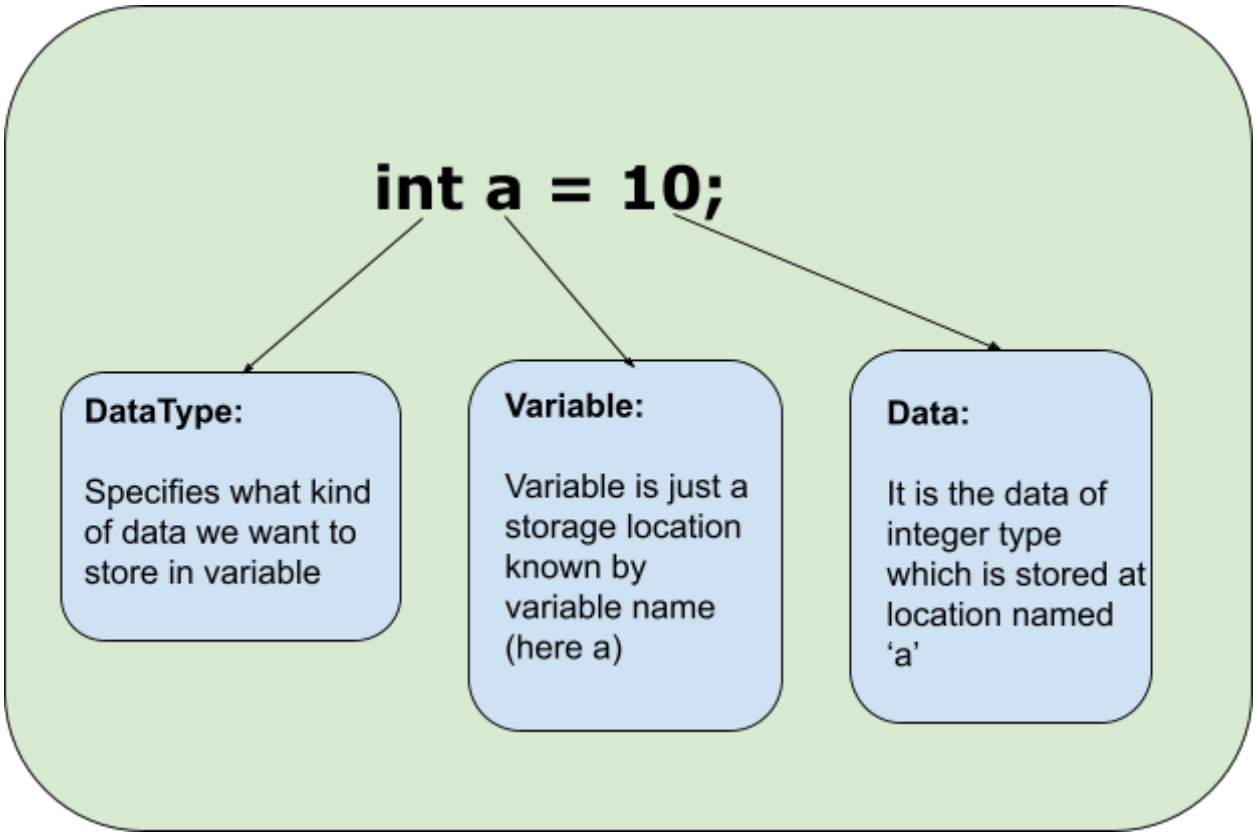
Java is Platform Independent

JVM
JRE
Compiled
Interpreted

Variables & DataTypes



Storing integral data

byte	1 byte
short	2 byte
int	4 byte
long	8 byte

Storing decimal numbers

float	4 bytes
double	8 bytes

Storing characters

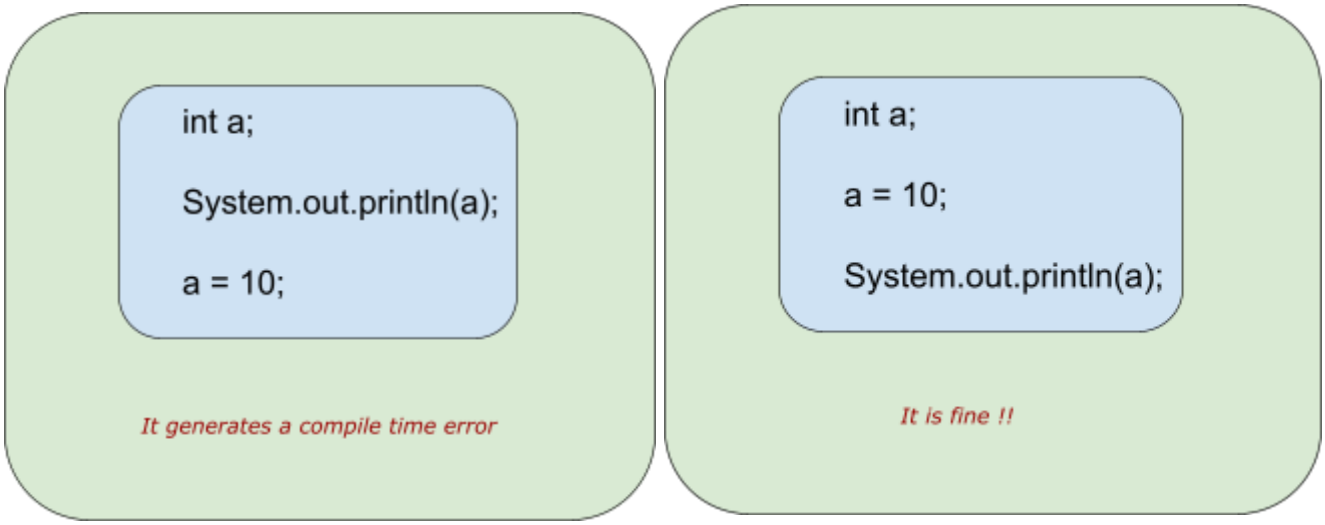
char	2 bytes

Storing true/false

boolean	Not defined

Ground Rules while working with variables

Rule 1: **you can not use a variable before initializing it.**



!!!! the java compiler never assigns a default value to uninitialized local variable

That's why : declare -> initialize -> use

Rule 2: **variable naming rules**

- All variable names may contain uppercase and lowercase letters (a-z, A-Z), underscore(_), dollar sign ($) and the digits 0 to 9. The dollar sign character is not intended for general use. No spaces and no other special characters are allowed.
- The Variable Names Must Not Begin With A Number.
- Java is case-sensitive. Uppercase characters are distinct from lowercase Characters.
- A Java keyword(reserved word)cannot be used as a variable name.

Arithmetic Operators & Precedence

Precedence	Operators	Associativity
high	*, /, %	Left to right
low	+, -	Left to right

Taking Input

🔥 The Java Scanner class breaks the input into tokens using a delimiter that is whitespace by default. It provides many ways to read and parse various primitive values.

In order to use scanner you have to write this import statement at the top –
import java.util.Scanner;

Example Code:

```
//Code for adding two integers entered by the user import java.util.Scanner;
class AddTwoNumbers
{
public static void main(String args[]) {
int a, b, c;
System.out.println("Enter two integers to calculate their sum: ");
// Create a Scanner
Scanner s = new Scanner(System.in); a = s.nextInt();
b = s.nextInt();
c = a + b;
System.out.println("Sum of entered integers = "+c);
}
}
```

Sample Input: 10 5
Output:
15

Here, s.nextInt() scans and returns the next token as int.

A token is part of an entered line that is separated from other tokens by space, tab or newline.

So when the input line is: "10 5" then s.nextInt() returns the first token i.e. "10" as int and s.nextInt() again returns the next token i.e. "5" as int.

Taking character input

To read a character as input, we use `next().charAt(0)`.

The next() function returns the next token in the input as a string and charAt(0) function returns the first character in that string.

Example code to read a character as input:

```
import java.util.Scanner;
public class ScannerDemo1 {

public static void main(String[] args) {
Scanner s = new Scanner(System.in);
char ch = s.next().charAt(0); // character input
System.out.println("input character = " +ch);
}
}
```

Sample Input:
k
Output:
input character = k

Example code to take a string as input:

```
public static void main(String[] args)
{
Scanner s = new Scanner(System.in); String str;
str = s.next();
System.out.print(str);
}
```

Sample Input:
Coding Ninjas

Output:
Coding

Here, s.next() returns the next token as String. So when input line is - “Coding Ninjas” then s.next() returns the first token i.e. “Coding”.