
데이터 처리

Variable naming rule

- 숫자는 변수이름 첫 자로 사용할 수 없다
- 변수이름은 영문, 숫자, _ 만 사용할 수 있다
- 변수이름은 영문 대소문자를 구별한다
- C++의 키워드(cin, cout 등)는 변수이름으로 사용할 수 없다
- __, _A 등과 같이 두 개의 밑줄, 밑줄 다음 대문자를 변수이름으로 사용하면 예상하지 못한 결과가 나올 수 있다
(__, _A 등은 header file에서 사용함)

※ 권장사항

- 여러 개의 단어를 결합한 변수이름은
단어 첫 글자를 대문자로 하거나, 각 단어를 _로 연결한다
- 변수의 용도를 알기 쉽게 하기 위해 **full name**을 사용한다

예) C++성적 : `cpp_sc`, `cpp_score`, `cppScore`

bit별 숫자표현

1bit

0 0

1 1

2bit

0 0 0

0 1 1

1 0 $2^1 + \cancel{2^0} = 2$

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

3bit

0 0 0

0 0 1

:

1 0 1

1 1 0

1 1 1

0

1

:

$2^2 + \cancel{2^1} + 2^0 = 5$

$2^2 + 2^1 + \cancel{2^0} = 6$

$2^2 + 2^1 + 2^0 = 7$

Variable의 종류

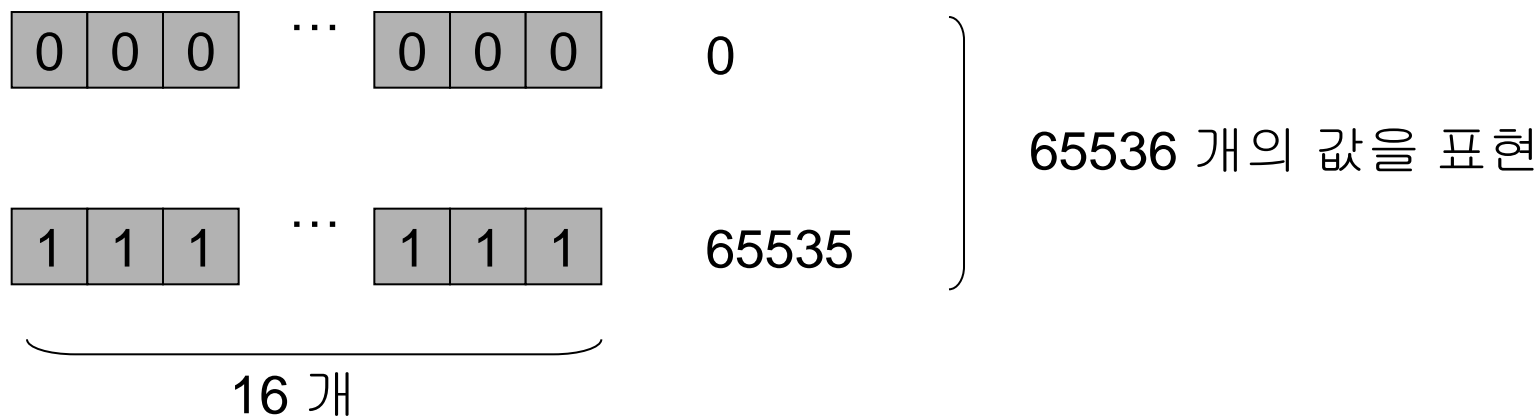


소수점형 { float - 32bit
double - 64bit
long double - 80 ~128bit

bool형 bool - 1bit

short

16bit메모리



$$2^{15} + 2^{14} + \dots + 2^1 + 2^0 = 65535$$

signed : -32768 ~ +32767

unsigned : 0~65535

```
//unsigned.cpp
#include <iostream>
using namespace std;

void main()
{
    short val1;
    unsigned short val2;
    val1 = 0;
    val2 = 0;

    val1 = val1 - 1;
    val2 = val2 - 1;
    cout << "signed short is " << val1 << endl;
    cout << "unsigned short is " << val2 << endl;
}
```



```
//unsigned.cpp
#include <iostream>
using namespace std;

void main()
{
    short val1 = 0; //선언과 동시에 초기화
    unsigned short val2 = 0; //선언과 동시에 초기화

    val1 = val1 - 1;
    val2 = val2 - 1;
    cout << "signed short is " << val1 << endl;
    cout << "unsigned short is " << val2 << endl;
}
```

char

8bit (0~255의 숫자저장이 가능)

```
char value = 'A';  
cout << value;
```

→ `char value = 'A';` //문자형 변수 `value`를 선언하고 `A`를 저장

→ 변수 `value`의 `memory`에는 `A`의 `ASCII code`에 해당하는 `65`가 저장됨

→ `cout << value;` // `65`는 `char`형으로 저장되었기 때문에
// `cout` 은 `65`를 `A`로 변환하여 출력

```
//char1.cpp
```

```
#include <iostream>
```

```
using namespace std;
```

```
void main()
```

```
{
```

```
    int val;
```

```
    val = 'A'; //+1
```

```
    cout << "val : " << val << endl;
```

```
}
```

```
//char2.cpp
```

```
#include <iostream>
```

```
using namespace std;
```

```
void main()
```

```
{
```

```
    char c_val;
```

```
    c_val = 65;
```

```
    cout << "c_val : " << c_val << endl;
```

```
}
```

```
//char3.cpp
#include <iostream>
using namespace std;

void main()
{
    int asc_dec;
    char asc_char;

    for(asc_dec=1; asc_dec < 128; asc_dec++)
    {
        asc_char = asc_dec;
        cout << "ASC DEC : " << asc_dec << " - ASC CHAR : " << asc_char << endl;
    }
}
```

bool

```
//bool.cpp
#include <iostream>
using namespace std;

void main()
{
    int iValue;
    bool bValue;

    iValue = true;
    bValue = 1; //0만 아니면 true
    //bValue = false;

    cout << "iValue : " << iValue << endl;
    cout << "bValue : " << bValue << endl;
}
```

제한자 (const)

초기화된 변수에 제한자를 붙이면 그 변수에는 다른 값을 대입할 수 없다

```
//const.cpp
#include <iostream>
using namespace std;

void main()
{
    const int seven = 7;
    int value;

    value = seven + 3;
    //seven = 9;
    cout << "Value is " << value << endl;
}
```