Working with Java Predefined Classes

1. Java Predefined Classes Documentation
2. Creating objects
3. Calling methods
4. Using fields

Java Predefined Classes Documentation

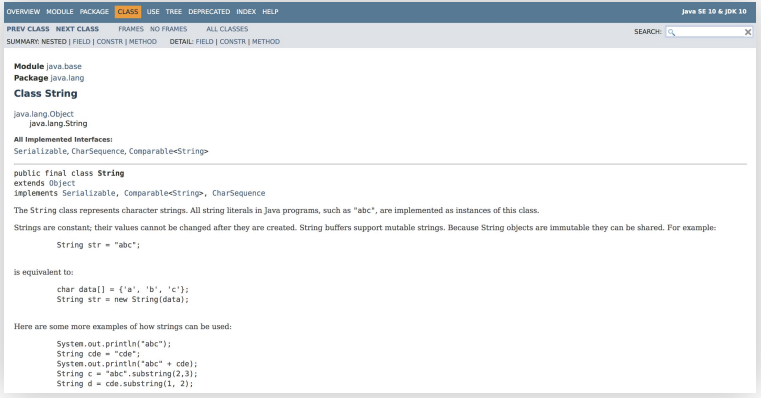
1. Java Predefined Classes Documentation

docs.oracle.com

- Class description
- Field Summary
- Constructor Summary
- Method Summary
  - Instance methods (with and without parameters)
  - Static methods (with and without parameters)

1. Java Predefined Classes Documentation

docs.oracle.com



Creating objects

docs.oracle.com

- Class description
- Field Summary
- Constructor Summary
- Method Summary
  - Instance methods (with and without parameters)
  - Static methods (with and without parameters)

`new` operator

```
String message = new String();
```

Constructor method

2. Creating objects

Constructor Summary

Constructors

Constructor	Description
`String()`	Initializes a newly created `String` object so that it represents an empty character sequence.
`String(byte[] bytes)`	Constructs a new `String` by decoding the specified array of bytes using the platform's default charset.
`String(byte[] ascii, int hibyte)`	**Deprecated.** This method does not properly convert bytes into characters.
`String(byte[] bytes, int offset, int length)`	Constructs a new `String` by decoding the specified subarray of bytes using the platform's default charset.
`String(byte[] ascii, int hibyte, int offset, int count)`	**Deprecated.** This method does not properly convert bytes into characters.
`String(byte[] bytes, int offset, int length, String charsetName)`	Constructs a new `String` by decoding the specified subarray of bytes using the specified charset.
`String(byte[] bytes, int offset, int length, Charset charset)`	Constructs a new `String` by decoding the specified subarray of bytes using the specified charset.
`String(byte[] bytes, String charsetName)`	Constructs a new `String` by decoding the specified array of bytes using the specified charset.
`String(byte[] bytes, Charset charset)`	Constructs a new `String` by decoding the specified array of bytes using the specified charset.
`String(char[] value)`	Allocates a new `String` so that it represents the sequence of characters currently contained in the character array argument.
`String(char[] value, int offset, int count)`	Allocates a new `String` that contains characters from a subarray of the character array argument.
`String(int[] codePoints, int offset, int count)`	Allocates a new `String` that contains characters from a subarray of the Unicode code point array argument.
`String(String original)`	Initializes a newly created `String` object so that it represents the same sequence of characters as the argument; in other words, the newly created string is a copy of the argument string.

String class Java Documentation

new operator

```
char[] letters= {'h','e','l','l','o'};
String message = new String(letters);
```

Constructor method

new operator

```
char[] letters= {'h','e','l','l','o'};
String message = new String(letters);
```

Method parameters
(parameter1, parameter2, ….)

docs.oracle.com

- Class description
- Field Summary
- Constructor Summary
- Method Summary
  - Instance methods (with and without parameters)
  - Static methods (with and without parameters)

2. Creating objects

Method Summary

All Methods	Static Methods	Instance Methods	Concrete Methods	Deprecated Methods

Modifier and Type	Method	Description
static String	copyValueOf(char[] data)	Equivalent to valueOf(char[]).
static String	copyValueOf(char[] data, int offset, int count)	Equivalent to valueOf(char[], int, int).
static String	format(String format, Object... args)	Returns a formatted string using the specified format string and arguments.
static String	format(Locale l, String format, Object... args)	Returns a formatted string using the specified locale, format string, and arguments.
static String	join(CharSequence delimiter, CharSequence... elements)	Returns a new String composed of copies of the CharSequence elements joined together with a copy of the specified delimiter.
static String	join(CharSequence delimiter, Iterable<? extends CharSequence> elements)	Returns a new String composed of copies of the CharSequence elements joined together with a copy of the specified delimiter.
static String	valueOf(boolean b)	Returns the string representation of the boolean argument.
static String	valueOf(char c)	Returns the string representation of the char argument.
static String	valueOf(char[] data)	Returns the string representation of the char array argument.
static String	valueOf(char[] data, int offset, int count)	Returns the string representation of a specific subarray of the char array argument.
static String	valueOf(double d)	Returns the string representation of the double argument.
static String	valueOf(float f)	Returns the string representation of the float argument.
static String	valueOf(int i)	Returns the string representation of the int argument.
static String	valueOf(long l)	Returns the string representation of the long argument.
static String	valueOf(Object obj)	Returns the string representation of the Object argument.

String class Java Documentation

static method

```
String text = String.valueOf(3.14f);
```

static method

```
String text = String.valueOf(3.14f);
```

String literal

```
String message= “hello”;
```

Calling methods

docs.oracle.com

- Class description
- Field Summary
- Constructor Summary
- Method Summary
  - Instance methods (with and without parameters)
  - Static methods (with and without parameters)

Instance method

```
[ClassDataType] objectVariable = …;

objectVariable.methodName(parameter1, parameter2, …, parameterN);
```

Syntax

```
String text= “Hello world”;

int numCharacters= text.length(); //11
char character= text.chartAt(0); //H
```

Example

Static method

```
ClassName.methodName(parameter1, parameter2, …, parameterN);
```

Syntax

```
String texto = String.valueOf(3.1416f);
```

Example

Using fields

docs.oracle.com

- Class description
- Field Summary
- Constructor Summary
- Method Summary
  - Instance methods (with and without parameters)
  - Static methods (with and without parameters)

4. Using fields

Field Summary

Fields

Modifier and Type	Field	Description
`static double`	`E`	The `double` value that is closer than any other to *e*, the base of the natural logarithms.
`static double`	`PI`	The `double` value that is closer than any other to *pi*, the ratio of the circumference of a circle to its diameter.

Math class Java Documentation

Static field

```
ClassName.fieldName;
```

Syntax

```
double radio = 4.5;
double areaCirculo= Math.PI*Math.pow(radio,2);
```

Example

“El riesgo más grande es no tomar ninguno. En un mundo que está cambiando tan rápido, la única estrategia que está garantizada a fracasar es no tomar riesgos.“

Mark Zuckerberg, co-fundador de Facebook y jefe ejecutivo