

변수

#01. VSCode 설정

1) Extention 설치

이름	개발자	설명
Extension Pack for Java	Microsoft	자바 개발에 필요한 필수 익스텐션 모음
PlantUML	jebbs	클래스 다이어그램 생성 도구
PlantUML Previewer	Mebrahtom Guesh	PlantUML로 작성된 다이어그램 뷰어
PlantUML Syntax	Quido Hoekman	PlantUML 문법 강조 도구

2) 프로젝트 생성

1. **Ctrl+Shift+P**를 눌러 명령창을 연 후 **Java: Create Java Project**를 선택한다.
2. 다음 단계에서 **No build tools**를 선택한다.
3. 프로젝트(각 단위별 폴더)가 생성될 작업 디렉토리를 지정한다.
4. 프로젝트의 이름을 입력한다. (여기서는 **02_변수와_자료형**)

3) 소스코드 작성

모든 소스코드는 **src** 폴더 안에서 작성되어야 한다.

4) 코드 실행 및 디버깅

F5를 눌러서 코드를 실행한다.

BreakPoint가 설정되어 있는 경우 디버깅이 진행된다.

5) VSCode 디버깅 단축키

단축키	설명
F9	중단점 설정/해제
F5	디버깅 시작 디버깅 중인 경우 - (다른 중단점이 존재할 경우) 다음 중단점으로 이동 - (다른 중단점이 없을 경우) 나머지 코드 일괄 실행
F10	다음 줄 실행
F11	메서드 호출 구문에서 해당 메서드 안으로 들어가기

#02. 메모리

데이터를 저장할 수 있는 물리적 장치

1) 휘발성 메모리

전원을 OFF하면 저장되어 있는 데이터가 모두 삭제된다.

- ex) RAM 카드

2) 비휘발성 메모리

전원을 OFF 하더라도 데이터가 삭제되지 않는다.

- ex) 하드 디스크, USB

💡 사용자가 입력한 어떤 내용을 휘발성 메모리에 저장하면 변수, 비휘발성 메모리에 저장하면 파일이 된다.

#03. 컴퓨터의 데이터 단위

컴퓨터에 저장될 수 있는 데이터는 이진수 형태이므로 0 아니면 1로만 구성된다.

1) bit

이진수 값 하나를 의미.

- **1bit**는 두 개의 정보를 표현할 수 있다. (0 혹은 1)
- **2bit**는 네 개의 정보를 표현할 수 있다. (00, 01, 10, 11)
- **nbit**는 2의 n제곱 개의 정보를 표현할 수 있다.
 - 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 ...

💡 알파벳 'a'는 이진수로 표현했을 때 '01100001'가 된다.

2) 인코딩

- 글자를 부호화 하기 위한 규격.
 - ex) 모스부호

ANSI

- 이진수를 사용해서 알파벳을 부호화 하기 위한 미국 표준 규격
- 7bit로 알파벳 한 글자를 표현한다.
- 미국 전용.

확장 ANSI

- ANSI에서 다른 언어를 표현하기 위한 1bit를 추가하여 8bit로 한 글자를 표현하는 방식
- 영어 + 1개국어 표현이 가능하기 때문에 언어별로 종류가 나누어진다.
 - ex) euc-kr, euc-jp, euc-ch ...
- 확장 ANSI가 국제표준으로 정해진 후 컴퓨터에서 데이터의 크기의 초소 단위를 8bit로 정하고 이를 1byte라고 부르기 시작함.
- 영어는 1byte로 한 글자를 표현 가능함.
- 한국어와 같은 조합형 글자는 두 개 이상의 byte가 모여야 한 글자를 구성할 수 있다.

UTF-8

- 확장 ANSI가 영어+1개국어 밖에 표현하는 한계를 개선하여 언어 종류에 상관 없이 3byte로 한 글자를 표현하는 방식.
- 한번에 모든 언어를 표현할 수 있다.
- 현재 사실상 산업 표준으로 사용되고 있다.

3) 컴퓨터의 용량 단위

- byte : 최소단위 (이진수 8개)
- kbyte : 1kb = 1024b
- mbyte : 1mb = 1024kb
- gbyte : 1gb = 1024mb
- tbyte : 1tb = 1024gb

💡 16gb의 RAM을 장착한 경우 $16 * 1024 * 1024 * 1024 * 8$ 개의 이진수를 저장할 수 있는 공간이 존재한다는 의미

#04. 변수

메모리에 저장된 데이터.

쉽게 생각하면 값을 대입하기 위한 메모리 영역이라고 할 수 있다.

1) 변수 사용하기

변수의 선언

변수에 저장할 수 있는 값의 종류를 명시하고 그 옆에 변수의 이름을 정의한 다음 세미콜론으로 명령을 종료한다.

int는 정수를 의미하고 4byte의 메모리 공간을 차지한다.

```
int num;
```

값의 할당

대입 연산자 =을 사용해서 선언한 변수에 데이터를 저장하는 것.

항상 오른쪽에서 왼쪽으로 대입된다.

```
num = 100;
```

💡 `num`이라고 이름 지어진 4byte의 메모리 공간에 10진수 100을 이진수로 변환하여 기록한다는 의미.

선언과 할당의 일괄 처리

```
int num = 100;
```

2) 데이터 타입 (Data Type)

메모리에 저장되어 있는 이진수들을 해석하기 위한 규칙들.

메모리에 저장할 수 있는 값의 종류를 의미한다.

종류	크기	설명	값 형태
byte	1byte	-128~127 까지의 범위를 갖는 정수	ex) 1
short	2byte	byte 보다 큰 범위를 저장할 수 있는 정수	ex) 200
int	4byte	short 보다 큰 범위를 저장할 수 있는 정수	ex) 1234567
long	8byte	int 보다 큰 범위를 저장할 수 있는 정수	생략
float	4byte	실수	ex) 3.14
double	8byte	float 보다 더 정밀하게 소수점 자리를 표현할 수 있는 실수	ex) 3.141592...
boolean	2byte	참/거짓을 표현하는 논리값	true / false
char	1byte	한 글자. 홀따옴표로 감싼다.	'a'

💡 예를 들어 int형 변수를 만든다는 것은 메모리의 4byte 공간(32칸)을 사용한다는 의미입니다.

3) 변수 값 활용하기

같은 데이터 타입을 갖는 변수 여러 개를 한 번에 선언하기

```
int num1, num2;
num1 = 100;
num2 = 200;
```

이미 값이 할당된 변수는 다른 값으로 변경 가능

```
int pay = 100;
pay = 200;
```

변수 값을 다른 변수에 복사 가능

```
int x, y;
x = 50;
y = x;

int z = y;
```

변수의 이름 규칙

1. 언더바(_), 숫자, 영어, \$기호만 사용 가능하다.
 - 관례상 \$는 잘 사용하지 않는다.
2. 첫 글자는 숫자가 될 수 없다.
3. 자바에서 사용하는 예약어를 사용할 수 없다.

4. 첫 글자는 대부분 소문자로 시작한다.

💡 소스코드가 UTF-8인 경우(서브라임을 사용해서 작성하는 경우) 한글도 변수 이름에 사용할 수 있습니다.

두 개 이상의 단어를 조합하여 이름을 정의하는 경우

1. 스네이크 표기법 : 띄어 쓰기를 언더바(_)로 표현

```
int my_age;
```

1. 카멜 표기법 : 띄어쓰기가 시작되는 단어의 첫 글자를 대문자로 변경

```
int myAge;
```

💡 자바의 경우 대부분 카멜표기법을 따릅니다.

문자열

쌍따옴표로 감싸진 문장을 저장할 수 있는 데이터 타입

```
String welcomeMessage = "Hello World";
```

💡 데이터 타입 첫 글자가 대문자로 시작하는 것에 주의!!!

4) 변수의 제약

선언되지 않은 변수는 사용할 수 없다.

```
int x;
x = 100;    // 선언된 변수를 사용하므로 ok
y = 200;    // 선언되지 않은 변수에 값을 대입하므로 에러
```

이미 선언된 변수를 중복 선언할 수 없다.

```
int z;
z = 200;

int z = 300;    // 중복선언 에러
```

float 타입 사용시 주의사항

double 타입과의 구분을 위해 대입시 F를 적용해야 한다. (필수)

```
float k = 12.345F;
```

💡 Java는 변수에 대입되는 모든 실수형 값을 double로 기본 인식합니다.

long 타입 사용시 주의사항

int 타입과의 구분을 위해 대입시 L을 적용해야 한다. (권장)

```
long k = 12345678L
```

💡 Java는 변수에 대입되는 모든 정수형 값을 int로 기본 인식합니다.

변수 값 출력하기

```
int myAge = 100;

System.out.println(myAge);    // 100이 출력된다.
System.out.println("myAge");  // myAge라는 문자열이 출력된다.
```