

탈잉 개발자와 프로그래밍 무작정 배워보기



1. Java 란 무엇인가?

강대규

목차



1. Java에 대해서 간단하게 알아보자
2. JDK, Eclipse 를 설치하고 HelloWorld 를 출력해보자
3. 자료형, 변수, 연산자 등에 대해서 알아보자
4. 기타

1. Java에 대해서 간단하게 알아보자

1. Java에 대해서 간단하게 알아보자

프로그래밍 언어



컴퓨터 시스템을 구동시키는
소프트웨어를 작성하기 위한 언어

Java



1995년, 제임스 고슬링과 그의 팀원에 의해서 시작된 프로젝트.

본래 가전제품을 제어하기 위한 언어로 고안되었지만

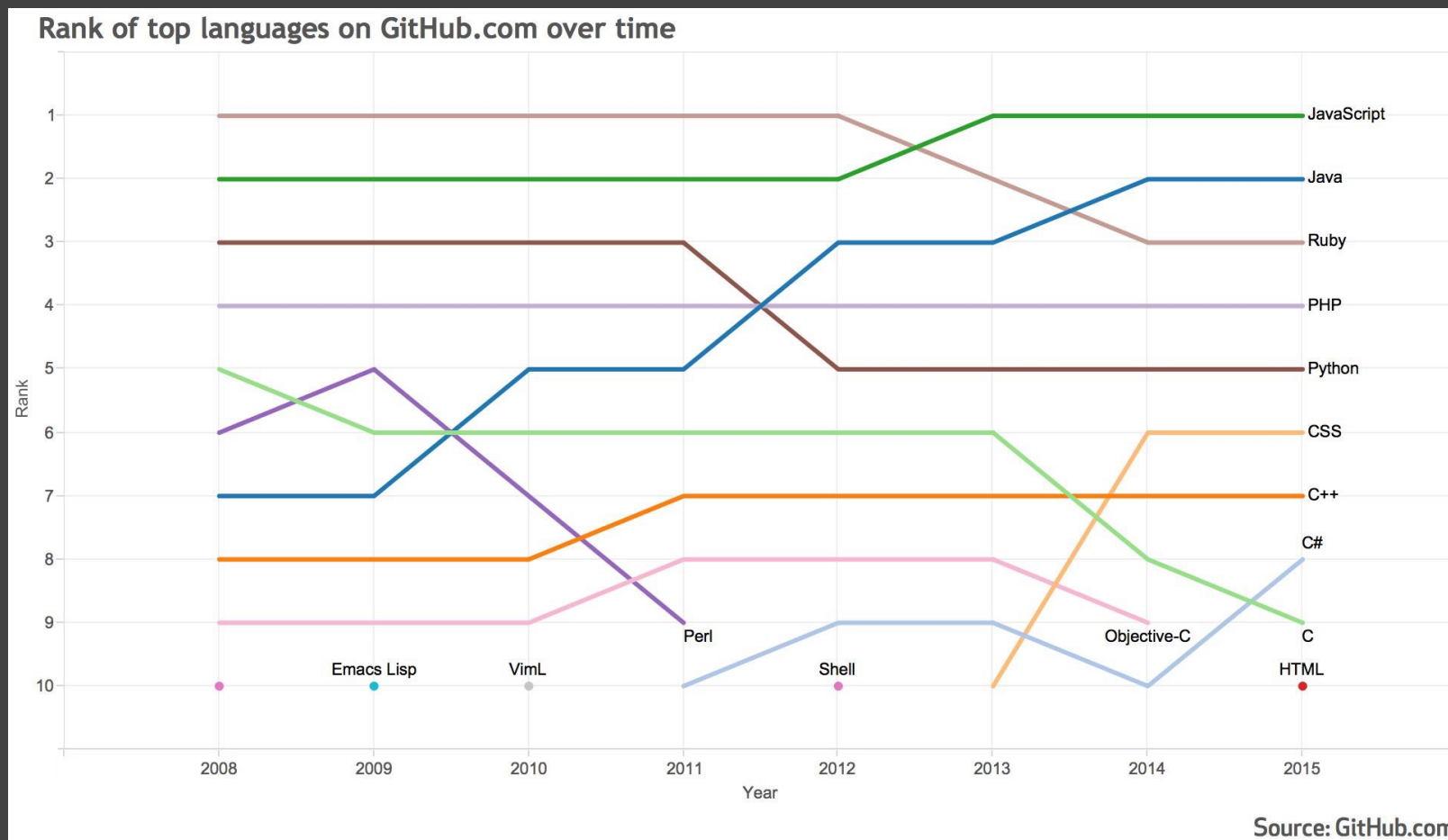
웹의 등장으로 엄청난 성공을 거두면서 주류 언어가 되었다.

Why Java ?



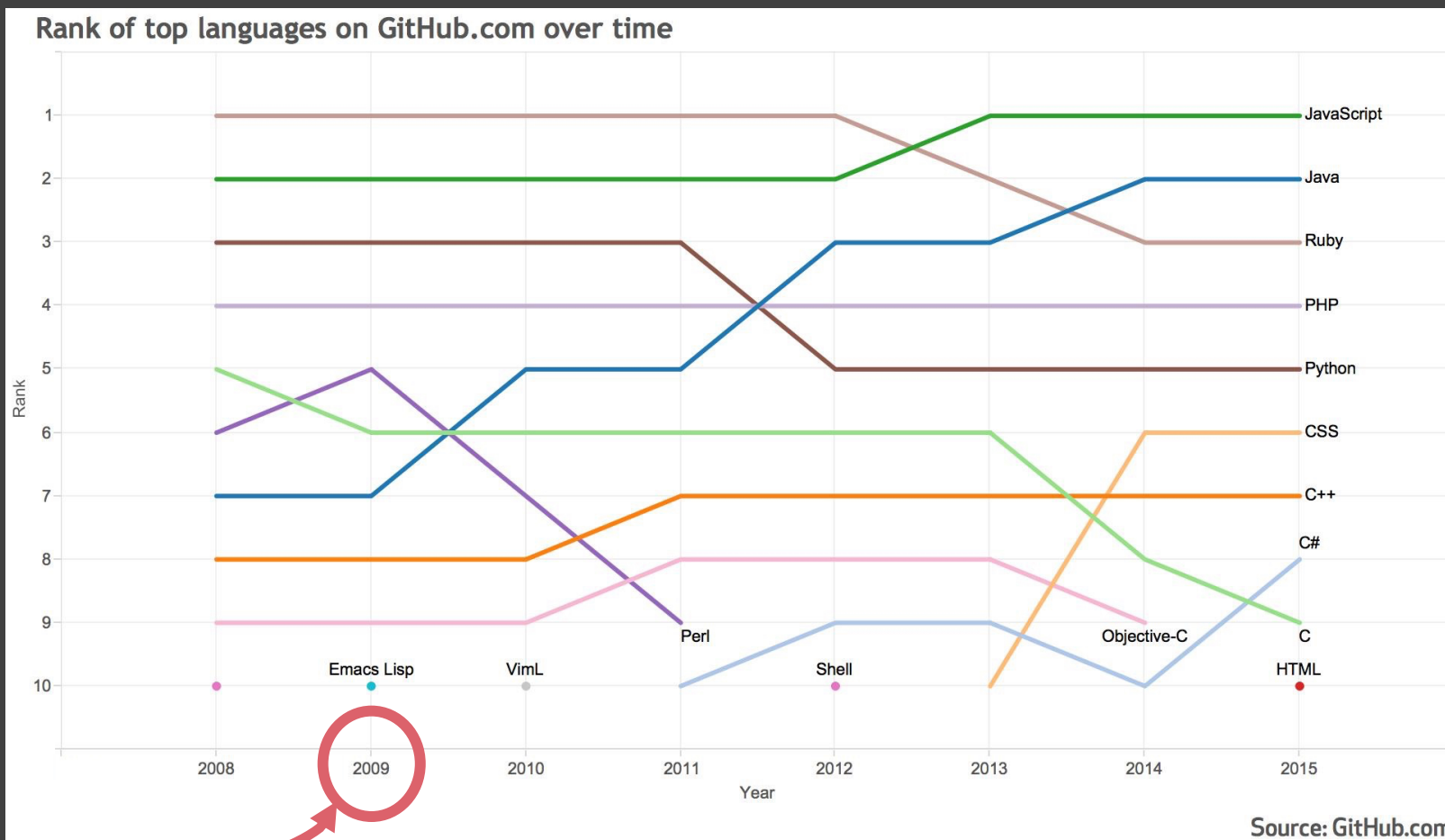
1. 전세계적으로 쓰이는 프로그래밍 언어이다.

Java



Java

Android의 등장

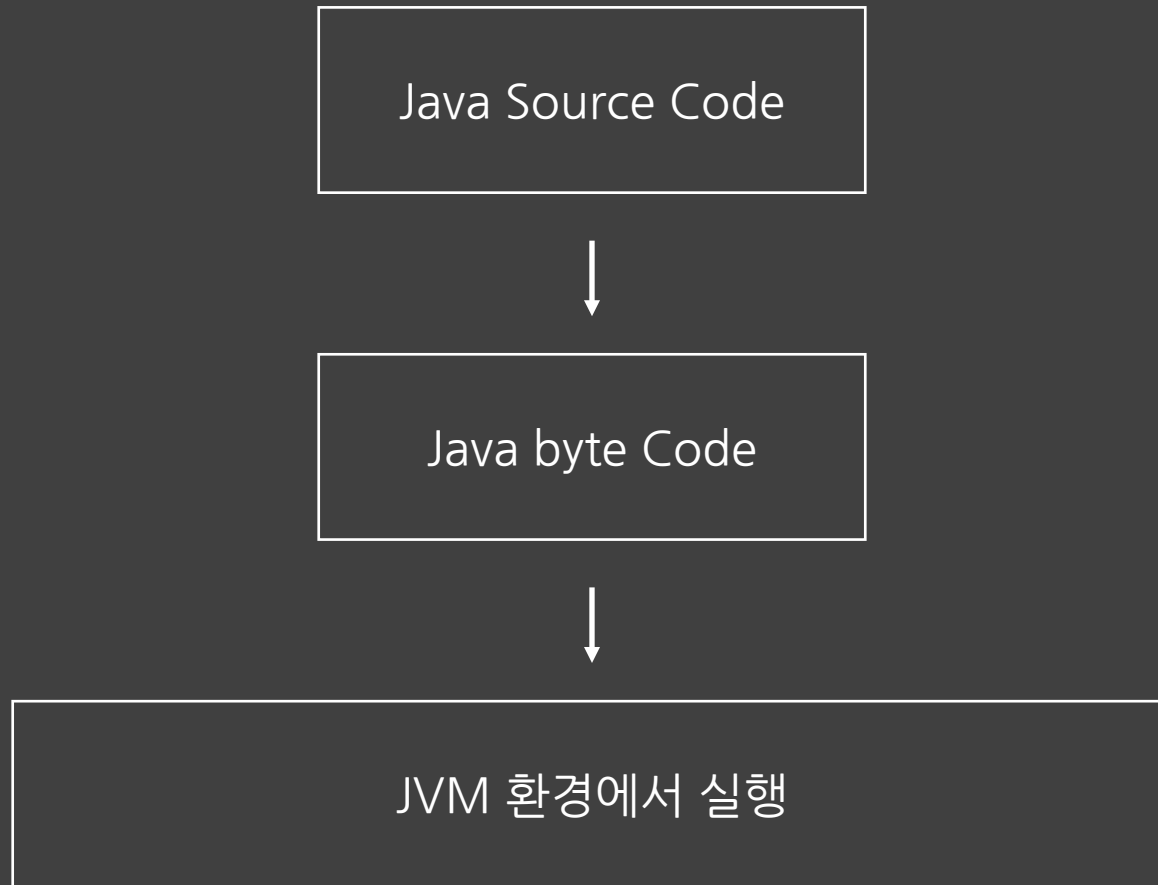


Why Java ?

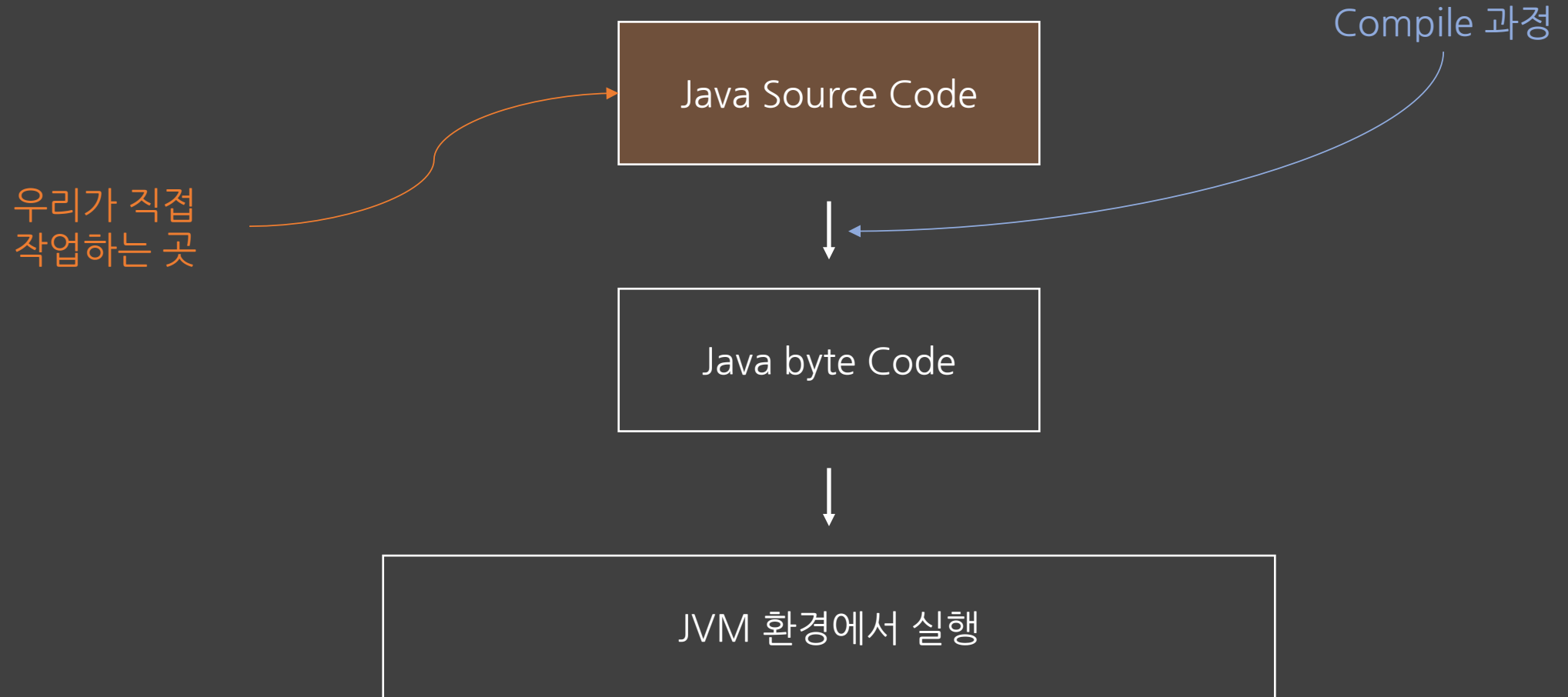


1. 전세계적으로 쓰이는 프로그래밍 언어이다.
2. JVM 환경에서 실행되어 어느 운영체제에서든지 동작가능하도록 할 수 있다.

Java 의 실행



Java 의 실행



Why Java ?



1. 전세계적으로 쓰이는 프로그래밍 언어이다.
2. JVM 환경에서 실행되어 어느 운영체제에서든지 동작가능하도록 할 수 있다.
3. Android / Server 개발에 필수적인 언어이다.

Why Java ?

1. 전세계적으로 쓰이는 프로그래밍 언어이다.
2. JVM 환경에서 실행되어 어느 운영체제에서든지 동작가능하도록 할 수 있다.
3. Android / Server 개발에 필수적인 언어이다.
4. 현재도 Oracle 사에서 꾸준히 업데이트 되고있다.

Why Java ?

1. 전세계적으로 쓰이는 프로그래밍 언어이다.
2. JVM 환경에서 실행되어 어느 운영체제에서든지 동작가능하도록 할 수 있다.
3. Android / Server 개발에 필수적인 언어이다.
4. 현재도 Oracle 사에서 꾸준히 업데이트 되고있다.
5. 객체지향 프로그래밍 언어의 선두주자이다.

2. JDK, Eclipse 를 설치하고 HelloWorld 를 출력해보자



1. JDK 다운로드 및 설치

JDK (Java Development Kit)

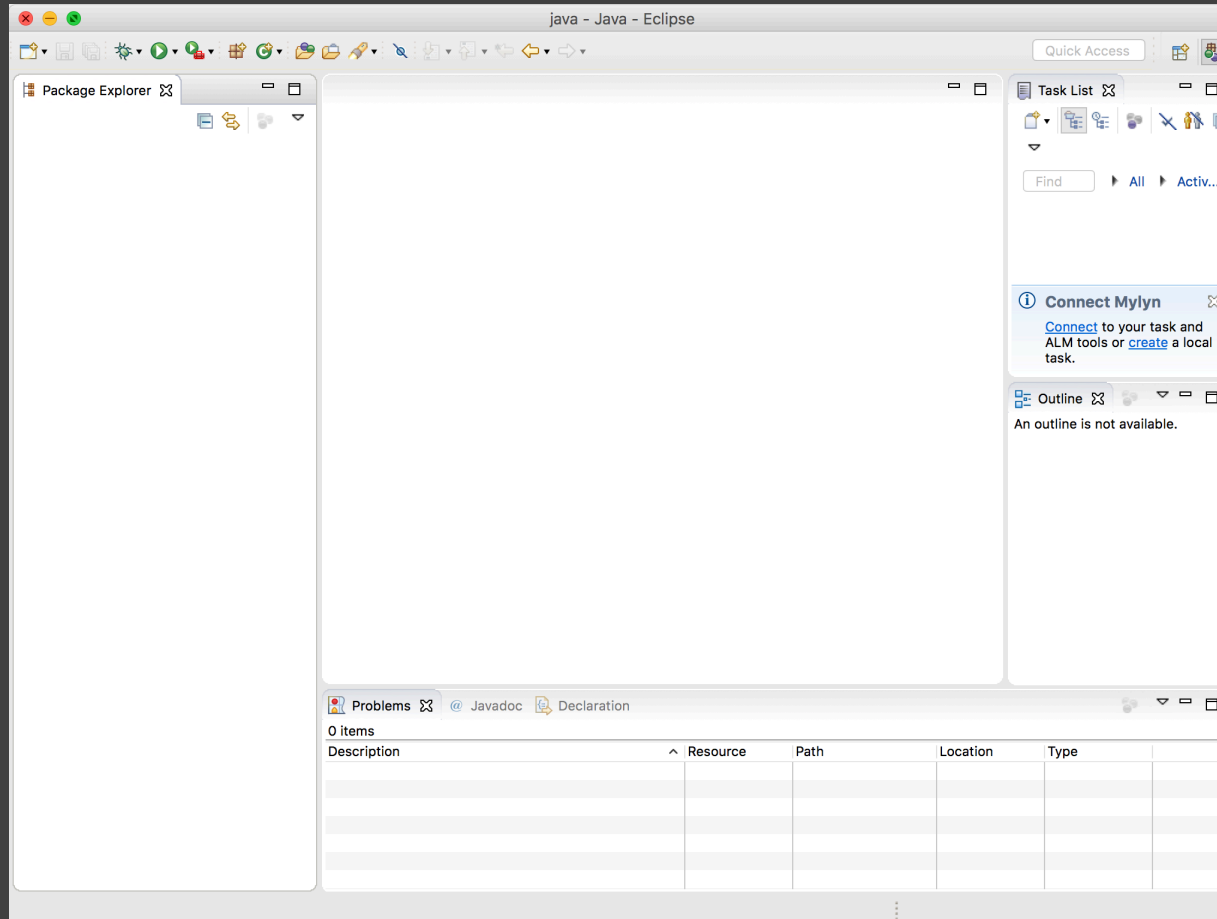
[JDK 8 download](#) (**Java SE Development Kit 8u131**)

3. Eclipse 다운로드 및 설치

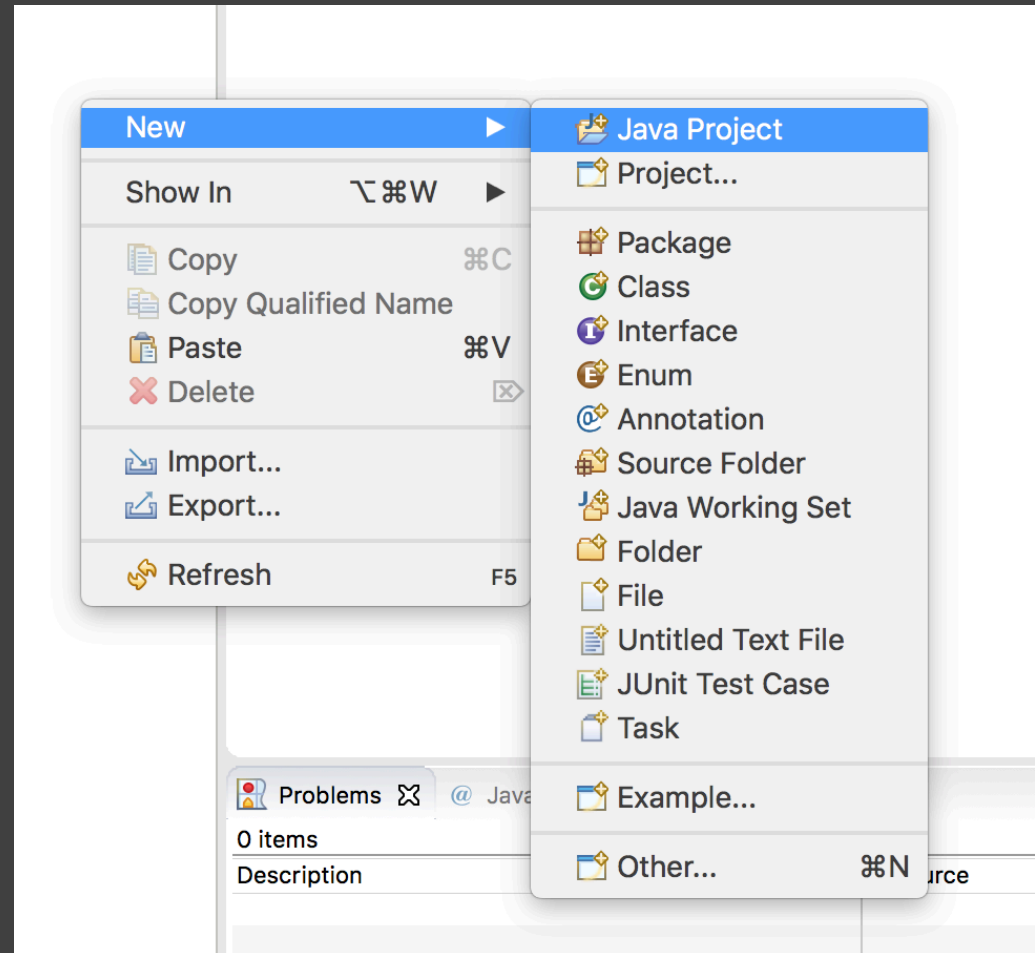
Eclipse (Editor)

[Eclipse Oxygen download](#)

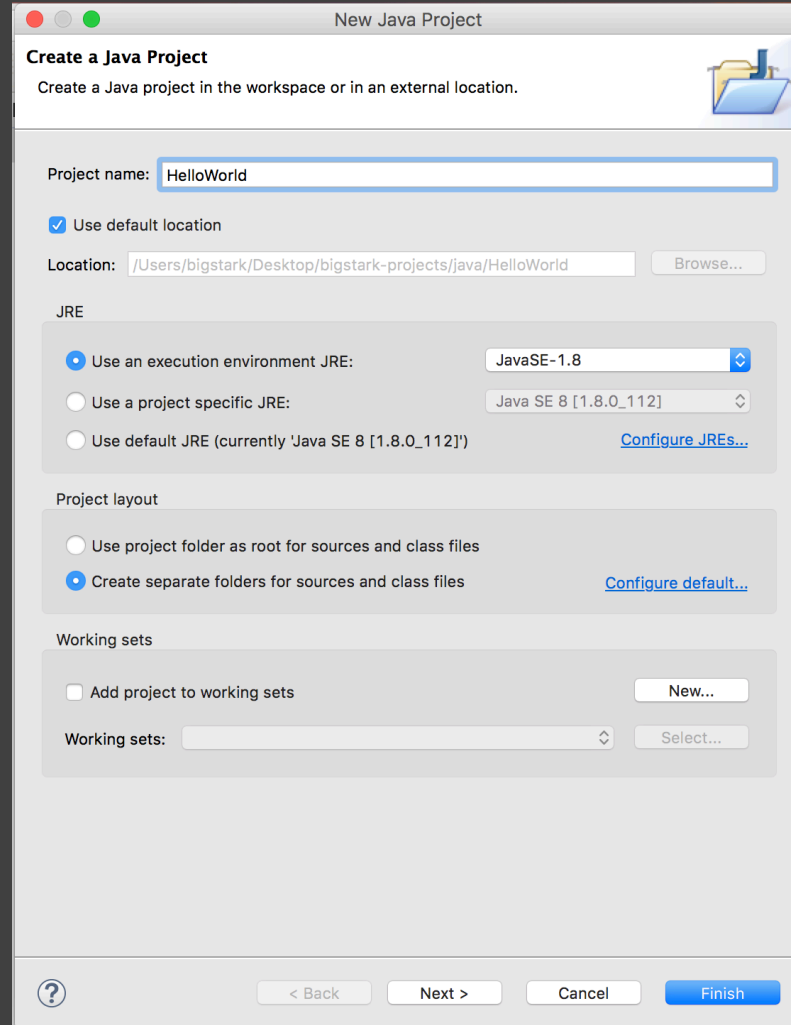
4. Eclipse 실행 & 프로젝트 생성



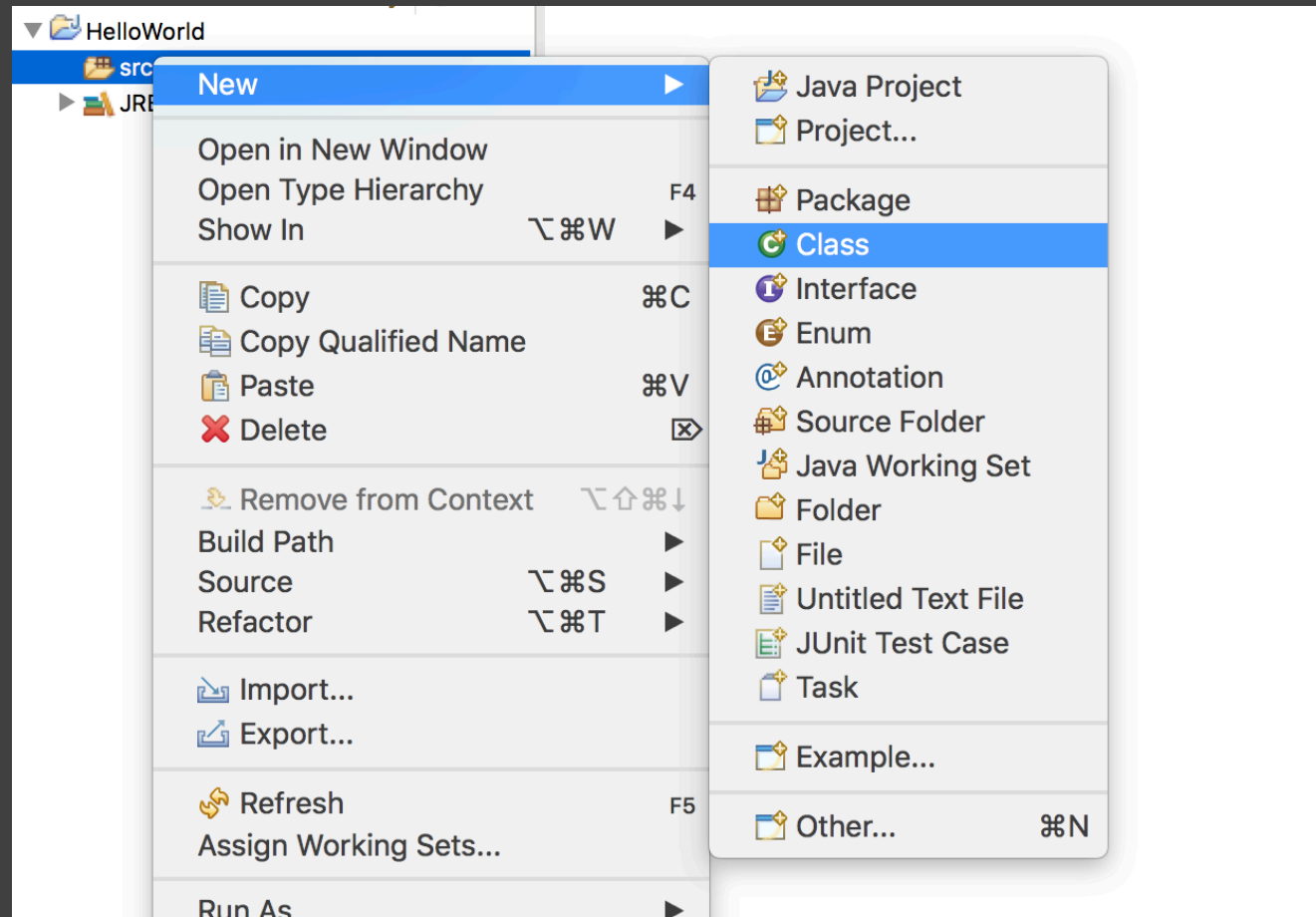
4. Eclipse 실행 & 프로젝트 생성



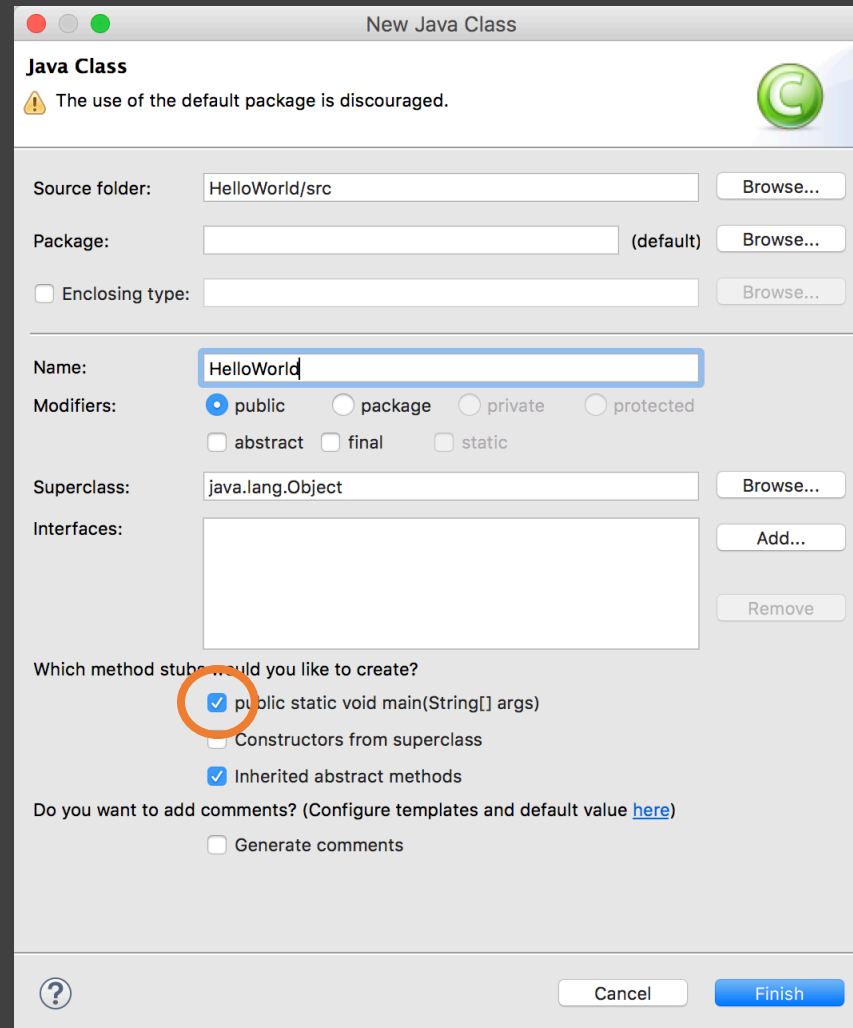
4. Eclipse 실행 & 프로젝트 생성



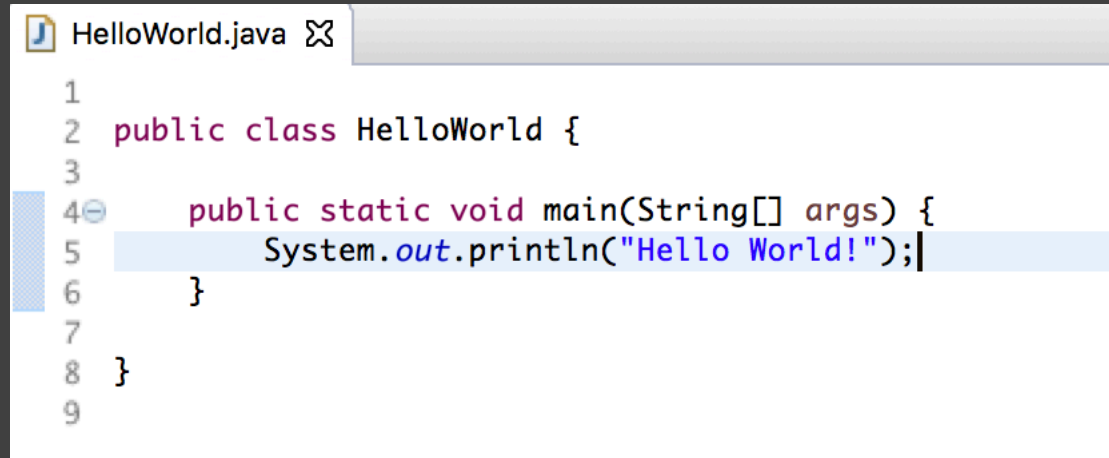
4. Eclipse 실행 & 클래스 파일 생성



4. Eclipse 실행 & 클래스 파일 생성



4. Eclipse 실행 & HelloWorld 출력

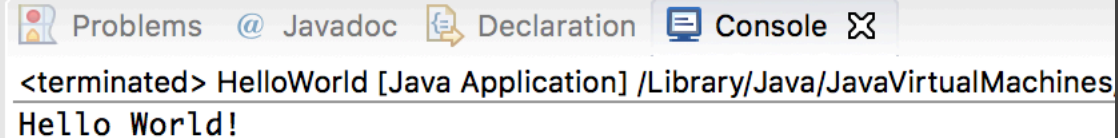
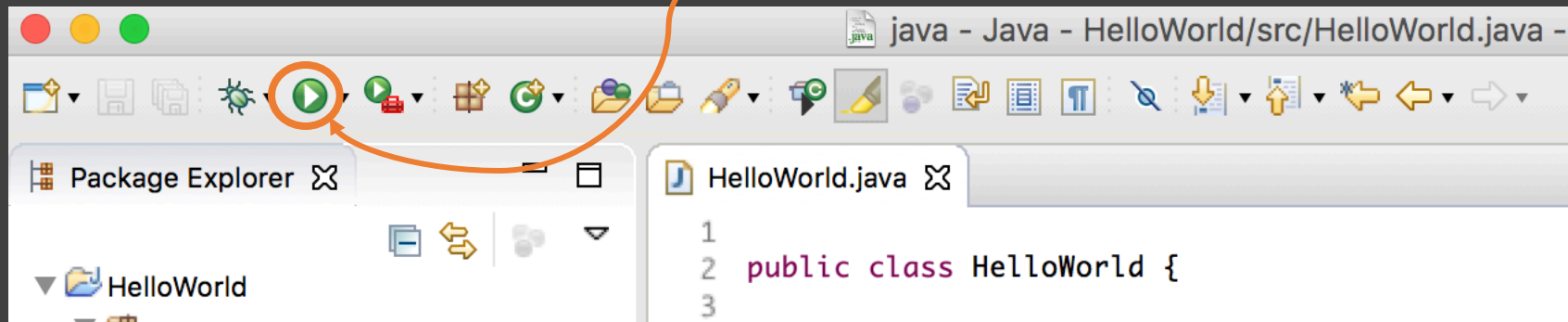


```
1
2 public class HelloWorld {
3
4 public static void main(String[] args) {
5     System.out.println("Hello World!");
6 }
7
8 }
9
```

System.out.println("Hello World!"); 을 써보자

4. Eclipse 실행 & HelloWorld 출력

Run 버튼 클릭



The screenshot shows the Eclipse IDE Console window. The Console tab is selected, and the output shows the following text:

```
<terminated> HelloWorld [Java Application] /Library/Java/JavaVirtualMachines  
Hello World!
```

결과값 확인!

3. 자료형, 변수, 연산자 등에 대해서 알아보자



몇 가지 규칙들

세미콜론 : Java에서는 문장의 끝에 “;”(세미콜론) 을 붙여주지 않으면, 컴파일 에러가 난다.

이름규칙

- 영어, 숫자, _ 만을 쓴다. (한글도 가능하지만 쓰지 않으며, 띄어쓰기도 사용하지 않는다.)
- 이름의 맨 앞에 숫자가 올 수는 없다.

코드는 중괄호 안에서 작성해야한다.

자료형과 변수(Data Type and Variable)

변수 : 필요한 데이터를 저장하고, 그 데이터를 불러들일 때 쓰는 메모리 공간

자료형 : 변수에 데이터를 저장할 때의 변수 종류

자료형과 변수(Data Type and Variable)

```
int a;
```

```
a = 10;
```

자료형과 변수(Data Type and Variable)

자료형

int a;

a = 10;

변수

자료형과 변수(Data Type and Variable)

변수 : 필요한 데이터를 저장하고, 그 데이터를 불러들일 때 쓰는 메모리 공간

자료형 : 변수에 데이터를 저장할 때의 변수 종류

- 논리형 (true or false)
- 문자형
- 정수형
- 실수형

자료형과 변수(Data Type and Variable)

```
boolean a = true;
```

```
String b = "hello World";
```

```
char c = 'a';
```

```
int d = 10;
```

```
short e = 10;
```

```
long f = 10;
```

```
float g = 3.5f;
```

```
double h = 3.5;
```


연산자

어떠한 대상 처리를 수행하도록 하는 특수한 문자 기호

산술연산자 : +, -, *, /, %

대입연산자 : =

비교연산자 : <, <=, ==, >, >=

논리연산자 : &&, ||

단항연산자 : +, -, ++, --, !

연산자 - 산술연산자

+ : 더하기

- : 빼기

* : 곱하기

/ : 나누기

% : 나머지

```
int a = 10;
```

```
int b = 2;
```

```
System.out.println(a + b);
```

```
System.out.println(a - b);
```

```
System.out.println(a * b);
```

```
System.out.println(a / b);
```

```
System.out.println(a % b);
```

연산자 - 산술연산자

+ : 더하기, 문자형에서는 문자를 붙이는 의미로 쓰인다.

```
String a = "Hello ";
```

```
String b = "World";
```

```
System.out.println(a + b);
```

연산자 - 대입연산자

$(\text{변수}) = (\text{값})$

왼쪽 변수에 오른쪽 값을 넣으라는 연산자

연산자 - 비교연산자

좌변과 우변을 비교하여, 해당 연산자의 작동과 맞으면 true이고, 아니면 false 이다.

< : 좌변이 우변보다 작으면 true

<= : 좌변이 우변보다 작거나 같으면 true

== : 좌변과 우변이 같으면 true

!= : 좌변과 우변이 같지 않으면 true

> : 좌변이 우변보다 크면 true

>= : 좌변이 우변보다 크거나 같으면 true

연산자 - 논리연산자

좌변과 우변을 논리연산한다.

&& : and 연산으로, 좌변과 우변이 모두 true 면 true, 아니면 false 이다.

|| : or 연산으로, 좌변이나 우변 둘 중 하나가 true면 true, 아니면 false 이다.

```
boolean a = true;
```

```
boolean b = false;
```

→ a && b 는 false, a || b 는 true 가 된다.

연산자 - 단항연산자

하나의 항을 대상으로 이루어지는 연산

+ : 양수를 표현

- : 음수를 표현

++ : 값을 1 증가한다.

-- : 값을 1 감소한다.

! : 논리값을 부정한다. (true -> false, false -> true)

연산자 실습

```
int a = 10;  
int b = 20;
```

```
int c = a + b;  
int d = a * b;
```

```
boolean e = true;  
boolean f = false;
```

```
boolean g = e && f;  
boolean h = e || f;  
boolean l = c > d;
```


연산자 실습

```
boolean a = true;  
boolean b = !a;
```

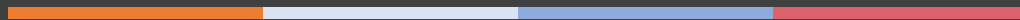
```
int c = 10;  
int d = -c;
```

연산자 실습 (a++ vs ++a)

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

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

4. 기타



기타 - 변수 바꾸기

* a 변수의 값과 b 변수의 값을 바꾸고 싶다.

ex)

```
int a = 30;
```

```
int b = 20;
```

기타 - 변수 바꾸기

* a 변수의 값과 b 변수의 값을 바꾸고 싶다.

ex)

```
int a = 30;
```

```
int b = 20;
```

```
a = b;
```

```
b = a;
```

기타 - 변수 바꾸기

* a 변수의 값과 b 변수의 값을 바꾸고 싶다.

ex)

```
int a = 30;
```

```
int b = 20;
```

~~a = b;~~

~~b = a;~~

$a \leftarrow 20, b == 20$

$b \leftarrow 20, a == 20$

즉 a 에 b 를 대입하는 순간부터
a와 b가 같게되어 의미가 없어짐

기타 - 변수 바꾸기

* a 변수의 값과 b 변수의 값을 바꾸고 싶다.

ex)

```
int a = 30;
```

```
int b = 20;
```

해결책은 임시 변수를 만들어
값을 저장해두는 것!

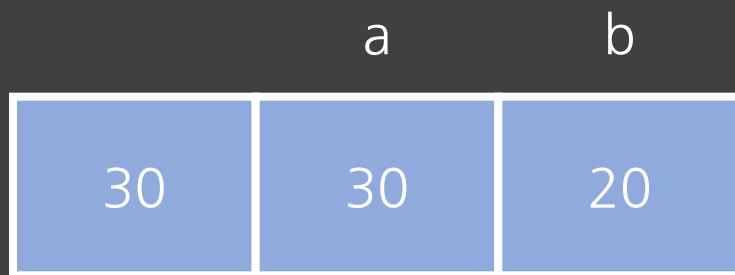
기타 - 변수 바꾸기

* a 변수의 값과 b 변수의 값을 바꾸고 싶다.

| | a | b |
|--|----|----|
| | 30 | 20 |

기타 - 변수 바꾸기

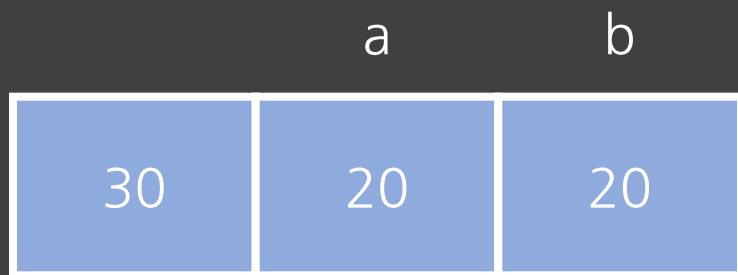
* a 변수의 값과 b 변수의 값을 바꾸고 싶다.



a값을 복사

기타 - 변수 바꾸기

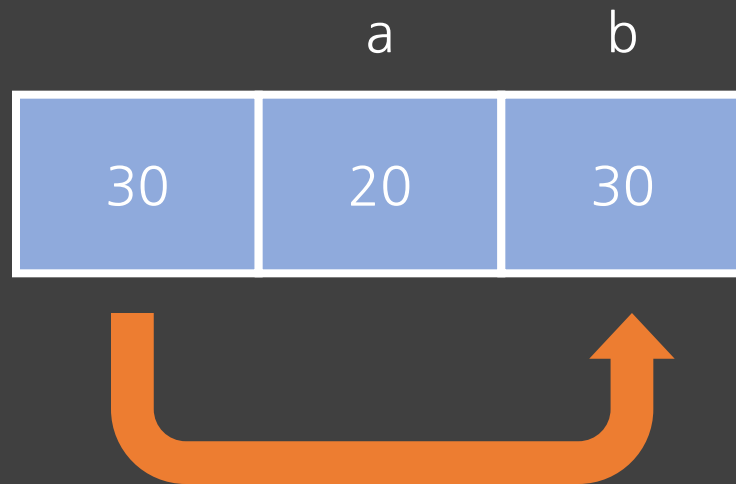
* a 변수의 값과 b 변수의 값을 바꾸고 싶다.



b값을 a에 대입

기타 - 변수 바꾸기

* a 변수의 값과 b 변수의 값을 바꾸고 싶다.



저장해둔 a 값을 b에 대입

기타 - 변수 바꾸기

* a 변수의 값과 b 변수의 값을 바꾸고 싶다.

ex)

```
int a = 30;
```

```
int b = 20;
```

```
int temp = a;
```

```
  a = b;
```

```
  b = temp;
```

기타 - String format

예를들어 $10 \times 20 = 10 * 20$ 을 출력한다고 해보자.

- `System.out.println("10 x 20 = 10 * 20");`

기타 - String format

예를들어 $10 \times 20 = 10 * 20$ 을 출력한다고 해보자.

- `System.out.println("10 x 20 = 10 * 20");`

하지만 여기에서 10 과 20 이 변수라면?

→ $a \times b = a * b$ 를 출력해야한다.

기타 - String format

String 의 산술연산자 + 는 문자를 붙이는 연산을 하므로

```
int a = 10;
```

```
int b = 20;
```

```
System.out.println(a + " x " + b + "=" + (a * b));
```

기타 - String format

String 의 산술연산자 + 는 문자를 붙이는 연산을 하므로

```
int a = 10;
```

```
int b = 20;
```

```
System.out.println(a + " x " + b + "=" + (a * b));
```

너무 복잡하다!

기타 - String format

String format 을 사용하자!

```
String.format("%d x %d = %d", 변수들);
```

기타 - String format

String format 을 사용하자!

```
String.format("%d x %d = %d", 변수들);
```

```
System.out.println(String.format("%d x %d = %d", a, b, a*b));
```

기타 - String format

%d : 정수형 변수를 표현할 때 (int)

%s : 문자열 변수를 표현할 때 (String)

%c : 문자 변수를 표현할 때 (char)

%b : 논리형 변수를 표현할 때 (boolean)

...

기타 - Scanner

```
Scanner scanner = new Scanner(System.in);  
    System.out.println("숫자를 입력하세요 :");  
        int a = scanner.nextInt();  
System.out.println("입력받은 숫자는 : " + a);
```

기타 - Scanner

```
Scanner scanner = new Scanner(System.in);  
System.out.println("숫자를 입력하세요 :");  
    int a = scanner.nextInt();  
System.out.println("입력받은 숫자는 : " + a);
```

→ 내가 직접 숫자를 입력할 수 있게 하는 것