

# **STRINGS**

**BY ANNEMARIE CABALLERO  
I.T. GIRLS AUGUST 13-17**

# WHAT IS A STRING?

Strings are objects, whose class (blueprint) is provided by Java

Search: String API Java

# COMMONLY USED STRING METHODS



//checks if s is equal to s2

```
boolean match = s.equals(s2);
```

//checks if s is equal to s2 regardless of capitalization

```
boolean match = s.equalsIgnoreCase(s2);
```

//tells you if it's alphabetically before

```
int result = s.compareTo(s2);
```

//tells you if it's alphabetically before disregarding capitalization

```
int result = s.compareToIgnoreCase(s2);
```

# COMMONLY USED STRING METHODS



//sets n to the length of the String

```
int n = s.length()
```

//returns the character at position pos

```
char ch = s.charAt(pos);
```

//returns a String from between two indices

```
String s2 = s.substring(fromPos);
```

```
String s2 = s.substring(fromPos, toPos);
```

//works same way as the +

```
String s2 = s.concat(str);
```

# COMMONLY USED STRING METHODS



//gets first index (a number) of a string or character

```
int k = s.indexOf(ch);
```

```
int k = s.indexOf(ch, fromPos);
```

```
int k = s.indexOf(str);
```

```
int k = s.indexOf(str, fromPos);
```

//gets last index of a string or character

```
int k = s.lastIndexOf(ch);
```

```
int k = s.lastIndexOf(ch, fromPos);
```

```
int k = s.lastIndexOf(str);
```

```
int k = s.lastIndexOf(str, fromPos);
```

# COMMONLY USED STRING METHODS



//s2 is set to s but all lowercase

```
String s2 = s.toUpperCase();
```

//s2 is set to s but all lowercase

```
String s2 = s.toLowerCase();
```

//s2 is set to same String as s but oldChar is replaced with newChar

```
String s2 = s.replace(oldChar, newChar);
```

//s2 is set to same String as s but w/o spaces at sides

```
String s2 = s.trim();
```

# CONVERTING NUMBERS INTO STRINGS (AND BACK)



## NUMBERS TO STRINGS

```
int n = 5;  
String s = "" + n;
```

```
int n = 5;  
String s = Integer.toString(n);
```

```
int n = 5;  
String s = String.valueOf(n);
```

## STRINGS TO NUMBERS

```
String s = "5";  
int n = Integer.parseInt(s);
```

It'll give you an error if the string is not a number

# COMMONLY USED CHARACTER METHODS

```
Character.isDigit(char)
```

```
Character.isLetter(char)
```

```
Character.isLetterOrDigit(char)
```

```
Character.isUpperCase(char)
```

```
Character.isLowerCase(char)
```

```
Character.toUpperCase('a');
```

```
Character.toLowerCase('V')
```





# EMPTY STRINGS VS. NO REFERENCE

## EMPTY STRING

- A String with this "" as its value
  - `String str = "";`
- You can use this string with methods
  - `"".equals("");`

## NO REFERENCE

- When you declare a String but don't initialize it
  - `String str;`
- The value of this String is null, so **you cannot use it with methods** or it will give you a `NullPointerException`