String Operations

charAt(), length(), indexOf(), lastIndexOf(), substring(), replace()

Strings

Strings are a sequence of characters. Many of the string operations are based on finding characters, or substrings of strings based on position.

0	1	2	3	4	5	6
C	a	r	m	i	n	e

charAt

You can get the character at a position in a string. charAt() is a function that strings have. Notice the index into the string starts with 0!

```
String name = "Carmine";
char initial = name.charAt(0);

System.out.println("First Initial = " + initial);
```

0	1	2	3	4	5	6
C	a	r	m	i	n	e

length

The **length**() function returns how many characters are in a string.

```
String s = "Cat";
int len = s.length();
char lastChar = s.charAt(len - 1);

System.out.println("The length is " + len);
System.out.println("Last character is " + lastChar);
```

0	1	2
C	a	t

concat

The function concat() returns the result of adding a string to a string. You could also do this using the + sign.

```
String name = "CS";
name = name.concat("121");
// Same as name = name + "121";
```

0	1	2	3	4
C	S	1	2	1

indexOf

You can find the first occurrence of a string in a string using indexOf().
If nothing is found, indexOf will return -1.

```
String name = "CS 121";
int position = name.indexOf(" ");
```

0	1	2	3	4	5
C	S		1	2	1

indexOf

You can start searching from a certain position using an optional “fromIndex” parameter.

```
String phrase = "I am Groot";
int index1 = phrase.indexOf(" ");
int index2 = phrase.indexOf(" ", index1 + 1);
```

0	1	2	3	4	5	6	7	8	9
I		a	m		G	r	o	o	t

lastIndexOf

You can start searching from the end of a string using lastIndexOf().

```
String list = "A,B,C,D";
int index = list.lastIndexOf(",");
```

0	1	2	3	4	5	6
A	,	B	,	C	,	D

substring

You can get a part of a string by using **substring()**.

```
String name = "CS 121";
int index = name.indexOf(" ");

String courseNumber = name.substring(index + 1);
```

0	1	2	3	4	5
C	S		1	2	1

substring

You can provide a start position and an end position.

```
String url = "http://www.pace.edu";

int p1 = url.indexOf("www.");
int p2 = url.indexOf(".edu");

System.out.println(url.substring(p1 + 4, p2));
```

0	1	2	3	4	5	6	7	8	9	0	11	12	13	14	15	16	17	18
h	t	t	p	:	/	/	w	w	w	.	p	a	c	e	.	e	d	u

replace

You can get the result of replacing all occurrences of a string with another string by using replace().

```
String phrase = "I am Carmine!";
System.out.println(phrase.replace("Carmine", "Groot"));

String list = "A_B_C_D";
list = list.replace("_", ",");
```

String function family (so far)

These are the string functions we have encountered so far.

`charAt`	- Returns the character at an index.
`length`	- Returns the number of characters.
`concat`	- Returns a string added to the string.
`indexOf`	- Returns the index of a string.
`lastIndexOf`	- Same as indexOf but starts at the end.
`substring`	- Returns a string inside another string.
`replace`	- Returns the replacement of all occurrences of a string.

Be Careful!

These string operations **return** values, but **do not alter** the original variable!

```
String phrase = "I am Carmine!";

phrase.replace("Carmine", "Groot");  // Does nothing!
System.out.println(phrase);          // I am Carmine!

phrase = phrase.replace("Carmine", "Groot");
System.out.println(phrase);          // I am Groot!
```

Output Formatting

123.45

printf

System.out.printf is a method for printing formatted output.

```
String name = scan.next();

System.out.printf("Hello %s, welcome to CS121.", name);


double value = 10.0 / 3.0;

System.out.printf("The value is: %f", value);
```

printf

You can have multiple kinds of format specifiers in a format string

```
String name = "Carmine";
double gpa = 4.0;

System.out.printf("Hello %s, your GPA is %f.", name, gpa);
```

printf

You can specify the width when printing.

```
int x = 5;
int y = 100;
int z = 1234;

// Notice: %4d
System.out.printf("%4d\n%4d\n%4d\n", x, y, z);

   5
 100
1234
```

printf

Some of the format specifiers you may use with System.out.printf

`%s`	`String`
`%c`	`Single Character`
`%d`	`Number (int, long, short)`
`%f`	`Number with decimal point (double, float)`
`%%`	`Prints the % symbol.`

printf

You can specify the number of decimal places.

```
double x = 123.45678;
double y = 2.555;
double z = 10.0 / 3.0;

// Notice: %.2f
System.out.printf("%.2f\n%.2f\n%.2f\n", x, y, z);

123.46
2.56
3.33
```

Let's Code

Don't Forget!

Check the syllabus / schedule for reading assignments and due dates!

Documentation

Where to learn more about Java Libraries (and more).

Java Documentation

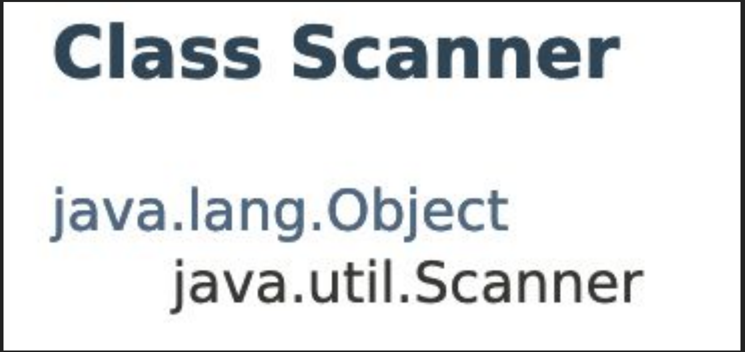
You can find the Java specification at the following url:
https://docs.oracle.com/javase/8/docs/api/

You can also google for:
java 8 docs

To jump to a specific class, you could search for:
java 8 docs Scanner

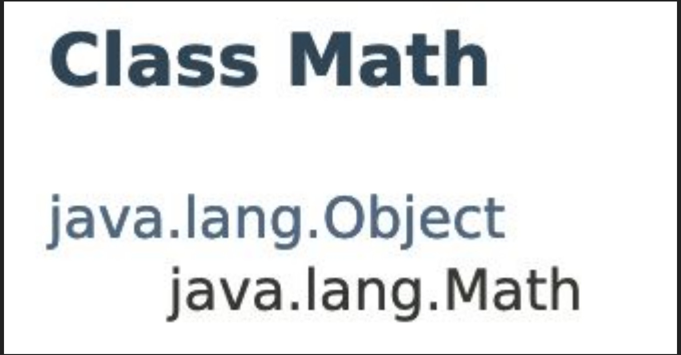
Java Documentation

Great for looking up what to **import** to use a class:



Class Scanner

java.lang.Object
java.util.Scanner

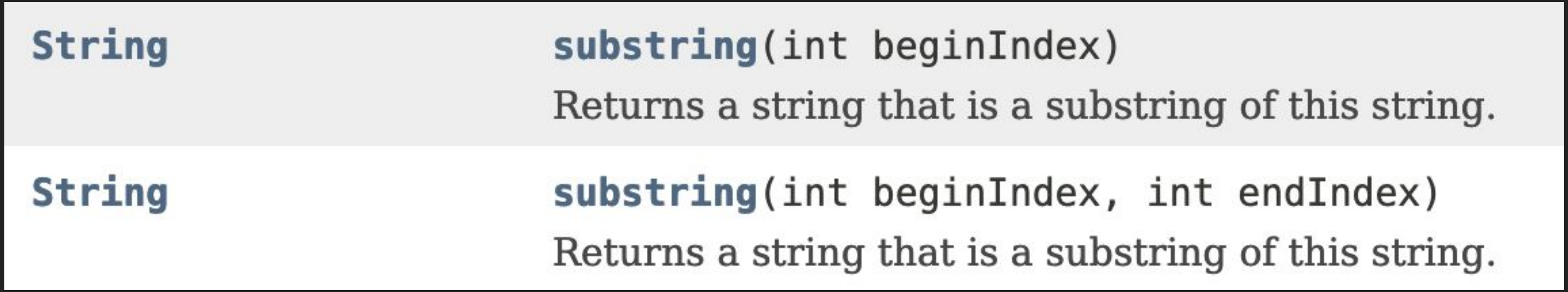


Class Math

java.lang.Object
java.lang.Math

Java Documentation

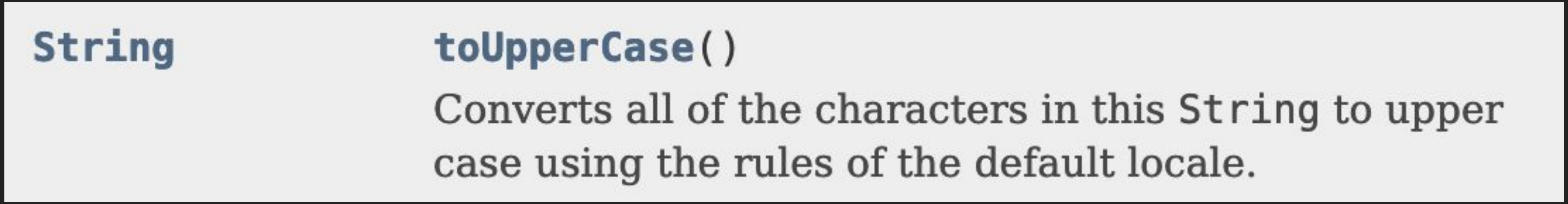
Find variations (and explanations) for functions.



`String`	`substring(int beginIndex)` Returns a string that is a substring of this string.
`String`	`substring(int beginIndex, int endIndex)` Returns a string that is a substring of this string.

Java Documentation

Discover functions you may need for other projects.

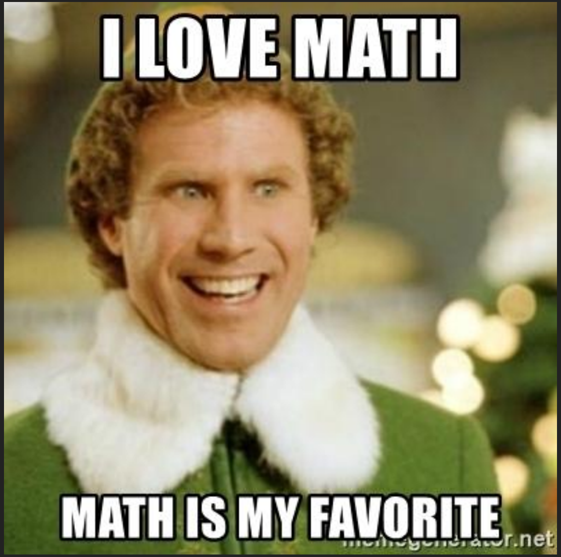


`String`	`toUpperCase()` Converts all of the characters in this `String` to upper case using the rules of the default locale.

Java Documentation Demo

The Math Class

For when you need to get past the basic operations: + - / * %



Math.sqrt and Math.pow

Math.sqrt is for calculating the square root. Math.pow is for calculating a number raised to the power of another number.

```
double x = Math.sqrt(9.0);        // Returns 3.0

double y = Math.pow(3.0, 2.0);    // Returns 9.0
```

Math.ceil and Math.floor

Math.ceil brings the number up to the next whole number. Math.floor brings the number down to the previous whole number.

```
double x = Math.ceil(2.3);     // Returns 3.0
double y = Math.ceil(-2.3);    // Returns -2.0

double x = Math.floor(2.3);    // Returns 2.0
double y = Math.floor(-2.3);   // Returns -3.0
```

-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0

Math.min , Math.max and Math.abs

Math.min returns the smaller of 2 numbers. Math.max returns the higher of two numbers. Math.abs will return the absolute value.

```
double x = Math.min(2.0, 5.0);// Returns 2.0

double y = Math.max(2.0, 5.0);// Returns 5.0

double y = Math.abs(-2.0);         // Returns 2.0
double y = Math.abs(2.0);          // Returns 2.0
```

int, long, double, float

Many of the Math functions have versions for handling int, long double and float. The data type you put in, is the one you get out.

```
double x = Math.max(2.0, 5.0);// Returns 5.0

int x = Math.max(2, 5);           // Returns 5
```

Computer Science!

Using just the Math.max method:
How can you write **one line of code** to get `maxValue`?

```
double x = 30;
double y = 2;
double z = 15;

double maxValue = ??
```

Computer Science!

Using just the Math.max method:
How can you write one line of code to get `maxValue`?

```
double x = 30;
double y = 2;
double z = 15;

double maxValue = Math.max(Math.max(x, y), z);
```

Computer Science!

Using just the Math.**max** method:
How can you write **one line of code** to get `maxValue`?

```
Math.max(Math.max(x, y), z);
           ↓
Math.max(Math.max(20, 2), z);
           ↙
Math.max(20, z);
           ↓
Math.max(20, 15);   →   20
```

Random

Random

Random is a class used to generate random numbers. Like Scanner, Random needs to be imported and initialized.

```
import java.util.Random;

public class RandomExample {
    public static void main(String[] args) {

        Random rand = new Random();    // Initialize

        int x = rand.nextInt();        // Get a random integer
        int y = rand.nextInt(10);      // Random from 0 to 9
    }
}
```

Random

nextInt(6) gives us a number from 0 to 5.
If we wanted a number from 1 to 6, what would we do?

What if we wanted a number between 10 and 20?

Random

nextInt(6) gives us a number from 0 to 5.
If we wanted a number from 1 to 6, what would we do? nextInt(6) + 1

What if we wanted a number between 10 and 20? nextInt(11) + 10

Random

nextInt(6) gives us a number from 0 to 5.
If we wanted a number from 1 to 6, what would we do? nextInt(6) + 1

Adding 1 will shift the values from 0 to 5 -> 1 to 6

What if we wanted a number between 10 and 20? nextInt(11) + 10

10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20 (we want 11 values)

nextInt(the number of values we want) + the starting number

Let's Code

Don't Forget!

Check the syllabus / schedule for reading assignments and due dates!