1		1			1		1	1				1	1	1
_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_

32 bit max value = 4294967295

 $4294967295 = 2^{32} - 1$

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

unsigned(부호 없는)

덧셈

$$6 + 7 = 13$$

0	0	0	0	0	1	1	0	$6 = 2^2 + 2^1$
0	0	0	0	0	1	1	1	$7 = 2^2 + 2^1 + 2^0$
0	0	0	0	1	1	0	1	$13 = 2^3 + 2^2 + 2^0$

```
1

Most Significant Bit(MSB)
Sign Bit
```



Simple method

-1

1	0	0	0	0	0	0	1

-0 55



2의 보수

- 각 비트 반전
- 1을 더함

예시)

$$3 = 2^1 + 2^0$$

0	0	0	0	0	0	1	1
_		_	_	_			

각 비트 반전

Sign Bit

1	1	1	1	1	1	0	0	

1 더하기

1 1 1 1 1 0 1

각 비트 반전

0 0 0 0 0 1 0

1 더하기

0 0 0 0 0 1 1

-> 3 = $2^1 + 2^1$

8 bit max value = 127 $127 = 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0$

 $127 = 2^7 - 1$

0 1 1 1 1 1 1

16 bit max value = 32767

 $32767 = 2^{15} - 1$

0 1 1 1 1 1 1 1 1 1 1 1 1 1 1

32 bit max value = 2147483647

 $2147483647 = 2^{31} - 1$

0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

8 bit min value = -128

1	0	0	0	0	0	0	0

16 bit min value = -32768

		1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
--	--	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

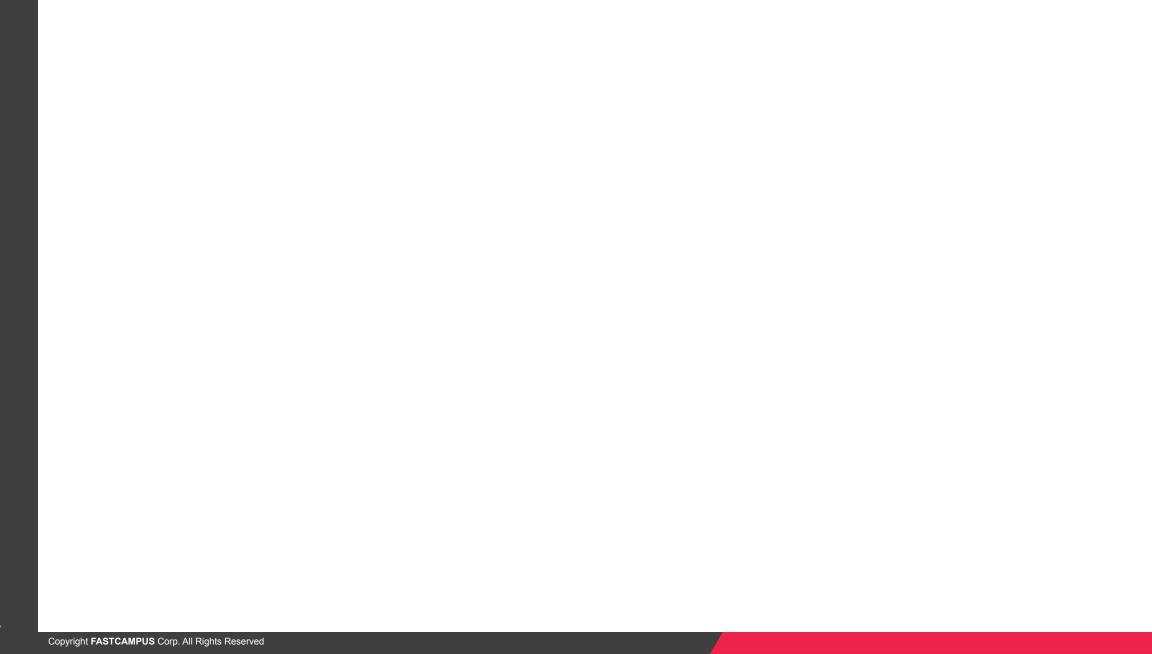
32 bit min value = -2147483648

1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

덧셈

$$-6 + (-7) = -13$$

-6(=	보수),	-7(5	선수),	-13(보수)		
1	1	1	1	1	0	1	0
1	1	1	1	1	0	0	1
1	1	1	1	0	0	1	1





덧셈(뺄셈)

$$6 - 7 = -1$$

$$6 + (-7) = -1$$

0	0	0	0	0	1	1	0
0	0	0	0	0	1	1	1
0	0	0	0	0	0	0	1

				*			
0	0	0	0	0	1	1	0
1	1	1	1	1	0	0	1
1	1	1	1	1	1	1	1

덧셈(뺄셈)

$$6 - 5 = 1$$

$$6 + (-5) = 1$$

0	0	0	0	0	1	1	0
0	0	0	0	0	1	0	1
0	0	0	0	0	0	0	1

0	0	0	0	0	1	1	0
1	1	1	1	1	0	1	1
0	0	0	0	0	0	0	1

Overflow

표현할 수 있는 최대값보다 더 커지는 상태



Overflow

Unsigned

1	1	1	1	1	1	1	1	255 + 1
0	0	0	0	0	0	0	0	= 0



Overflow

Signed

0	1	1	1	1	1	1	1	127 + 1
1	0	0	0	0	0	0	0	= -128

아래의 경우는 Overflow는 아님

1	1	1	1	1	1	1	1	-1 + 1
0	0	0	0	0	0	0	0	= 0

Underflow

표현할 수 있는 최저값보다 더 적어지는 상태



Underflow

Unsigned

0	0	0	0	0	0	0	0	0 + (-1)
1	1	1	1	1	1	1	1	= 255

Underflow

Signed

1	0	0	0	0	0	0	0	-128 + (-1)
0	1	1	1	1	1	1	1	= 127

