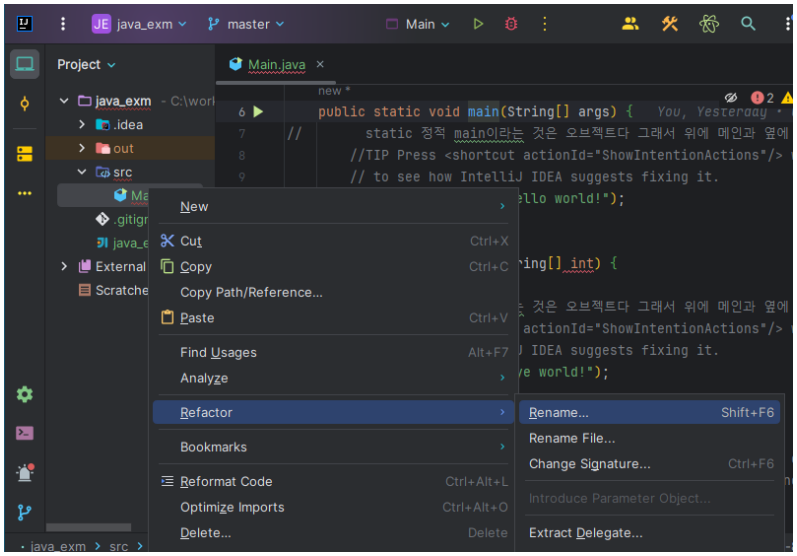
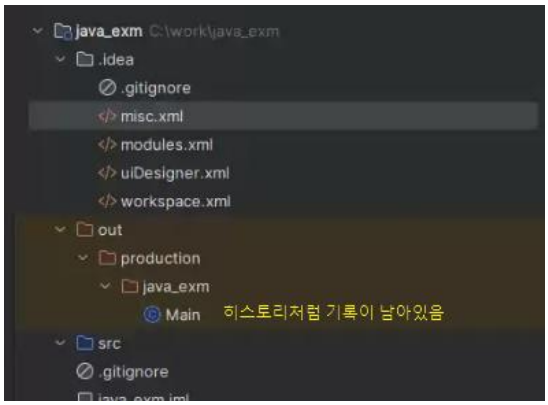
23-12-14

- JAVA 문장구조

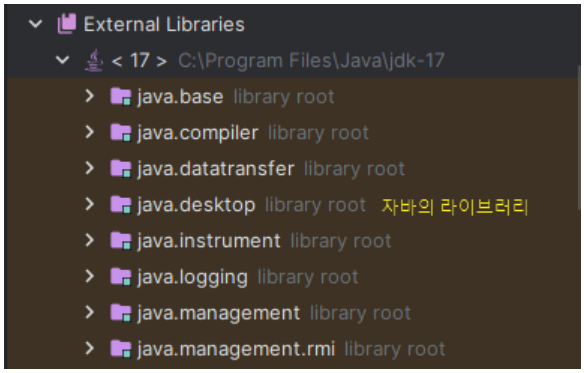
자바 파일명과 class 명은 동일해야 한다.



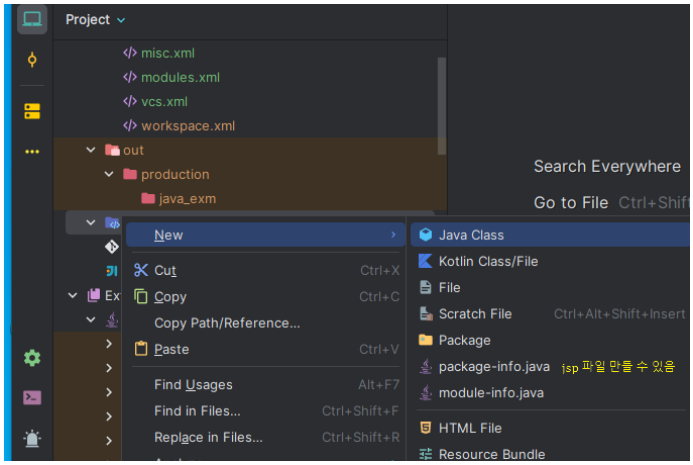
- Refactor > Rename (shift+F6) : class 파일 이름변경



파일을 삭제해도 out 디렉토리에 들어가면 Main이라는 것을 확인할 수 있다(기록이 남는다)

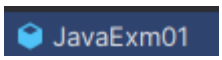


자바의 라이브러리

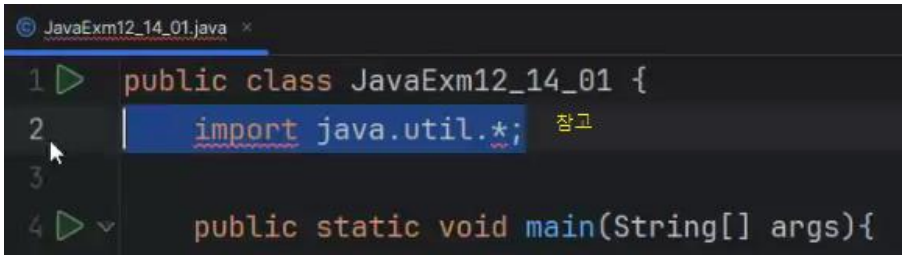


- src > new > java class : 새로운 파일 만들기

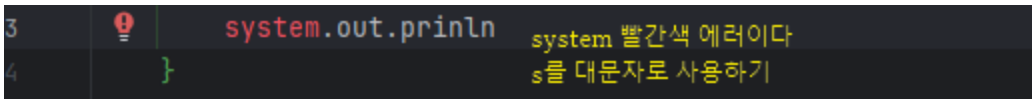
(아래 package-info.java는 jsp파일 만들 수 있는 파일이다)



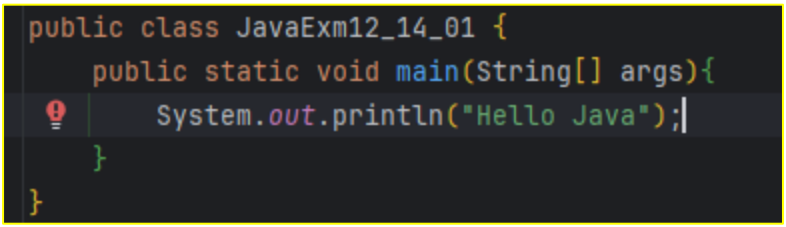
- 파일명 : 숫자로 시작할 수 없고, - 사용불가



참고만 하기



system부분에 빨간색은 에러인 것. -> s를 대문자 S로 바꿔주기



main은 오브젝트 -> 그것을 string으로 설정해서 배열(문자열)로 받겠다

string[ ] args는 첫 스레드인 main에게 데이터를 넘겨주고 싶을 때를 위한 파라미터이다.

string[ ] 문자를 배열로 받겠다 args는 main에 대한 변수명(arguments)

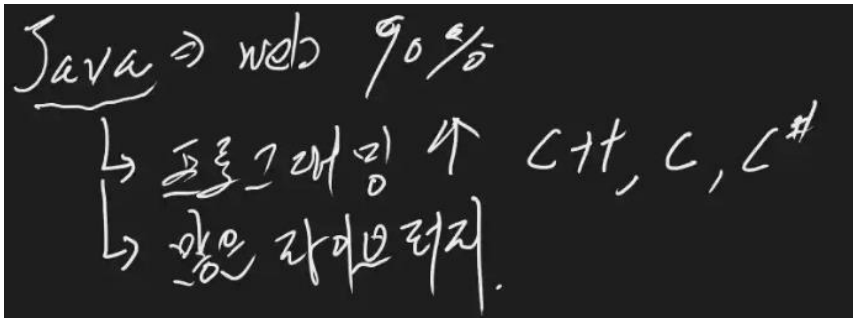
public 내가 만든 클래스 이외에 다른 클래스에서도 사용하려고 붙여둔 것

static 정적메소드, 객체 생성없이 메소드를 사용하려면 static이 붙어 있어야함

System.out.println은 출력해라 ("hellow Java") hello java라고!

void : return 값이 없다

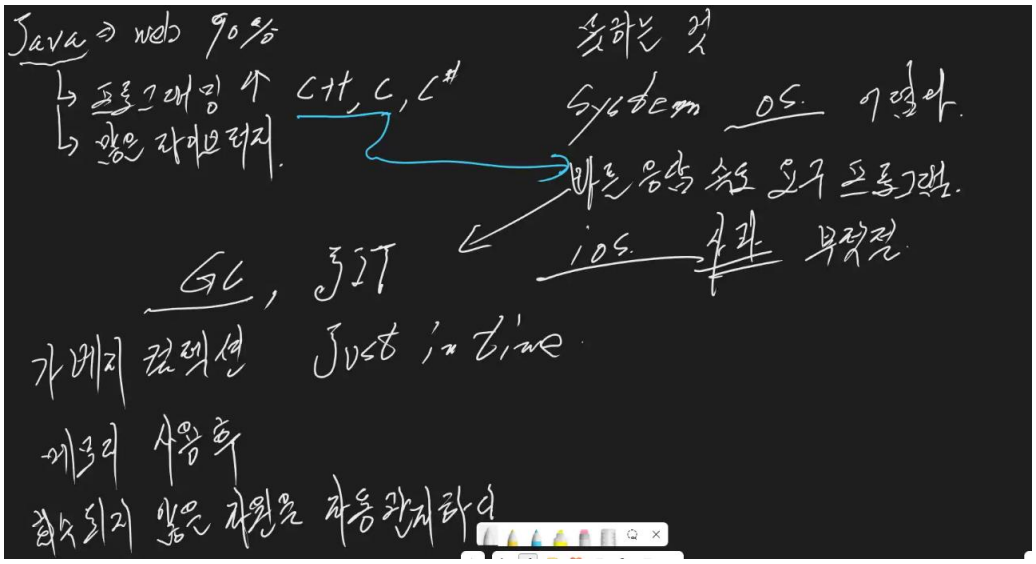
https://lordofkangs.tistory.com/12



Java가 web에서 90% 사용한다 -> 프로그래밍↑, 많은 라이브러리가 있다 (뿌리는 자바다)

웹 개발 언어

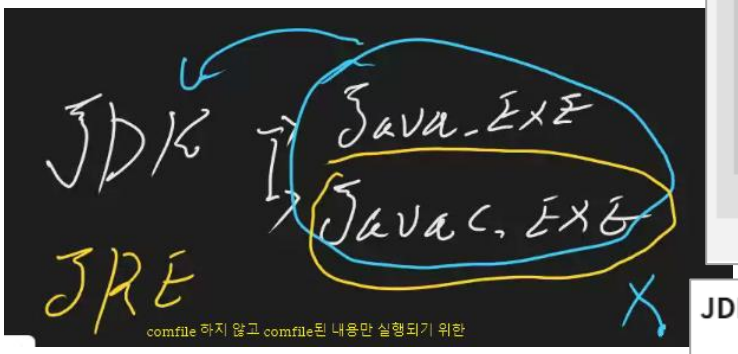
1. JAVA
2. C++
3. C(보안사업), C#

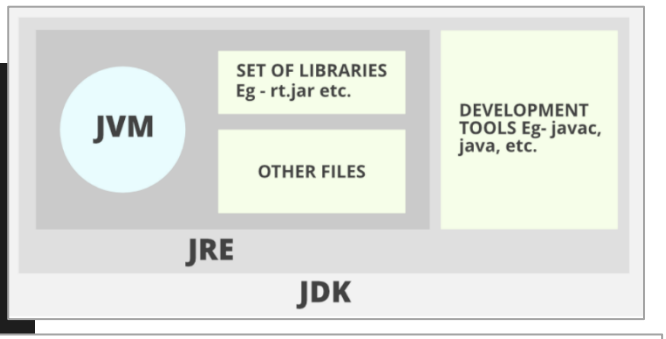


java로 못하는 것

1. system 프로그램 (os 운영체제 만드는 것 -> 할 수는 있는데 어렵다)
2. 빠른 응답 속도 요구 프로그램(빠른 응답 요구하는 건 c나 c++로 만든다
   (ex_원자력제한시스템,화력발전소터빙관리,재난방지문자))
   자바로 만들 수 있지만 c와 c++보다 느리다
   -> GC(가베지 컬렉션_메모리 사용후 회수되지 않은 자원을 자동 관리하다), JIT(just in time) 때문이다.
3. ios 애플 프로그램 부적절(자바는 android 에 최적화)

~~(자바는 메모리 규격하지만~~
~~파일이 메모리 부분 꽂히고 사용하는~~
~~쓰레기 처리 부담을 빼버린 것이 자바)~~
~~메모리 부분을 날려버린 것이 자바이다 -> 회수하는 기간이 생기기 때문에 반응 속도가 늦다~~
~~(c++이런것은 c로 바로 수정하고 변경 가능하다?)~~



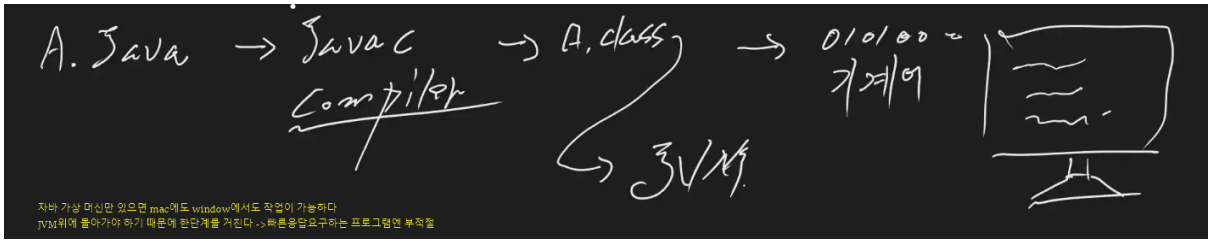


JDK 에서 개발도구인 java와 javac 란?

javac는 *.java 파일을 *.class 파일로 컴파일 해주는 **컴파일러** 이다.
java는 javac로 컴파일 된 class 파일을 실행하는 **프로그램**이다.

JDK : (java_EXE와 javac_EXE), JRE

- java_EXE
- javac_EXE : c 는 comfile(번역기)할 수 있다.



자바 가상머신만 있으면 mac에도 window에서도 작업이 가능하다
JVM위에 돌아가야 하기 때문에 한단계를 거친다. -> 빠른 응답 요구하는 프로그램엔 부적절

접근제한자	기능
public	모든 곳에서 접근 가능
private	자기 자신 클래스에서만 접근 가능
protected	자기 자신 클래스와 상속된 클래스에서 접근 가능

method(메소드) 접근자 -> public, private, protected

스레드(thread)는 어떠한 프로그램 내에서, 특히 프로세스 내에서 실행되는 흐름의 단위를 말한다. 일반적으로 한 프로그램은 하나

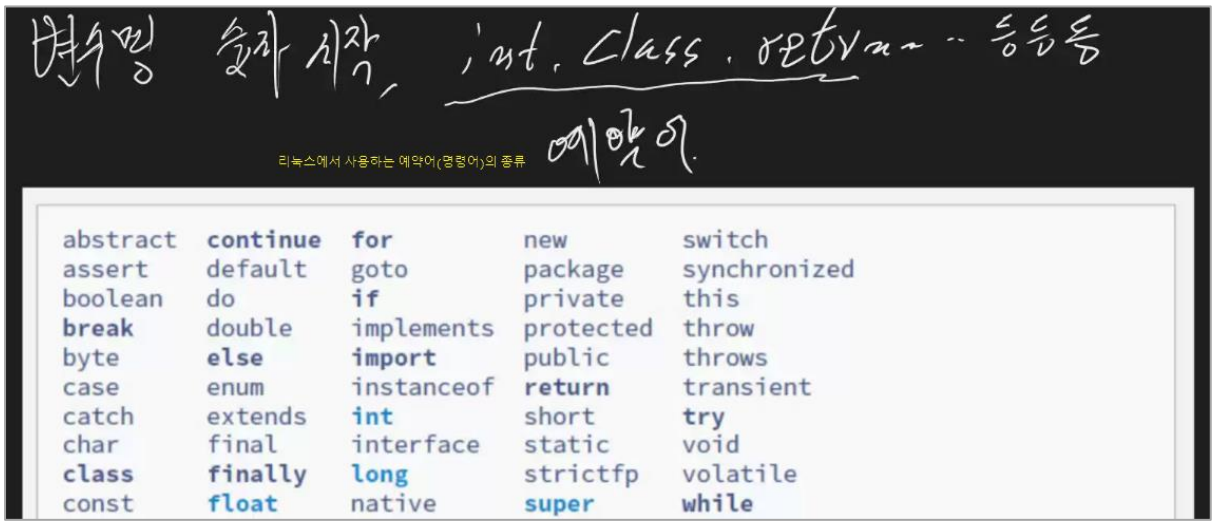
method. 일정한 계획에 따른 방법, 격식과 정연에 따른 순서를 의미한다.

*** class 구조**

class가 하나의 파일이다 (자바파일)
그 파일 안에 메소드가 있다.
메소드 안에는 명령문을 포함하고 있다
ex) 자료처리형 메소드를 만들어 놓으면 그 안에서 구구단을 실행할 수 있는 것

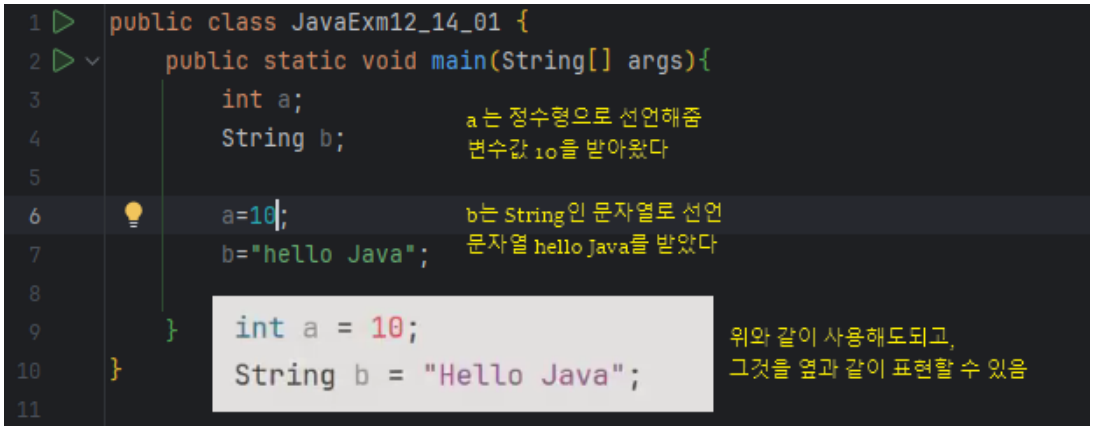
함수(function)와 메소드 (method)의 차이점

함수는 존재하는 명령문의 집합이고 로직 처리 이후 사용자가 원하는 결과를 반환할 수 있다.
클래스, 객체와 연관돼있지 않고 독립적으로 사용할 수 있고 메소드는 클래스에 속해 있는 함수를 말한다.
자바는 클래스를 벗어나 함수를 생성할 수 없는 언어로 자바에서의 모든 함수는 곧 메소드인것이다.
메소드는 함수에 포함된 개념이라 볼 수 있다.



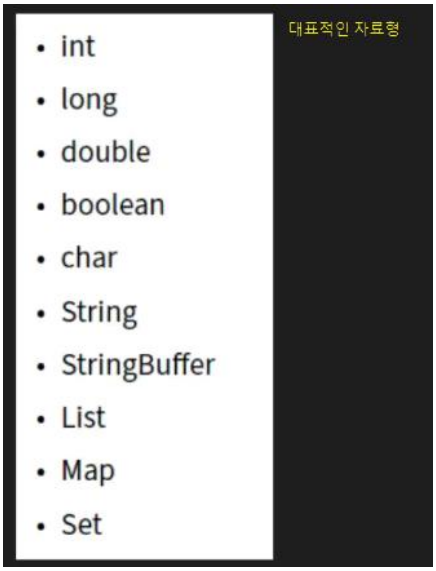
리눅스에서 사용하는 대표적인 명령어 종류

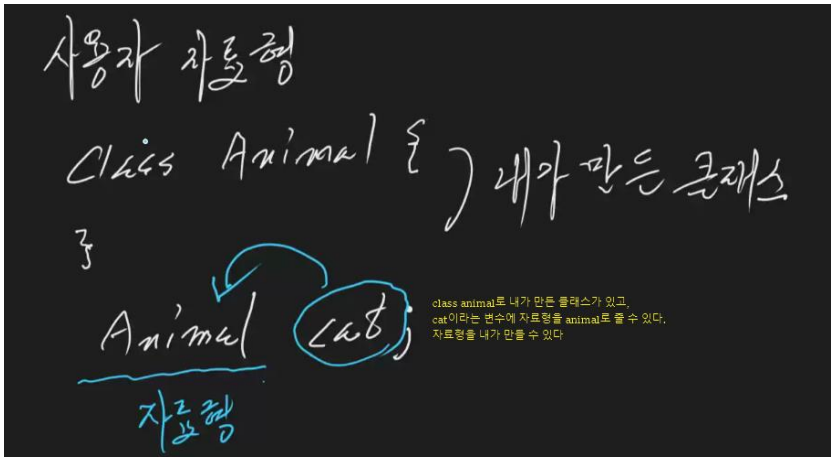
파란색 : 자료형, 남색 : 조건문

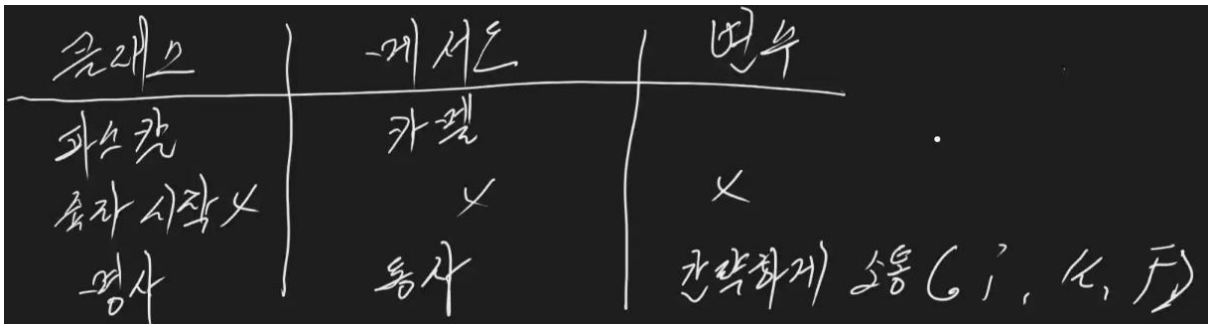


대표적인 자료형

- int : 정수처리에 사용
- long : 정수처리에 사용
- boolean true false
- char 문자(단어)
- string 문자열
- List : 배열
- Map : 배열







클래스 명은 파스칼 명사

메소드는 카멜로 만들고 동사

변수 명은 어떠한 것을 사용해도 상관없다,간략하게 소통 가능하게 만들면됨(i,k,f이렇게 변수명 만들지 말기)

- **정수**

2 ^ 31 =
2,147,483,648

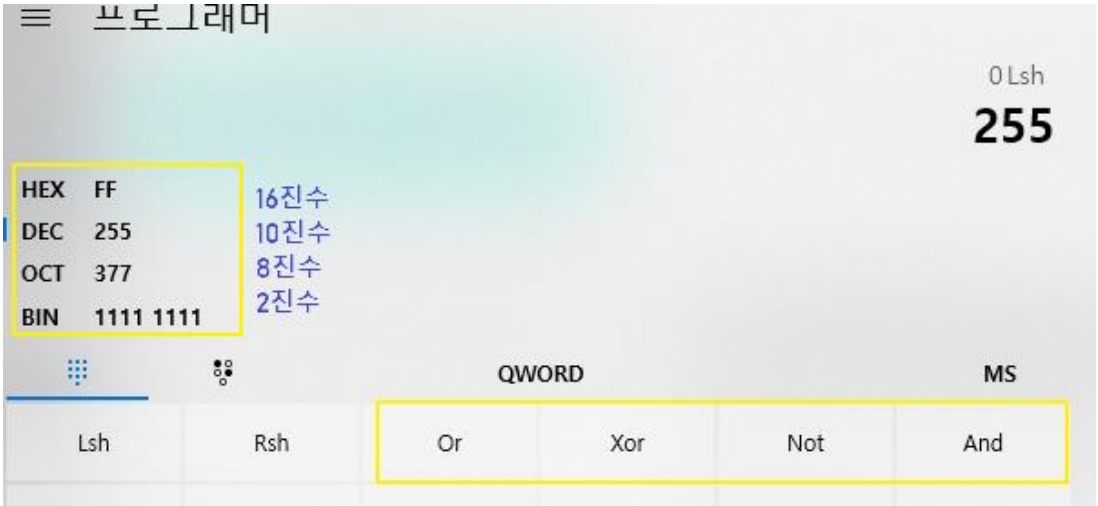
int로 쓸 수 있는 최대 숫자 : 2,147,483,694 (2^31)

long : int의 두배 : Byte , 우리가 사용하는 경우에는 int로 가능하여 long 사용할 일이 없다
int의 값 합이 되었을 때 long사용할 수도 있다
(최종 합산 값은 long으로 놓는다 int의 최대값 넘어가면 에러가 날 수도 있기 때문에)

- **실수**

float 표현하는 범위 3.4*10^38

double 은 float의 2배



- 인텔리제이로 정수 계산 출력 해보기

```
int a =10;
int b = 5;
System.out.println(a+b);
System.out.println(a-b);
System.out.println(a*b);
System.out.println(a/b);
```

```
15
5
50
2
```

System.out.println(7/3); //값2나옴

System.out.println(3/7); //값0나옴

int로 정의하지 않아도 기본적으로 int로 계산되어 출력된다

System.out.println(7%3); //값1나옴

System.out.println(3%7); //값3나옴

%를 사용하면 몫이 아니라 **나머지 값**이 출력됨

a++ 후위법 : 출력하고 더할건지

++a 전위법 : 더하고 출력할건지 차이

```
int a =0;
int b=10;
System.out.println(a++);
System.out.println(b++);
```

```
0
10
```

값을 출력하고 +1을 한다.

```
int a =0;
int b=10;
System.out.println(++a);
System.out.println(++b);
```

```
1
11
```

값에 +1을 하고 출력한다

```
int i = 0;
System.out.println(i++);
System.out.println(i);
```

```
0
1
```

후위법은 값이 다르게 나옴

```
int i = 0;
System.out.println(++i);
System.out.println(i);
```

```
1
1
```

전위법은 값이 같게 나옴

```java
boolean isSuccess = true;
boolean isTest = false;

int a = 5;
int b = 7;
String c = "";

c = a > b;
a == b;
```

```java
System.out.println(3<1);
```

```
false
```

불대수 값을 설정하지 않아도 이렇게 입력하면 값은 false라고 boolean형태의 값이 나온다.

```java
int base = 180;
int high = 185;
boolean isTall = high  >  base;

if(isTall){
    System.out.println("평균보 다 크네");
}
```

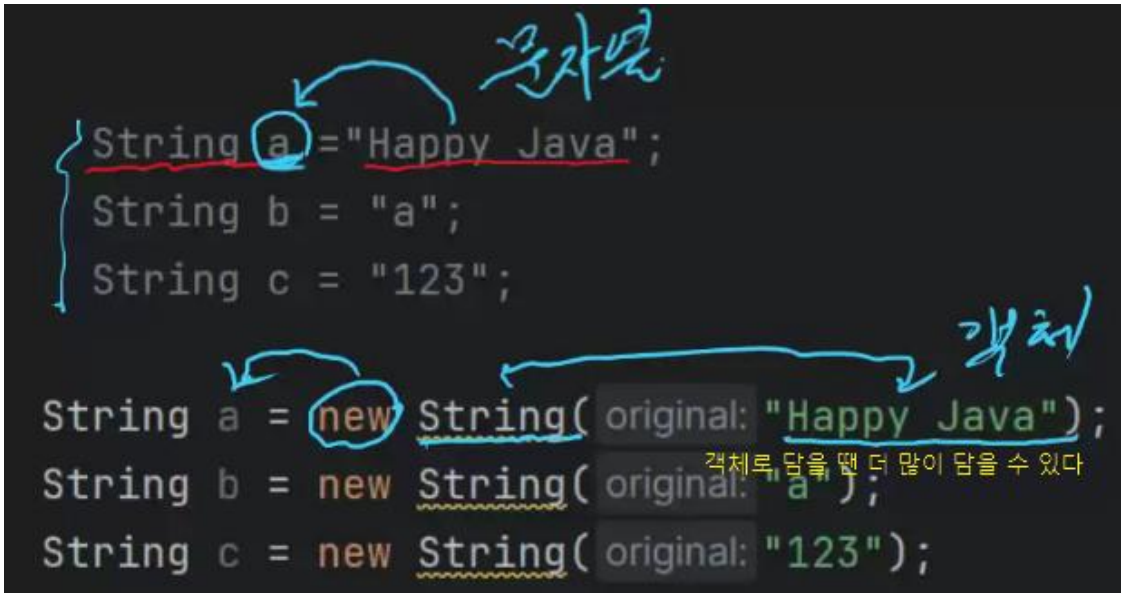
boolean 불대수 -> 조건문을 사용한다

```java
char a1 = 'a';
char a2 = 97;
char a3='\u0061';
System.out.println(a1);
System.out.println(a2);
System.out.println(a3);
```

```
a
a
a
```

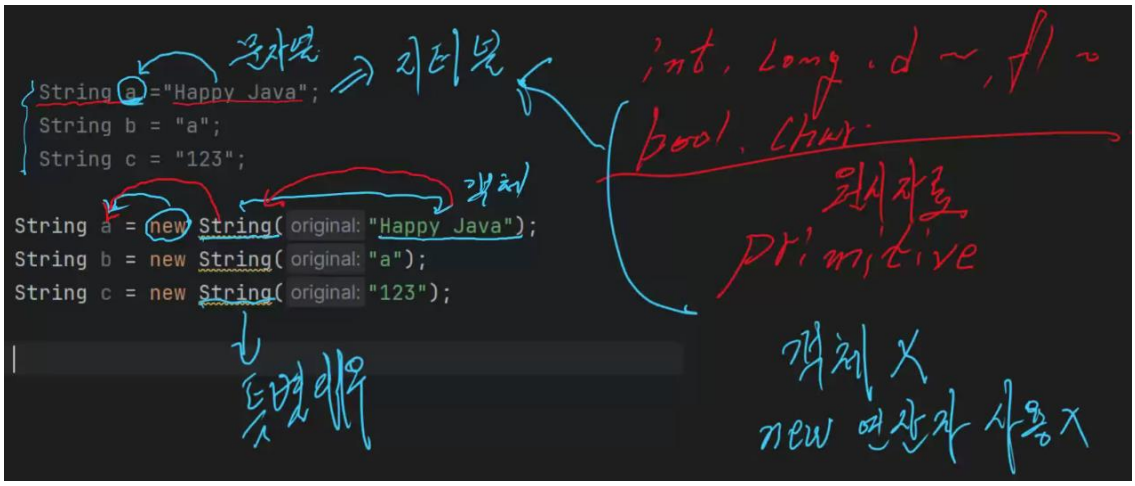
제어 문자			공백 문자			구두점			숫자			알파벳			
10진	16진	문자	10진	16진	문자	10진	16진	문자	10진	16진	문자	10진	16진	문자
0	0x00	NUL	32	0x20	SP	64	0x40	@	96	0x60	`
1	0x01	SOH	33	0x21	!	65	0x41	A	97	0x61	a
2	0x02	STX	34	0x22	"	66	0x42	B	98	0x62	b
3	0x03	ETX	35	0x23	#	67	0x43	C	99	0x63	c
4	0x04	EOT	36	0x24	$	68	0x44	D	100	0x64	d
5	0x05	ENQ	37	0x25	%	69	0x45	E	101	0x65	e
6	0x06	ACK	38	0x26	&	70	0x46	F	102	0x66	f
7	0x07	BEL	39	0x27	'	71	0x47	G	103	0x67	g

char a2 = 97; 아스키 코드 값으로 97은 a이다.



1. 리터널 방식 : 해피자바를 문자열로 a에 집어넣겠다 -> 객체가 아니라 문자열로 들어감
2. new : 해피자바라는 문자열 자체를 문자열, 하나의 덩어리 => 객체가 된다. (new를 써줌으로써)
   그리고 그것을 다시 a 객체로 담는다

   => a는 해피자바 문자열이 들어가는 건 맞긴한데 객체를 저장하는 것이다. 객체에서 해피자바를 끌어오는 구조이다.



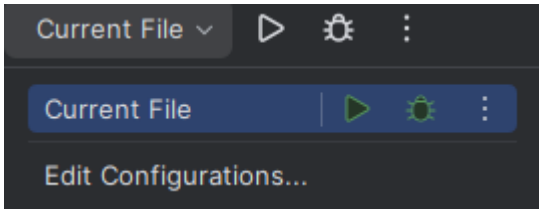
new가 객체형태로 집어넣겠다는 뜻이다
string에 집어넣을 건데 happy java를 스트링(문자열)로 받는 것.

해피자바를 문자열로 a에 집어넣겠다 -> 객체가 아니라 문자열로 들어감
-> int ~ 원시자료이다 primitive -> 객체가 아님 , new 연산자 사용 못함

해피자바라는 문자열 자체를 문자열, 하나의 덩어리 => 객체가 된다.
(new를 써줌으로써) 그리고 그것을 다시 a 객체로 담는다
-> string은 특별대우,문자열도 될 수 있고 객체도 될 수 있는 유일한아이

string은 원시자료이기도 객체이기도 하다

원시 자료형	Wrapper 클래스
int	Integer
long	Long
double	Double
float	Float
boolean	Boolean
char	Character



current File이 내 파일

```
String a = "hello";
String b = "java";
String c = "hello";
String d = "hello java";
System.out.println(a.equals(b));
System.out.println(a.equals(c));
System.out.println(d.indexOf("java"));
"java"라는 글자가 몇번째에 위치하는지 알려주는 것 => 6
```

hello Java
0 1 2 3 4 5 6 7 8 9

```
false
true
6
```

- **equals** -> a와 b와 비교해서 같으면 true, 같지 않으면 false 출력

password 같은지 확인할 때 사용할 수 있음

문자열이 같고 그 문자가 같으면 true, 틀리면 false 출력되도록 하는것

- **indexOf** -> 뒤에 입력한 "java"라는 글자가 몇 번째 위치하는지 찾아서 출력

```
System.out.println(d.contains("ja"));
System.out.println(String.format("나는 사과  %d 를 가지고 있지롱",3));
```

```
true
나는 사과  3 를 가지고 있지롱
```

- **contains** -> 뒤에 입력한 'ja"라는 글자가 포함되어 있는지 불대수로 출력(true, false)

@ 이메일에 포함되어 있는지 확인할 때 활용할 수 있음

아이디 사용불가능 가능 활용할 때 ~~포함하면 안될때~~

- String.format -> 컴퓨터에서 포맷은 형식지정할 때 사용한다, 출력된 값의 형식을 변경하는

```
System.out.println(String.format("나는 사과  %d%s 가지고 있지롱",3,"개를"));
```

%d 는 정수 값을 받는것

%s 는 문자열을 받는 것

그 값과 일치하는 값이 아니면 출력되지 않는다.

ex)%s, 3 이렇게 연결하여 입력하면 출력 X

%f	실수(float)
%d	정수(integer)
%s	문자열(string)

코드	설명
%s	문자열 (String)
%c	문자 1 개 (character)
%d	정수 (Integer)
%f	부동소수 (floating-point)
%o	8 진수
%x	16 진수
%%	Literal % (문자 % 자체)

```
String data = "papa";
System.out.println(data);
System.out.println(String.format("%10s",data));
```

10개 공간 안에서 데이터 입력해라
=> 6칸 띄고 4칸에 papa를 넣은 것

```
papa
      papa
```

%10s 라고 하는 것은 10개의 공간안에서 데이터 입력해라 -> papa라는 데이터 포함하여 10공간

```
System.out.println(String.format("%-10s%s",data,"jin")); //papa포함해서 10자리 + jin(3자리) => 총13자리
System.out.println(String.format("%-10s",data)+"jin");
System.out.println(String.format("%.4f",3.141592));
```

```
papa      jin
papa      jin
3.1416
```

%-10s하면 왼쪽 정렬

jin을 포함하면 3자리가 추가된 것이니까 총 13자리이다.

원주율 구하는 것으로 %.4f는 실수형이고, .뒤에 4자리까지 출력한다는 것

```
String result =sb.toString();
System.out.println(result);
System.out.println(sb.substring(0,4));
```

시작위치와 끝나는 위치 설정해서 출력

```
hello go to sky
hell
```

StringBuffer 문자열 추가, 변경시 사용

append 추가

StringBuffer(); 빈배열임

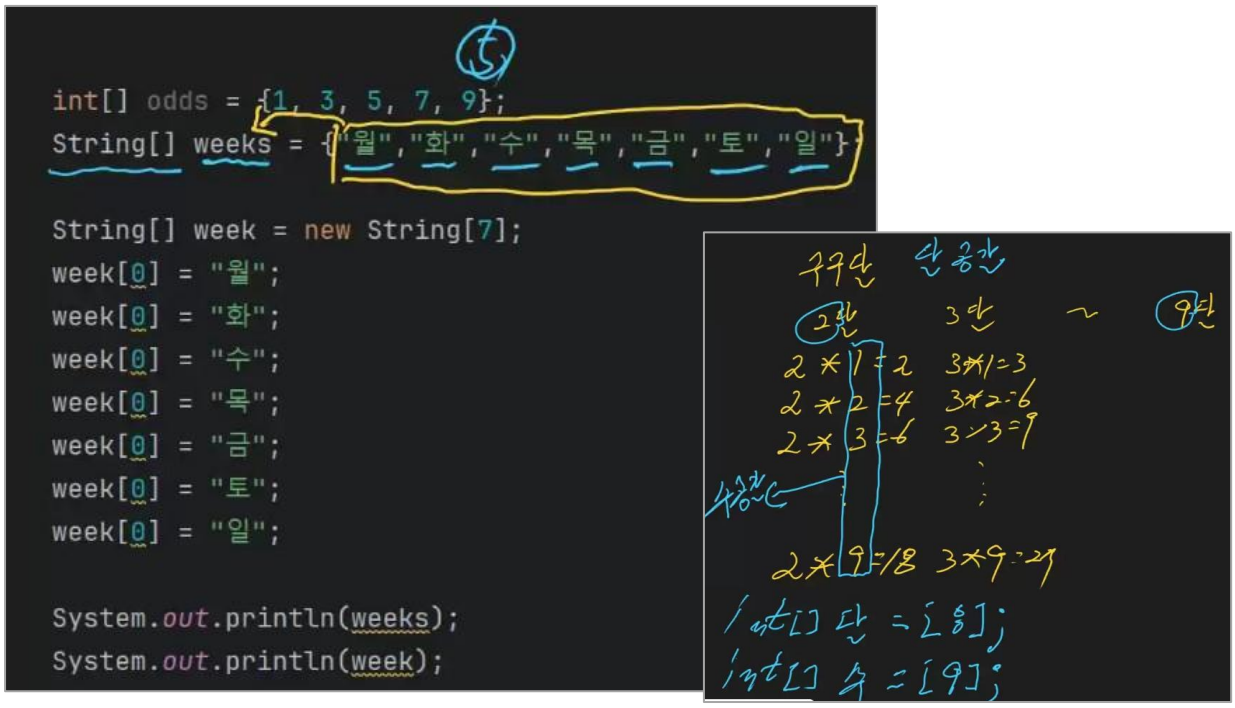
String result : result를 문자열로 바꿀거야

toString(); 배열로 입력한 값을 문자열로 바꿔주는 것

=> 빈배열만들면 메모리 소모가 크다

배열 문자열로 바꾸고 문자열을 다시 배열로 바꿈-> 덩치가 크다

~~배열 다음에 HashMap?~~



배열 : 저장되는 공간이 하나씩 위치값 갖는 것, 저장하는 공간이다.

String[] week = new String[7]; 배열 7개 설정함(0번~6번) , 배열을 오브젝트로 받아왔다 -> 배열의 주소에 따라 해당 하는 값을 넣어줬다

(String[7]자리에 빈 배열String[ ]로 주지 않기)

---이해한 방식으로 정리하기

int라는 배열을 선언, 그 이름은 odds고, 거기에 값을 1,3,5,7,9로 주었다

String이라는 문자열 배열을 선언, 그 이름은 weeks고, 거기에 값을 월화수목금토일로 주었다(1차원배열)

그리고 똑같이 String이라는 문자열 배열을 줬는데 그것은 객체로 new를 넣어서 7개로 선언해주었다 -> 값은 주지 않음

값은 배열로 0부터 6까지 각각 넣어줌

```
System.out.println(Arrays.toString(week)); //문자열로 바꿔줘야함
System.out.println(week[6]);
```

문자열로 바꿔줘야 배열이 출력됨(출력되는 방법 중 하나)

궁금한점 : string[7]로 줬는데 덜입력하거나 더 입력하는경우는?

-> 덜입력할 경우 null로 나오고, 더 입력하는경우 오류가 난다