Strings manipulation

1. Methods
2. Instance methods
3. Static methods
4. Casting

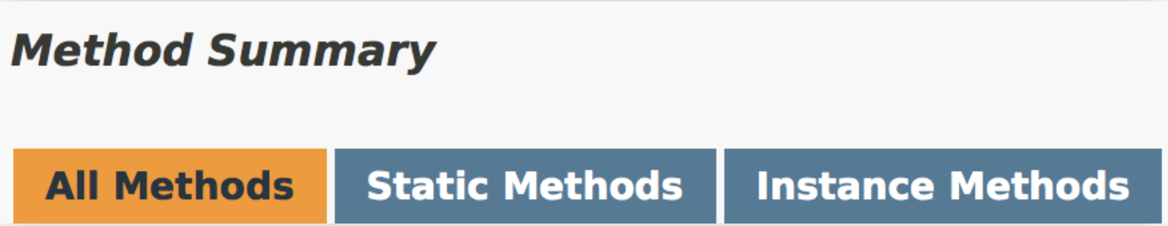
Methods

1. Methods

Instance methods

Static methods

1. Methods



Method Summary

String class Java Documentation

Instance methods

2. Instance methods

```
String variable = ...;

variable.methodName(arguments);
```

2. Instance methods

```
String variable = ...;

variable.methodName(arguments);
```

2. Instance methods

```
String variable = …;

variable.methodName(arguments);
```

2. Instance methods

```
String variable = …;

variable.methodName(arguments);
```

JAVA DOCUMENTATION

2. Instance methods

```
String variable = …;

variable.methodName(arguments);
```

Method parameters

2. Instance methods

```
String variable = …;

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

Method parameters

2. Instance methods

```
String variable = …;

variable.methodName();
```

NO Method parameters

2. Instance methods

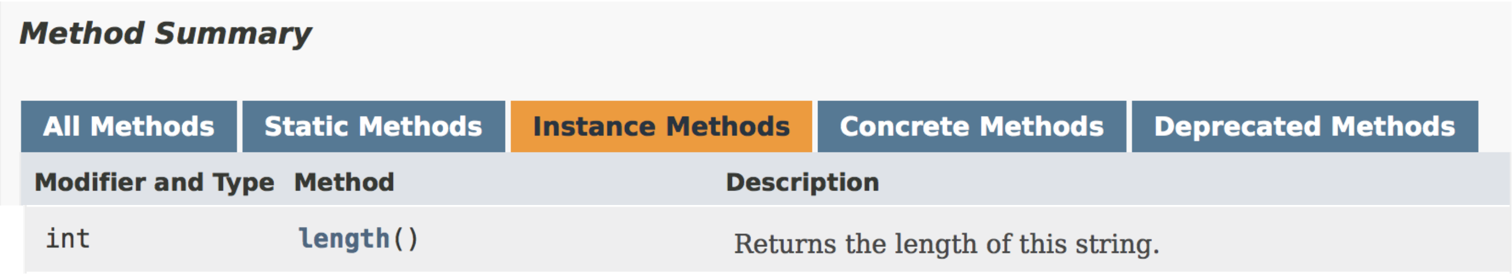
length method

En un lugar de la Mancha

24 characters

2. Instance methods

length method



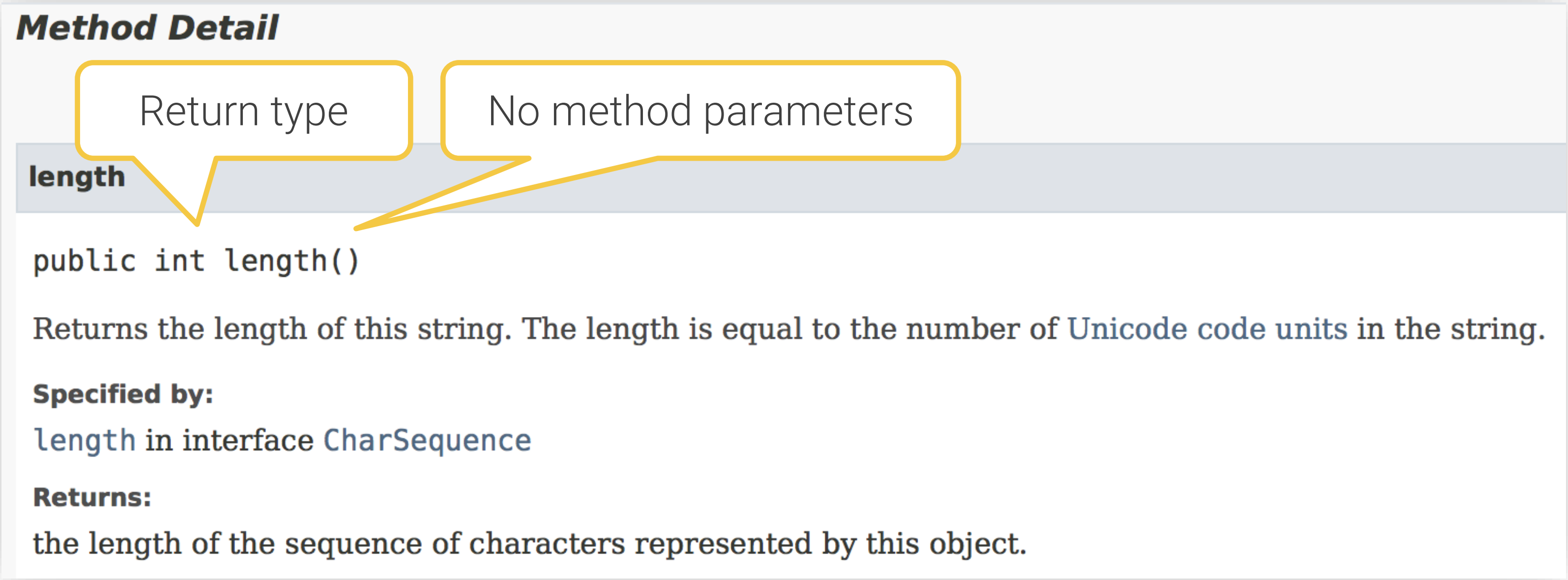
Method Summary

Modifier and Type	Method	Description
`int`	`length()`	Returns the length of this string.

String class Java Documentation

2. Instance methods

length method



Method Detail

length

```
public int length()
```

Returns the length of this string. The length is equal to the number of Unicode code units in the string.

Specified by:

`length` in interface `CharSequence`

Returns:

the length of the sequence of characters represented by this object.

String class Java Documentation

2. Instance methods

length method

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

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("Introduce texto: ");
        String frase = input.nextLine();
        int num = frase.length();
        System.out.println("El texto tiene "+num+" caracteres");
    }
}
```

2. Instance methods

length method

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

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("Introduce texto: ");
        String frase = input.nextLine();
        int num = frase.length();
        System.out.println("El texto tiene "+num+" caracteres");
    }
}
```

length method

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

   public static void main(String[] args) {
      Scanner input = new Scanner(System.in);
      System.out.println("Introduce texto: ");
      String frase = input.nextLine();
      int num = frase.length();
      System.out.println("El texto tiene "+num+" caracteres");
   }
}
```

2. Instance methods

length method

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

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("Introduce texto: ");
        String frase = input.nextLine();
        System.out.println("El texto tiene "+frase.length()+" caracteres");
    }
}
```

2. Instance methods

¿Cual será el valor de la variable index?

```
String message = “Hello”;
int index = message.indexOf(‘a’);
```

?

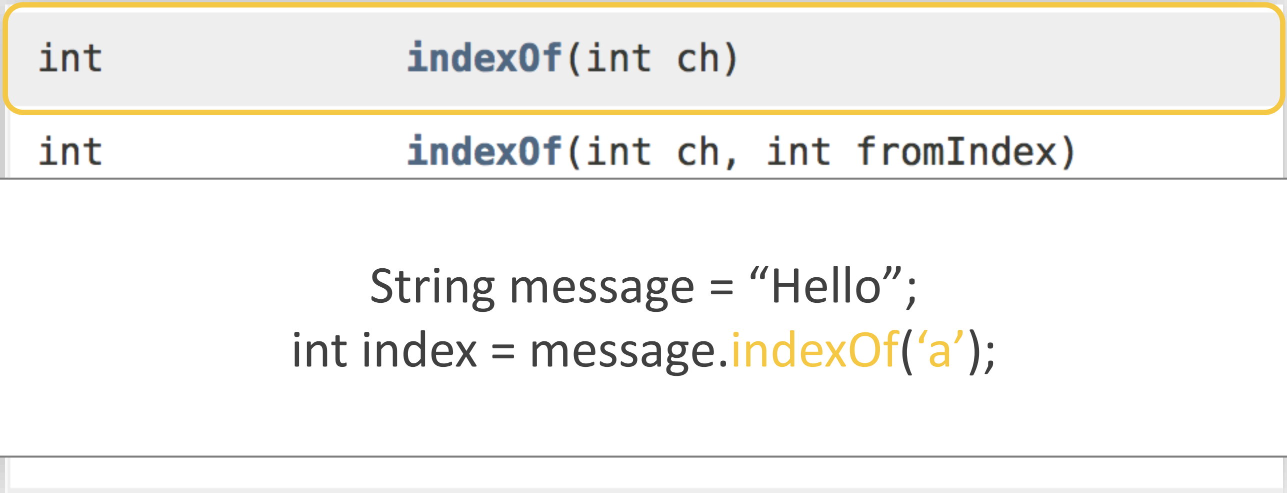
2. Instance methods

¿Cual será el valor de la variable index?

```
String message = “Hello”;
int index = message.indexOf(‘a’);
```

2. Instance methods

¿Cual será el valor de la variable index?



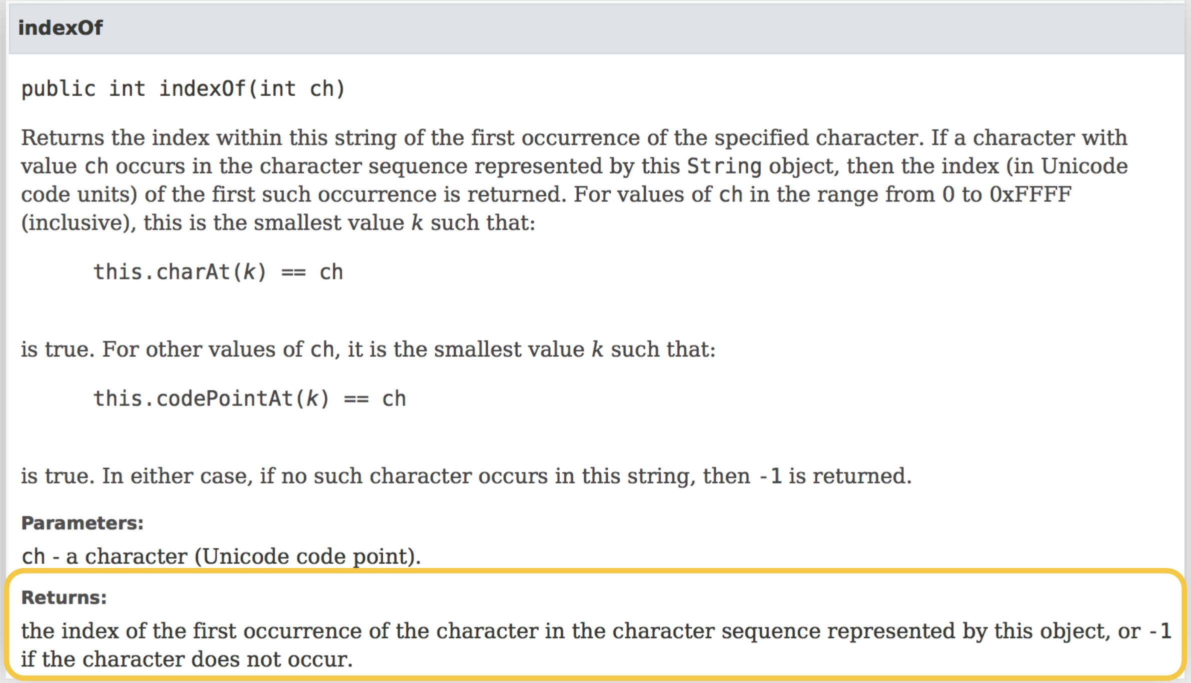
```
int          indexOf(int ch)
int          indexOf(int ch, int fromIndex)
```

```
String message = “Hello”;
int index = message.indexOf(‘a’);
```

String class Java Documentation

2. Instance methods

¿Cual será el valor de la variable index?



indexOf

```
public int indexOf(int ch)
```

Returns the index within this string of the first occurrence of the specified character. If a character with value ch occurs in the character sequence represented by this String object, then the index (in Unicode code units) of the first such occurrence is returned. For values of ch in the range from 0 to 0xFFFF (inclusive), this is the smallest value *k* such that:

```
    this.charAt(k) == ch
```

is true. For other values of ch, it is the smallest value *k* such that:

```
    this.codePointAt(k) == ch
```

is true. In either case, if no such character occurs in this string, then -1 is returned.

Parameters:

ch - a character (Unicode code point).

Returns:

the index of the first occurrence of the character in the character sequence represented by this object, or -1 if the character does not occur.

2. Instance methods

¿Cual será el valor de la variable index?

```
String message = “Hello”;
int index = message.indexOf(‘a’);
```

-1

2. Instance methods

¿Cual será el valor de la variable index?

```
int index = “abba”.lastIndexOf(‘a’);
```

?

2. Instance methods

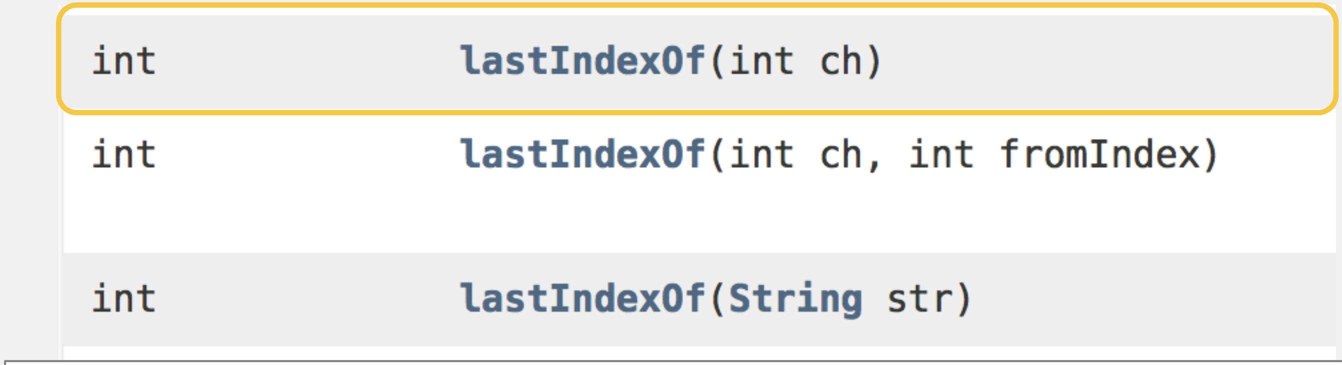
¿Cual será el valor de la variable index?

```
int index = “abba”.lastIndexOf(‘a’);
```

?

2. Instance methods

¿Cual será el valor de la variable index?



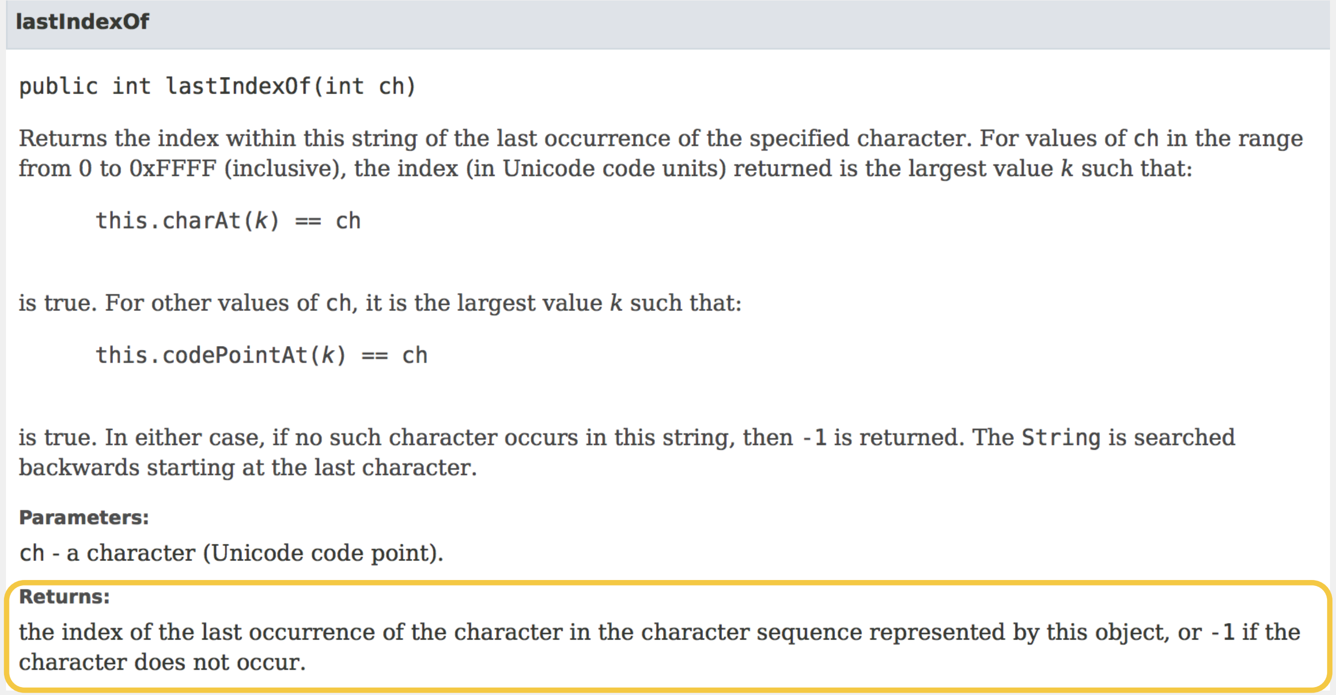
int	lastIndexOf(int ch)
int	lastIndexOf(int ch, int fromIndex)
int	lastIndexOf(String str)

```
int index = “abba”.lastIndexOf(‘a’);
```

[String class Java Documentation]()

2. Instance methods

¿Cual será el valor de la variable index?



lastIndexOf

```
public int lastIndexOf(int ch)
```

Returns the index within this string of the last occurrence of the specified character. For values of `ch` in the range from 0 to 0xFFFF (inclusive), the index (in Unicode code units) returned is the largest value *k* such that:

```
    this.charAt(k) == ch
```

is true. For other values of `ch`, it is the largest value *k* such that:

```
    this.codePointAt(k) == ch
```

is true. In either case, if no such character occurs in this string, then `-1` is returned. The `String` is searched backwards starting at the last character.

Parameters:

`ch` - a character (Unicode code point).

Returns:

the index of the last occurrence of the character in the character sequence represented by this object, or `-1` if the character does not occur.

2. Instance methods

¿Cual será el valor de la variable index?

```
int index = “abba”.lastIndexOf(‘a’);
             0123
```

3

Static methods

3. Static methods

```
ClassName.methodName(arguments);
```

3. Static methods

```
ClassName.methodName(arguments);
```

3. Static methods

```
ClassName.methodName(arguments) ;
```

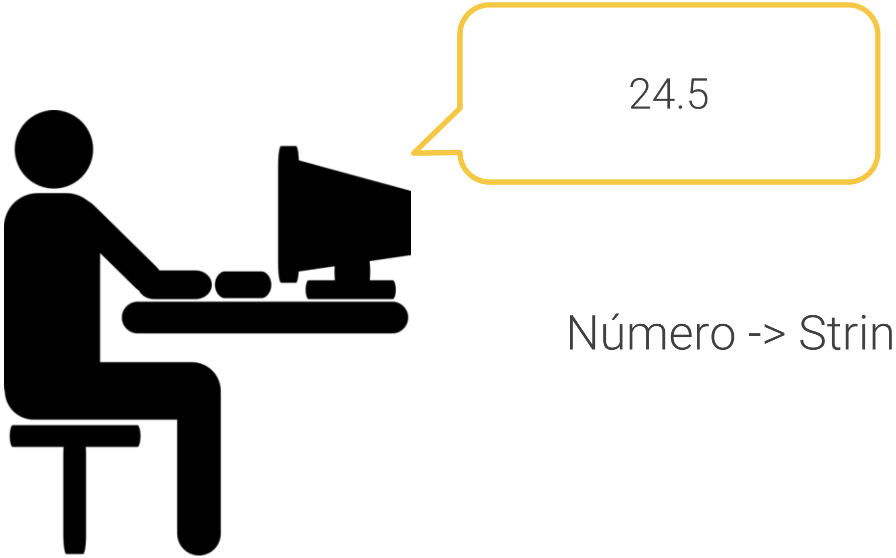
3. Static methods

~~String variable = ...;~~

```
ClassName.methodName(arguments);
```

3. Static methods

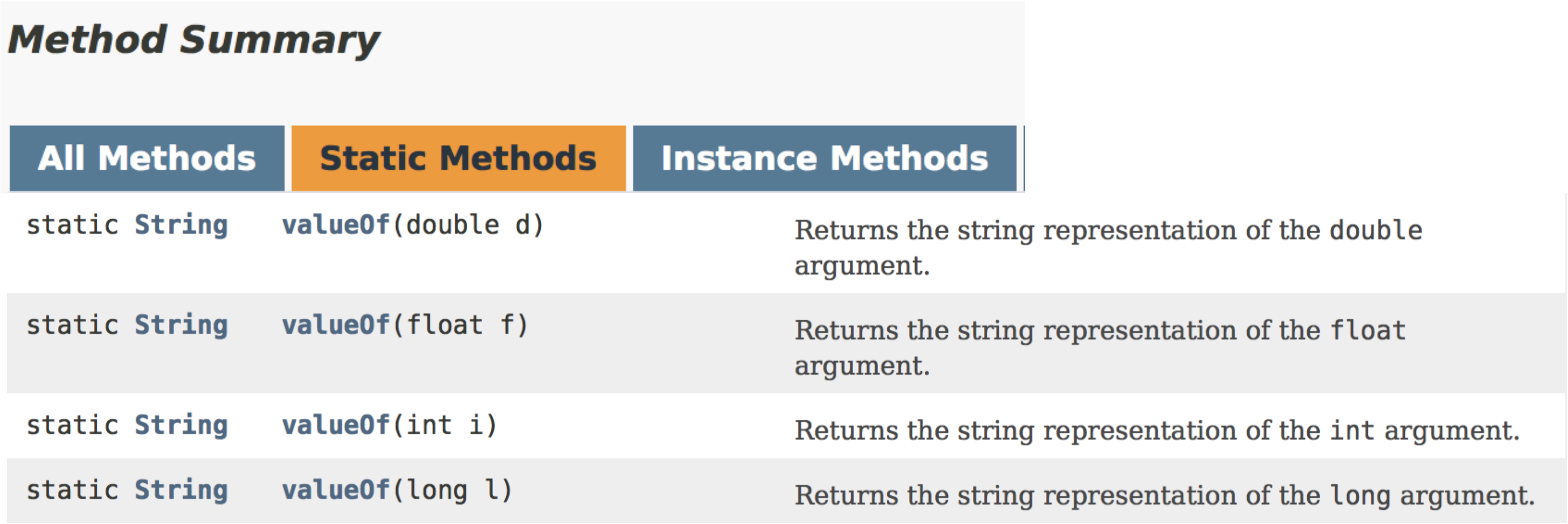
valueOf method



Número -> String

3. Static methods

valueOf method



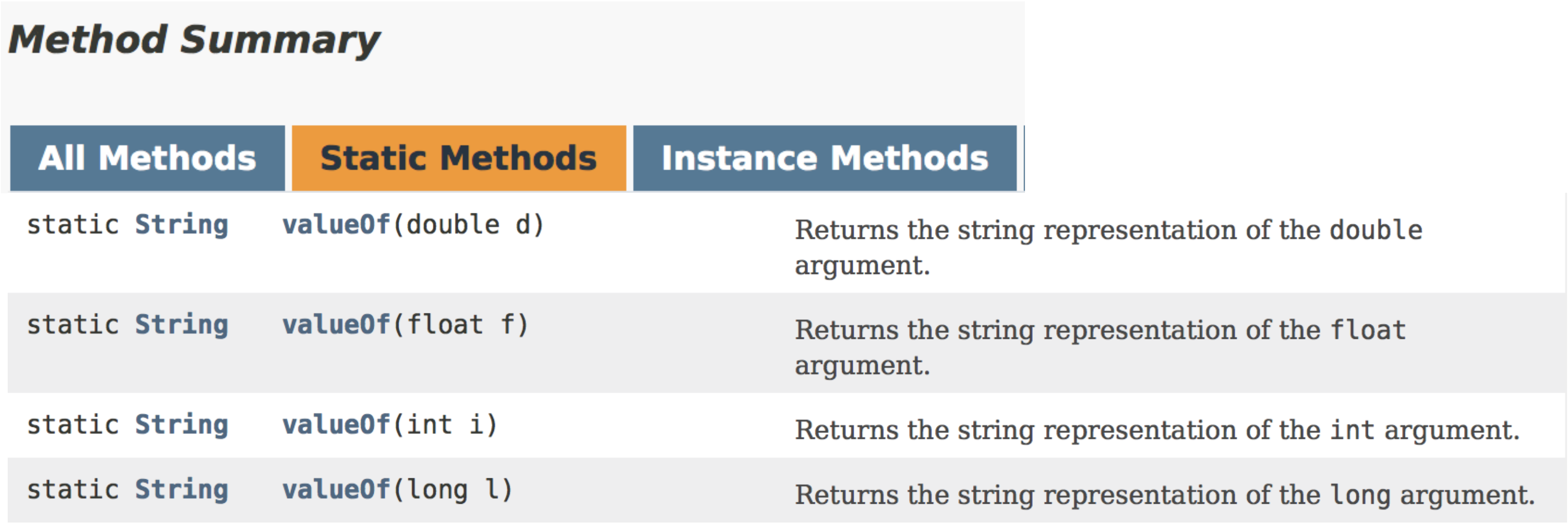
Method Summary

All Methods | Static Methods | Instance Methods

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.

String class Java Documentation

valueOf method



Method Summary

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 keyword

String class Java Documentation

3. Static methods

valueOf method

```
public class Main {

   public static void main(String[] args) {
         float numero = 3.1415926535f;
         //Lo convertimos en una cadena de texto.
         String numeroText = String.valueOf(numero);
   }
}
```

Casting

4. Casting

String -> número

4. Casting

Instrucción	Tipo de dato convertido
Byte.parseByte(text);	byte
Short.parseShort(text);	short
Integer.parseInt(text);	int
Long.parseLong(text);	long
Float.parseFloat(text);	float
Double.parseDouble(text);	double

4. Casting

Instrucción	Tipo de dato convertido
Byte.parseByte(text);	byte
Short.parseShort(text);	short
Integer.parseInt(text);	int
Long.parseLong(text);	long
Float.parseFloat(text);	float
Double.parseDouble(text);	double

4. Casting

```
public class Main {

    public static void main(String[] args) {
        String arg0="30.5";
        String arg1="24.5";
        double dividendo = Double.parseDouble(arg0);
        double divisor = Double.parseDouble(arg1);
        double resultado = dividendo/divisor;
        System.out.println("El resultado es " + resultado);
    }
}
```

“Cree que puedes y estarás a mitad del camino.“

Theodore Roosevelt, presidente EEUU (1901-1909)