

- Important String Methods
 - Searching Strings [`indexOf()` and `lastIndexOf()`]
 - Modifying a String [`substring()`, `replace()` and `trim()`]
 - Data Conversion [`valueOf()`]
 - Changing Case of Characters [`toUpperCase()` and `toLowerCase()`]

Searching Strings : indexOf() and lastIndexOf()



- **int indexOf(int ch)** → Searches the index of character 'ch' in **this** string and returns that index value (if found) otherwise returns -1 (if not found)
- **int lastIndexOf(int ch)** → Searches the last index of character 'ch' in **this** string and returns that index value (if found) otherwise returns -1 (if not found)
- **int indexOf(String str)** → Searches the index of string "str" in **this** string and returns that index value (if found) otherwise returns -1 (if not found)
- **int lastIndexOf(String str)** → Searches the last index of string "str" in **this** string and returns that index value (if found) otherwise returns -1 (if not found)
- **int indexOf(int ch, int startIndex)** → Searches the index of character 'ch' in **this** string starting from **startIndex** and returns that index value (if found) otherwise returns -1 (if not found)
- **int lastIndexOf(int ch, int startIndex)** → Searches the last index of character 'ch' in **this** string starting from **startIndex** and returns that index value (if found) otherwise returns -1 (if not found)
- **int indexOf(String str, int startIndex)** → Searches the index of string "str" in **this** string starting from **startIndex** and returns that index value (if found) otherwise returns -1 (if not found)
- **int lastIndexOf(String str, int startIndex)** → Searches the last index of string "str" in **this** string starting from **startIndex** and returns that index value (if found) otherwise returns -1 (if not found)

Searching Strings : indexOf() and lastIndexOf() : Example



```
String s = "Now is the Time For All Indian";  
s = s + "to come to the aid of India";
```

```
System.out.println(s.indexOf("x"));
```



-1

```
System.out.println(s.indexOf("a"));
```



28

```
System.out.println(s.lastIndexOf("a"))
```



56

```
System.out.println(s.indexOf("Ind"));
```



24

```
System.out.println(s.lastIndexOf("Ind"));
```



52

```
System.out.println(s.indexOf("Ind",30));
```



52

Modifying Strings : substring()



- **String substring(int startIndex) →** Returns a substring starting from **startIndex** (inclusive) to the **last** of **this** string [Note: $0 \leq \text{startIndex} \leq L$, where **L** is length of **this** String]
- **String substring(int startIndex, int endIndex) →** Returns a substring starting from **startIndex** (inclusive) to the **endIndex** (exclusive) of **this** string. Returned substring will have characters from **startIndex** to **endIndex-1** [Note: $0 \leq \text{startIndex}, \text{endIndex} \leq L$, where **L** is length of **this** String and **startIndex** \leq **endIndex**]
 - If values of **startIndex** and **endIndex** are equal, then no character will be extracted
 - If value of **startIndex** $<$ **endIndex** OR **startIndex**, **endIndex** $>$ **L** o, then **StringIndexOutOfBoundsException** will be thrown

Modifying Strings : substring() Example



String s = "Now is the Time For All Indian";

System.out.println(s.length());

→ 30

System.out.println(s.substring(30,30));

→ <<No Character
Will be
extracted

System.out.println(s.substring(20,20));

→ <<No Character
Will be
extracted

System.out.println(s.substring(0,2));

→ No

System.out.println(s.substring(11));

→ Time For All Indian

Replacing Characters in String



- **String replace(char original, char replacement)** → Replaces all occurrences of a character **original** in **this** String with a character **replacement** and returns the modified String
- Example : `System.out.println("Object-Oriented Programming".replace('m','x'))` will result in "Object-Oriented Prograxxing" as output
- **String trim()** → Removes the leading and trailing white spaces from the invoking string and returns the modified String
- Example

`System.out.println(" My Name is David ".length());` → 20
`System.out.println(" My Name is David ".trim());` → My Name is David
`System.out.println(" My Name is David ".trim().length());` → 16

Changing the Case of Characters



- `String toUpperCase()` → Changes each character of the invoking string to upper case and returns the modified string
- `String toLowerCase()` → Changes each character of the invoking string to lower case and returns the modified string
- For Example, the statement `System.out.println("My Name is David".toUpperCase());` displays "MY NAME IS DAVID" over console
- For Example, the statement `System.out.println("My Name is David".toLowerCase());` displays "my name is david" over console

Thank You