The String Constructors

```
String s = new String();   //1

char chars[] = { 'a', 'b', 'c' };
String s = new String(chars); //2

char chars[] = { 'a', 'b', 'c', 'd', 'e', 'f' };
String s = new String(chars, 2, 3);//3

String s = new String(String strObj);//4

String s = new String(byte asciiChars[ ]);//5
byte ascii[] = {65, 66, 67, 68, 69, 70 };
String s1 = new String(ascii);

String s = new String(byte asciiChars[ ], int startIndex, int numChars);//6
```

String Length

```
int length( )
```

Special String Operations
String Literals

```
String s2 = "abc"; // use string literal
System.out.println("abc".length());
```

String Concatenation (+ operator)

String Conversion and toString( )
String.valueOf()
String.toString()

Character Extraction

```
char charAt(int where)
void getChars(int sourceStart, int sourceEnd, char target[ ], int targetStart)
class getCharsDemo {
        public static void main(String args[]) {
                String s = "This is a demo of the getChars method.";
                int start = 10;
                int end = 14;
                char buf[] = new char[end - start];
                s.getChars(start, end, buf, 0);
```

```
                System.out.println(buf);
        }
}
```

byte[ ] getBytes( )
char[ ] toCharArray( )

String Comparsion
equals( ) and equalsIgnoreCase( )
boolean regionMatches(int startIndex, String str2,int str2StartIndex, int numChars)
boolean regionMatches(boolean ignoreCase,int startIndex, String str2,int str2StartIndex, int numChars)
boolean startsWith(String str)
boolean endsWith(String str)

the equals( ) method compares the characters inside a String object.
The == operator compares two object references to see whether they refer to the same instance

int compareTo( )
int compareToIgnoreCase(String str)

Value	Meaning
Less than zero	The invoking string is less than str.
Greater than zero	The invoking string is greater than str.
Zero	The two strings are equal

Searching Strings
int indexOf(int ch/ String str)
int lastIndexOf(int ch/String str)

int indexOf(int ch, int startIndex)
int lastIndexOf(int ch, int startIndex)
int indexOf(String str, int startIndex)
int lastIndexOf(String str, int startIndex)

Modifying a String
String substring(int startIndex)
String substring(int startIndex, int endIndex)
concat( )
replace( )
trim( )

Data Conversion Using valueOf( )
when a string representation of some other type of data is needed—for example, during concatenation operations

String.vlaueOf(double d)

Changing the Case of Characters Within a String
String toLowerCase( )
String toUpperCase( )

Other methods
contains()
String[ ] split(String regExp)
String[ ] split(String regExp, int max)
Operations:
Character Extraction
Searching
Modifying
Comparsion