—

static attributes

static <type> <varName>

Declaring Variables

static byte age = 30;

—

instance attributes

<type> <varName>

Declaring Variables

byte age = 30;

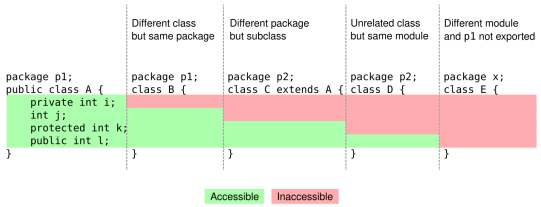
long viewsCount = 3123456L;

float price = 10.99F;

char letter = ‘A’;

boolean isEligible = true;

access



Simple main

```
public class Main  {

  public static void main(String[] args)  {

  }

}
```

Output

System.out.println("Hello World!");
System.out.print("Hello World!");

System.out.printf("Hello World!"); //this can run format แบบในไพท่อน เช่น

System.out.printf("Test %d Test %s", 5, “this Test”);

//output is : Test 5 Test this Test

\n	New Line
\r	Carriage Return
\t	Tab
\b	Backspace
\f	Form Feed

String.format ใช้แบบปริ้น f แต่จะไม่ปริ้น แต่รวมไว้เป็นสตริง

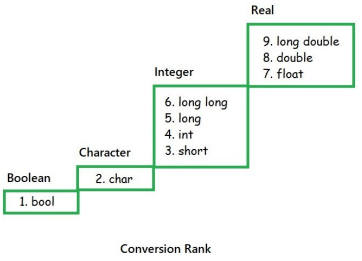
Java Type Casting

To convert a string to a number, we use one of the following methods:

• Byte.parseByte(“1”)
• Short.parseShort(“1”)
• Integer.parseInt(“1”)
• Long.parseLong(“1”)
• Float.parseFloat(“1.1”)
• Double.parseDouble(“1.1”)

java implicit casting

ตัวอย่าง



long = 100000l;

int var;

u can do this : var = (int) long;

Reading Input

Scanner scanner = new Scanner(system.in);

double number = scanner.nextDouble();
byte number = scanner.nextByte();
String name = scanner.next();
String line = scanner.nextLine();
.close(); ระวังลืม

Comparison Operators

== , !=, <, >, <=, >=

Logical Operators

• x && y (AND): if both x and y are true, the result will be true.

• x || y (OR): if either x or y or both are true, the result will be true.

• !x (NOT): reverses a boolean value. True becomes false.

Array Class

Methods ที่น่าใช้

asList() Returns a fixed-size list backed by the specified Arrays

compare(array 1, array 2) Compares two arrays passed as parameters lexicographically.

copyOf(originalArray, newLength) Copies the specified array, truncating or padding with the default value (if necessary) so the copy has the specified length.

copyOfRange(originalArray, fromIndex, endIndex) Copies the specified range of the specified array into a new Arrays.

equals(array1, array2) Checks if both the arrays are equal or not.

fill(originalArray, fillValue) Assigns this fill value to each index of this arrays.

setAll(originalArray, functionalGenerator) Sets all the elements of the specified array using the generator function provided.

sort(originalArray) Sorts the complete array in ascending order.

sort(originalArray, fromIndex, endIndex) Sorts the specified range of array in ascending order.

sort(T[] a, Comparator< super T> c) Sorts the specified array of objects according to the order induced by the specified comparator.

Java ArrayList Methods

ArrayList<type> var = new ArrayList<type>();

ArrayList<Class> var = new ArrayList<Class>();
สามารถใส่ Class ใน ArrayList ได้

KEY WORD

package
import java.util.*;

- ArrayList
- Collections
- Comparator
- Arrays
- Scanner

import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.time.temporal.ChronoUnit;

enum **<name>** {}

class <name> extends <Superclass> implements class1, class2
interface class1{} !ระวังถ้าประกาศตัวแปรในนี้จะเป็น final
interface class2{} !ระวังถ้าประกาศตัวแปรในนี้จะเป็น final
abstract class subclass จะต้องมี con ที่กำหนด

toString
super.
this.

import java.util.Comparator;
จากห้องเรียน

```
public class FavoriteCourseComputer implements
Comparator<Student23> {

   @Override

   public int compare(Student23 o1, Student23 o2) {

      return
o1.favoriteCourse.compareTo(o2.favoriteCourse) ;

   }

}
```

Exception Handling

```
try {

} catch (ExceptionType e1) {

} catch (Exception e2) {

} finally {}
```

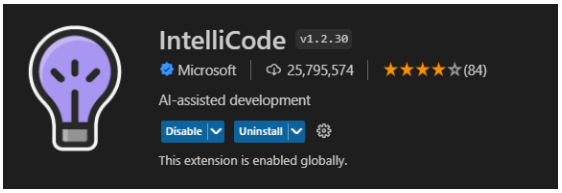
Arrays key word

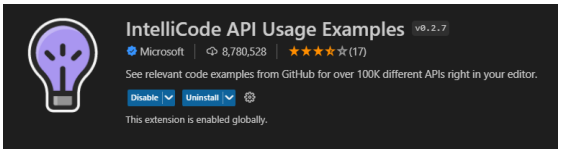
- <type> [] <varName>;
- <type> [] <varName> = {“val”, “val, “val”, “val”};
- int x = myValues [1][2];
- int[][]myValues = { {1,2,3,4}, {5,6,7}};
- **.clone();**
- **.length; !no()**
- .equals(another array); !return bool
- Arrays.toString(array); !return no void

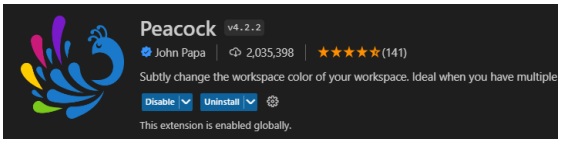
Arrays key word

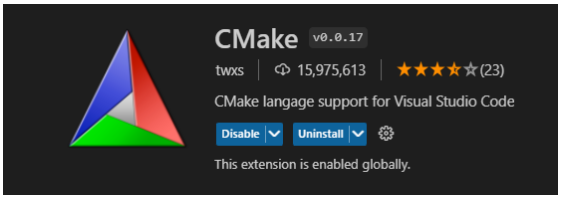
- <> เขียนด้วยถ้าสีแดง
- `Arraylist<type> name= new Arraylist<type>();`
- `name.add(val);`
- `name.set(<index>,val);`
- `name.size();`
- `name.toString();`

แนะนำงับ









Array vs ArrayList

Array	ArrayList
length predefined	size() variable
manual shift	Auto shift
faster	slower
primitive+Ref type	ref type only
equals() not overriden	overriden equals()
toString() not overriden	overriden toString()
Arrays.sort()	Collections.sort()
Arrays.binarySearch()	Collections.binarySearch()

Java String Methods

s.length()	length of s
s.charAt(i)	extract ith character
s.substring(start, end)	substring from start to end-1
s.toUpperCase()	returns copy of s in ALL CAPS
s.toLowerCase()	returns copy of s in lowercase
s.indexOf(x)	index of first occurence of x
s.replace(old, new)	search and replace
s.split(regex)	splits string into tokens
s.trim()	trims surrounding whitespace
s.equals(s2)	true if s equals s2
s.compareTo(s2)	0 if equal/+ if s > s2/- if s < s2

JAVA - ArrayList - Cheat Sheet

`ArrayList<String>listName = new ArrayList<String>();`	*Declaring an ArrayList*
`ArrayList<String> listName = new ArrayList<String>(5);`	*Declaring an ArrayList with specific index size (5)*
`listName.add("penguin");`	*Adding to ArrayList*
`listName.remove(0);`	*//removes index [0]*
`listName.remove("penguin");`	*//removes string penguin wherever it is*
`listName.set(0, "tux");`	*Replacing an existing Item in ArrayList*
`listName.size();`	*Checking the Size (how many indexes)*
`listName.indexOf(item)`	*Searching under which index is (item)?*
`int index? = myArrayList.indexOf("penguin");`	
`listName.contains(item);`	*Verifying Contents*
`if( myArrayList.contains("penguin") )`	*(is there an item with such and such name or value)*
`listName.isEmpty();`	*Checking if Empty*
`while( myArrayList.isEmpty() );`	
`newListName.addAll(listName);`	*copy the contents of an existing ArrayList to the new one.*
`ArrayList<String> copyArrayList = new ArrayList<String>();`	
`copyArrayList.addAll(myArrayList);`	
`listName.clear();`	*Clearing an ArrayList*
`Collections.sort(listName);`	*Sorting an ArrayList*
`for( <type> varName : listName )`	*Outputing an ArrayList*
`System.out.println(varName);`	
`for( String ix : myArrayList )`	
`System.out.println(ix);`	
`for(Object ix : myArrayList )`	
`System.out.println(ix);`	
`listName.toArray(arrayName);`	*Conversion - ArrayList to an Array*
`String[] regArray = new String[myArrayList.size()];`	
`myArrayList.toArray(regArray);`	
`ArrayList listName = Array.asList(arrayName)`	*Array to an ArrayList*
`ArrayList<String> myArrayList = Arrays.asList(regArray);`	