**String**

জাভাতে String ক্লাসটি অনেকগুলি বিল্ট-ইন মেথড প্রদান করে যা স্ট্রিং ম্যানিপুলেশন, সার্চিং, কম্পেয়ারিং এবং অন্যান্য অপারেশনের জন্য ব্যবহৃত হয়। নিচে জাভার String ক্লাসের সমস্ত গুরুত্বপূর্ণ মেথডগুলির একটি তালিকা দেওয়া হলো:

**স্ট্রিং ম্যানিপুলেশন মেথড:**

1. **charAt(int index)**:
   * নির্দিষ্ট ইনডেক্সে অবস্থিত ক্যারেক্টার রিটার্ন করে।

java

Copy

String str = "Hello";

char ch = str.charAt(1); // ch = 'e'

1. **concat(String str)**:
   * দুটি স্ট্রিং কে যুক্ত করে একটি নতুন স্ট্রিং রিটার্ন করে।

java

Copy

String str1 = "Hello";

String str2 = "World";

String result = str1.concat(str2); // result = "HelloWorld"

1. **substring(int beginIndex)**:
   * beginIndex থেকে শুরু করে স্ট্রিং এর শেষ পর্যন্ত অংশ রিটার্ন করে।

java

Copy

String str = "HelloWorld";

String sub = str.substring(5); // sub = "World"

1. **substring(int beginIndex, int endIndex)**:
   * beginIndex থেকে endIndex - 1 পর্যন্ত অংশ রিটার্ন করে।

java

Copy

String str = "HelloWorld";

String sub = str.substring(0, 5); // sub = "Hello"

1. **toLowerCase()**:
   * স্ট্রিং এর সমস্ত ক্যারেক্টারকে লোয়ার কেসে রূপান্তর করে।

java

Copy

String str = "Hello";

String lower = str.toLowerCase(); // lower = "hello"

1. **toUpperCase()**:
   * স্ট্রিং এর সমস্ত ক্যারেক্টারকে আপার কেসে রূপান্তর করে।

java

Copy

String str = "Hello";

String upper = str.toUpperCase(); // upper = "HELLO"

1. **trim()**:
   * স্ট্রিং এর শুরু এবং শেষের স্পেস (হোয়াইটস্পেস) রিমুভ করে।

java

Copy

String str = " Hello ";

String trimmed = str.trim(); // trimmed = "Hello"

1. **replace(char oldChar, char newChar)**:
   * স্ট্রিং এর সমস্ত oldChar কে newChar দিয়ে রিপ্লেস করে।

java

Copy

String str = "Hello";

String replaced = str.replace('l', 'p'); // replaced = "Heppo"

1. **replace(CharSequence target, CharSequence replacement)**:
   * স্ট্রিং এর সমস্ত target সিকোয়েন্সকে replacement দিয়ে রিপ্লেস করে।

java

Copy

String str = "Hello World";

String replaced = str.replace("World", "Java"); // replaced = "Hello Java"

**স্ট্রিং কম্পেয়ারিং মেথড:**

1. **equals(Object obj)**:
   * দুটি স্ট্রিং এর কন্টেন্ট সমান কিনা তা চেক করে (কেস সেনসিটিভ)।

java

Copy

String str1 = "Hello";

String str2 = "hello";

boolean isEqual = str1.equals(str2); // isEqual = false

1. **equalsIgnoreCase(String str)**:
   * দুটি স্ট্রিং এর কন্টেন্ট সমান কিনা তা চেক করে (কেস সেনসিটিভ নয়)।

java

Copy

String str1 = "Hello";

String str2 = "hello";

boolean isEqual = str1.equalsIgnoreCase(str2); // isEqual = true

1. **compareTo(String str)**:
   * দুটি স্ট্রিং লেক্সিকোগ্রাফিকালি কম্পেয়ার করে (কেস সেনসিটিভ)।

java

Copy

String str1 = "apple";

String str2 = "banana";

int result = str1.compareTo(str2); // result = -1 (str1 < str2)

1. **compareToIgnoreCase(String str)**:
   * দুটি স্ট্রিং লেক্সিকোগ্রাফিকালি কম্পেয়ার করে (কেস সেনসিটিভ নয়)।

java

Copy

String str1 = "Apple";

String str2 = "apple";

int result = str1.compareToIgnoreCase(str2); // result = 0 (str1 == str2)

**স্ট্রিং সার্চিং মেথড:**

1. **indexOf(int ch)**:
   * স্ট্রিং এর মধ্যে প্রথম ch ক্যারেক্টারের ইনডেক্স রিটার্ন করে।

java

Copy

String str = "Hello";

int index = str.indexOf('e'); // index = 1

1. **indexOf(String str)**:
   * স্ট্রিং এর মধ্যে প্রথম str সাবস্ট্রিং এর ইনডেক্স রিটার্ন করে।

java

Copy

String str = "Hello World";

int index = str.indexOf("World"); // index = 6

1. **lastIndexOf(int ch)**:
   * স্ট্রিং এর মধ্যে শেষ ch ক্যারেক্টারের ইনডেক্স রিটার্ন করে।

java

Copy

String str = "Hello";

int index = str.lastIndexOf('l'); // index = 3

1. **contains(CharSequence s)**:
   * স্ট্রিং এর মধ্যে s সাবস্ট্রিং আছে কিনা তা চেক করে।

java

Copy

String str = "Hello World";

boolean contains = str.contains("World"); // contains = true

**স্ট্রিং কনভার্সন মেথড:**

1. **toCharArray()**:
   * স্ট্রিং কে ক্যারেক্টার অ্যারেতে রূপান্তর করে।

java

Copy

String str = "Hello";

char[] charArray = str.toCharArray(); // charArray = ['H', 'e', 'l', 'l', 'o']

1. **valueOf()**:
   * বিভিন্ন ডেটা টাইপ (যেমন int, double, char[]) কে স্ট্রিং এ রূপান্তর করে।

java

Copy

int num = 123;

String str = String.valueOf(num); // str = "123"

**অন্যান্য মেথড:**

1. **length()**:
   * স্ট্রিং এর দৈর্ঘ্য রিটার্ন করে।

java

Copy

String str = "Hello";

int len = str.length(); // len = 5

1. **isEmpty()**:
   * স্ট্রিং খালি কিনা তা চেক করে।

java

Copy

String str = "";

boolean isEmpty = str.isEmpty(); // isEmpty = true

1. **startsWith(String prefix)**:
   * স্ট্রিং prefix দিয়ে শুরু হয় কিনা তা চেক করে।

java

Copy

String str = "Hello World";

boolean startsWith = str.startsWith("Hello"); // startsWith = true

1. **endsWith(String suffix)**:
   * স্ট্রিং suffix দিয়ে শেষ হয় কিনা তা চেক করে।

java

Copy

String str = "Hello World";

boolean endsWith = str.endsWith("World"); // endsWith = true

1. **split(String regex)**:
   * স্ট্রিং কে একটি রেগুলার এক্সপ্রেশন অনুযায়ী ভাগ করে অ্যারে রিটার্ন করে।

java

Copy

String str = "Hello,World,Java";

String[] parts = str.split(","); // parts = ["Hello", "World", "Java"]

**সারসংক্ষেপ:**

জাভার String ক্লাসে অনেকগুলি মেথড রয়েছে যা স্ট্রিং ম্যানিপুলেশন, সার্চিং, কম্পেয়ারিং এবং অন্যান্য অপারেশনের জন্য ব্যবহৃত হয়। এই মেথডগুলি ব্যবহার করে আপনি সহজেই স্ট্রিং ডেটা প্রসেস করতে পারেন।