

variables and constants

# Hello World 분석하기

Hello World에 대해

- 1) 처음부터 Hello World 프로그램에 들어간 모든 소스코드를 이해할 수는 없음.
- 2) 반복적인 학습을 통해 모든 소스코드의 내용을 이해할 수 있도록 함.

```
#include <stdio.h>

int main(void) { printf("Hello World!");
    return 0;
}
```

# variables and constants

## 라이브러리 불러오기

- 1) C/C++에서는 `#include` 명령어를 이용해 다양한 라이브러리를 불러올 수 있음.
- 2) 집을 지으려면 망치와 같은 도구가 필요하듯이 `stdio.h`는 여러 기본적인 기능을 담고 있음.

```
#include <stdio.h>

int main(void)
{ printf("Hello World!");
  return 0;
}
```

# variables and constants

## 메인 함수

- 1) C/C++에서는 다양한 함수가 사용될 수 있으나 프로그램은 항상 메인(main) 함수로부터 시작됨.
- 2) 함수는 반환 값(Return Value)이 없을 수도 있으나 메인 함수에서는 항상 0을 반환하는 것이 일반적임.

```
#include <stdio.h>

int main(void) {
    printf("Hello World!");
    return 0;
}
```

# variables and constants

## 기본적인 출력 함수

- 1) C/C++에서 사용자에게 특정한 문자들을 출력하기 위해서 printf() 함수를 사용함.
- 2) printf()는 stdio.h 헤더 파일에 포함되어 있음.

```
#include <stdio.h>

int main(void) {
    printf("Hello World!");
    return 0;
}
```

# variables and constants

## 세미콜론

- 1) C언어에서는 하나의 명령어가 끝났음을 알리기 위해 세미콜론(;)을 붙임.

```
#include <stdio.h>

int main(void)
{
    printf("Hello World!");
    return 0;
}
```

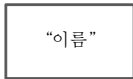
# variables and constants

## 변수와 상수의 개념

- 1) 변수(Variable)는 변할 수 있는 데이터임.
- 2) 상수(Constant)는 변하지 않는 데이터임.



이름표(Name Tag)



# variables and constants

## 변수의 선언

- 1) 변수를 선언할 때는 자료형과 변수명을 입력함. 원하는 경우 초기값을 적용할 수 있음.
- 2) 가장 많이 사용되는 변수는 정수형(Integer) 변수임.

```
int a;
```

```
int a = 7;
```



# variables and constants

## 기본 출력

- 1) `stdio.h` 헤더 파일에 선언된 `printf()`를 이용해서 기본적인 정수 데이터를 출력함.

```
#include <stdio.h>

int main(void) {
    int a = 7;
    printf("The number is %d.\n", a);
    return 0;
}
```

# variables and constants

## 변수의 초기화와 쓰레기 값

- 1) 초기화 되지 않은 변수는 쓰레기 값이 들어감.
- 2) Visual Studio는 기본적으로 초기화 되지 않은 변수를 감지하고 오류를 출력함.

```
#include <stdio.h>

int main(void) {
    int a;
    printf("The number is %d.\n", a);
    return 0;
}
```

# variables and constants

## 변수의 초기화와 쓰레기 값

- 1) 정적 변수로 선언된 것은 기본적으로 0으로 값이 초기화 됨.
- 2) 정적 변수가 아닌 수를 0으로 초기화 하려면 값을 일일이 넣어주어야 함.

```
#include <stdio.h>

int a;

int main(void) {
    printf("The number is %d.\n", a);
    return 0;
}
```

# variables and constants

## 기본적인 자료형

int

일반적인 정수형을 표현할 때 사용함.(억 단위)

long long

숫자가 긴 정수형을 표현할 때 사용함.

double

일반적인 실수형을 표현할 때 사용함.

string

문자열을 표현할 때 사용함.

bool

참/거짓을 표현할 때 사용함.

char

한 문자를 표현할 때 사용함.

# variables and constants

## 예약어와 식별자

- 1) 식별자(Identifier)란 변수나 함수 등의 고유한 이름을 지정할 때 사용함.
- 2) 이 때 C언어 문법으로 정해진 예약어는 식별자로 사용할 수 없음.

string

for

void

bool

if

while

char

return

double

# variables and constants

## 정수의 표현 방법

- 1) 컴퓨터가 정수(Integer)를 처리하는 방법을 이해함.
- 2) 컴퓨터는 내부적으로 2진수로 숫자를 표현함.
- 3) 9는 다음과 같이 나타냄.

0	0	0	0	1	0	0	1
부호	64	32	16	8	4	2	1

# variables and constants

## 정수의 표현 방법

- 1) -9는 어떻게 표현할까요?
- 2) 부호 절대값 방식은 다음과 같음.

1	0	0	0	1	0	0	1
부호	64	32	16	8	4	2	1

# variables and constants

## 정수의 표현 방법

- 1) 부호 절댓값 방식은 다양한 값의 연산을 수행하기 적합하지 않음.
- 2) 따라서 2의 보수를 사용함.
- 3) 2의 보수 = 1의 보수 + 1

1	1	1	1	0	1	1	1
부호	64	32	16	8	4	2	1



# variables and constants

## 정수의 표현 방법

- 1) 2의 보수를 이용하면 덧셈 연산만을 이용해서 양수와 음수를 연산할 수 있음.
- 2) 이 때 올림 수가 발생하면 무시함.

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

+

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

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

# variables and constants

## 실수의 표현 방법

- 1) 일반적으로 컴퓨터는 오차 없이 정확히 실수를 표현할 수 없음.
- 2) 따라서 넓은 범위의 실수를 표현하는 방식을 사용함.