



Seoul
Software
ACademy

웹 개발자 부트캠프 과정

SeSAC x CODINGOn

With. 팀 리처드

식별자 원칙

식별자란?

- 클래스, 변수, 상수, 메소드 등에 붙이는 이름

식별자의 원칙

- @, #, ! 와 같은 특수문자 사용 불가
 - 예외로 ‘_(언더바)’, ‘\$(달러)’는 사용 가능
- 공백 불가
- 한글 사용 가능하나... 다들 준개발자시니 당근 영어 사용하실거죠?
- 미리 정의된 키워드(예약어)는 식별자로 사용 불가
- 숫자 사용 가능
 - 단, 첫 글자로는 사용 불가

식별자 원칙

- 사용 가능한 예

```
char name;
int student_ID; // '_' 사용 가능
void $func() { } // '$' 사용 가능
class Hello2 { } // 숫자 사용 가능
int 가격; // 한글 사용 가능
```

- 잘못된 예

```
int if; // 예약어 사용 불가
class 2Hello { } // 숫자 첫 글자 사용 불가
char name^^; // 특수문자 사용 불가
```

자바의 예약어

abstract	continue	for	new	switch
assert	default	if	package	synchronized
boolean	do	goto	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const	float	native	super	while

데이터 타입

데이터 타입

Java는 강한 타입 언어로,
Javascript에서와 달리 Java에서는 데이터 타입이 중요 함!

자바의 데이터 타입은 크게 아래 두 가지로 나뉨

1. 기본 타입 (Primitive)
2. 레퍼런스 타입 (참조형, Reference)

기본형 데이터 타입

- boolean --- 논리 타입
 - char --- 문자 타입
 - byte
 - short
 - **int**
 - long
 - float
 - **double**
- } 정수 타입
- } 실수 타입

타입	크기
byte	$-2^7 \sim 2^7 - 1$
short	$-2^{15} \sim 2^{15} - 1$
int	$-2^{31} \sim 2^{31} - 1$
long	$-2^{63} \sim 2^{63} - 1$
float	$-3.4 \times 10^{38} \sim 3.4 \times 10^{38}$
double	$-1.7 \times 10^{308} \sim 1.7 \times 10^{308}$

주의) 실수형 타입

- float 또는 double과 같은 기본 부동소수점 자료형은 근사값을 사용하므로 금융 계산과 같은 정밀도가 중요한 연산에는 적합하지 않음
- BigDecimal 클래스는 이러한 문제를 해결하기 위해 정확한 소수점 연산을 제공

- BigDecimal은 불변(immutable) 객체. 즉, 한 번 생성되면 그 값을 변경할 수 없음
- 문자열 생성자를 사용하는 것이 부동소수점 오류를 피하는 가장 안전한 방법
- 기본 산술 연산자(+, -, *, /)를 직접 사용할 수 없고 대신 메서드를 사용

```
import java.math.BigDecimal;

public class HelloWorld {

    public static void main(String[] args) {
        BigDecimal a = new BigDecimal("0.1");
        BigDecimal b = new BigDecimal("0.2");

        BigDecimal sum = a.add(b);
        BigDecimal difference = a.subtract(b);
        BigDecimal product = a.multiply(b);
        BigDecimal quotient = a.divide(b);

        System.out.println(sum);
        System.out.println(difference);
        System.out.println(product);
        System.out.println(quotient);
    }
}
```

참조형 데이터 타입

기본형 타입을 뺀 모든 형태를 참조형 타입으로 봄

크게 아래 세 가지로 구분 할 수 있음.

- 배열에 대한 레퍼런스
- 클래스에 대한 레퍼런스 (ex. `String`)
- 인터페이스에 대한 레퍼런스

입력과 출력

출력

```
System.out.print();
```

```
System.out.println();
```

 : 마지막에 Enter(\n)를 자동으로 넣어 줌.

예시

```
System.out.print("Hello 1");
System.out.println("Hello 2");
System.out.print("Hello 3");
```

입력

Scanner 클래스를 이용함.

```
import java.util.Scanner; // import 문 필요
...

Scanner a = new Scanner(System.in); // Scanner 객체 생성
```

- System.in에게 키를 읽게 하고, 읽은 값을 String, int, double, boolean, char 등 다양한 타입으로 변환하여 리턴
- 입력되는 키 값을 공백(`\n`, `\t`,)으로 구분되는 아이템 단위로 읽음.

Scanner 클래스 주요 메소드

- next() : 공백 이전까지의 문자열을 읽음.
- nextLine() : 엔터(\n) 이전까지의 문자열을 읽음.
- next자료형()
 - nextInt() : 공백 이전까지의 정수를 읽음.
 - nextDouble() : 공백 이전까지의 실수를 읽음.
 - nextBoolean(), nextFloat(), ..etc
- close() : scanner 닫음.

Scanner 클래스 주요 메소드

```
System.out.println("이름 나이, 키, 결혼 여부를 빈칸으로 분리하여 입력하세요");

Scanner scanner = new Scanner(System.in);

String name = scanner.next(); // 문자열 읽기
int age = scanner.nextInt(); // 정수 읽기
double height = scanner.nextDouble(); // 실수 읽기
boolean single = scanner.nextBoolean(); // 논리값 읽기

System.out.print("이름은 " + name + ", ");
System.out.print("나이는 " + age + "살, ");
System.out.print("키는 " + height + "cm, ");
System.out.println("결혼 여부는 " + single + "입니다.");

scanner.close(); // scanner 닫기
```


연산자

연산자 – 정수형

연산자	예시	결과
+	$3 + 2$	5
-	$3 - 2$	1
*	$3 * 2$	6
/	$5 / 2$	2
%	$5 \% 2$	1

나눗셈 결과의 몫

나눗셈 결과의 나머지

연산자 – 실수형

연산자	예시	결과
+	$3.0 + 2.0$	5.0
-	$3.0 - 2.0$	1.0
*	$3.0 * 2.0$	6.0
/	$5.0 / 2.0$	2.5
%	$5.0 \% 2.0$	1.0

실습. 기본 실습

- “이름을 입력하세요.” 라는 문구를 출력하고 사용자로부터 이름을 입력 받기
- “나이를 입력하세요.” 라는 문구를 출력하고 사용자로부터 나이를 입력 받기
- 이름과 나이의 입력이 끝나면 "안녕하세요! OOO님(OO세)" 라는 문구 출력

```
이름을  입력하세요
```

```
코딩온
```

```
나이를  입력하세요
```

```
99
```

```
안녕하세요!  코딩온님 (99세)
```