

Numbers

```
static MAX_VALUE
xxxValue()
public int compareTo(XXX)
public boolean equals(Object o)
static Integer valueOf(int i)
static Integer valueOf(String s)
static Integer valueOf(String s, int radix)
String toString()
static String toString(int i)
static int parseInt(String s)
static int parseInt(String s, int radix)
```

Math

```
xxx abs(xxx x)
double ceil(double d) double ceil(float f)
double floor(double d) double floor(float f)
double int(double d)
long round(double d) int round(float f)
double exp(double d)
double log(double d)
double pow(double base, double exponent)
Double E, PI
```

Strings

```
char charAt(int index)
int indexOf(char ch)
String concat(String s)
String substring(int beginIndex)
String substring(int beginIndex, int endIndex)
String toLowerCase()
String toUpperCase()
boolean contains(CharSequence s)
static String copyValueOf(char[] data)
char[] toCharArray()
int length()
boolean contentEquals(StringBuffer sb)
boolean startsWith(String prefix)
boolean endsWith(String suffix)
boolean matches(String regex)
String intern()
String trim()
```

```
String replace(char oldChar, char newChar)
String replaceAll(String regex, String
replacement)
String replaceFirst(String regex, String
replacement)
String[] split(String regex)
String[] split(String regex, int limit)
static String valueOf(XXX)
```

Array & ArrayList

```
int[][] myArr = new int[5][10];
ArrayList<String> myArr=new
ArrayList<String>();
myArr.add("dynamically resizing");
System.out.println(myArr.get(0));
```

Vector

```
Vector<String> myVect=new Vector<String>();
myVect.add("synchronized");
System.out.println("myVect.get(0)");
```

LinkedList

```
LinkedList<String> myList=new
LinkedList<String>();
myList.add("build-in list");
Iterator<String> iter=myList.iterator();
while (iter.hasNext())
System.out.println(iter.next());
```

HashMap

```
HashMap<String,String> map=new
HashMap<String,String>();
map.put("key","value");
System.out.println(map.get("key"));
```

Regex

```
String pattern=".*(\\d+)";
Pattern r=Pattern.compile(pattern);
```

```
String line="lalala";
Matcher m=r.matcher(line);
if(m.find()) line=line.replaceAll("-", "");
```

Read with Scanner

```
Scanner sc=new Scanner(System.in);
while(sc.hasNextLine())
System.out.println(sc.nextLine());
```

Read with BufferedReader

```
String thisLine=null;
try {
    BufferedReader br = new
    BufferedReader("c:/test.txt");
    while ((thisLine = br.readLine()) != null)
    {
        System.out.println(thisLine);
    }
} catch(Exception e) {
    e.printStackTrace();
}
```

StdIn

```
boolean isEmpty()
int readInt()
boolean hasNextChar()
char readChar()
int[] readAllInts()
String[] readAllLines()
String readAll()
```

Arrays

```
public static void fill(int[] a, int val)
public static void sort(Object[] a)
```

Enums

```
boolean hasMoreElements()
Object nextElement()
```

Stacks

```
Object push(Object element)
Object pop()
Object peek()
boolean isEmpty()
```

Queues

```
Object offer()
Object poll()
boolean isEmpty()
```

Dictionaries

```
Object put(Object key, Object value)
Object get(Object key)
Enumeration elements()
Enumeration keys()
Object remove(Object key)
boolean isEmpty()
boolean contains(Object key)
int size()
```

StringBuilder

```
StringBuilder append(char c)
StringBuilder deleteCharAt(int index)
int length()
```

```
Collections.sort(arrayList)
```

by 甜菜欣欣

[微博](#) [知乎](#) [Github](#)
甜欣屋 : www.tcxx.info
公众号 : 甜菜欣欣