# Strings

A Sequence of Characters

## What is a String?

- String is NOT a primitive type
  - Primitive types are int, double, and boolean
- Strings are objects
  - Objects types are capitalized: the String type is capitalized; int, double, and boolean are not capitalized.
  - Strings contain data values and methods

## What's in a String?

- Strings have char data
  - char is another primitive data type for characters
  - char values have " " surrounding the characters
- Strings are special objects:
  - Normally, when creating new objects, you say:
    ObjectType objName = new ObjectType(arg);
  - Strings can be initialized with the actual data:
    String myStr = "Hello world!";
  - Strings can also initialized the usual way for objects:
    String myStr = new String("Hello world!");

## String Methods

- Strings are objects, objects have methods:
  - compareTo(Object other)
  - equals(Object other)
  - indexOf(String s)
  - length()
  - substring(int firstIndex)
  - substring(int firstIndex, int secondIndex)

## Compare strings

- Strings are not primitive types so you CANNOT use ==
- String s = "Java is cool";
  String t = "Java is cool";
  s == t → incorrect, may return false
  s.equals(t) → correct, returns true
- Can also use compareTo, which compares Strings lexicographically (see Javadoc for more info)

## Strings vs arrays

- Both Strings and arrays are objects
- Strings are like arrays of char
  - Strings access individual elements using charAt(int index)
  - "Hello world!".charAt(4) → o
  - Or "Hello world!".substring(4,5) → o
- Strings can access their length
  - "Hello world!".length() → 12

## length() vs length

- length() is a method for Strings
  - String s = “Hello world!”;
    s.length() → 12
    s.length → syntax error
- length is an instance variable for arrays
  - int[] arr = {3, 6, 11, 14, 18};
    arr.length() → syntax error
    arr.length → 5

## Substring methods

- Substring methods return a part of the original string
- String s = “Hello world!”;
  s.substring(1) → “ello world!”
  s.substring(6) → “world!”
  s.substring(9) → “ld!”
  s.substring(1, 4) → “ell”
  s.substring(6, 8) → “wo”
  s.substring(9,12) → “ld!”

## Strings are immutable

- Cannot change the String
  - Methods do not change the original string
  - String s = “Computer Science”;
    s.substring(10) → “cience”
    s → “Computer Science”
- Can change reference to String
  - Will discuss later when we talk about Object references

## + is a special method

- Normally methods need arguments in parentheses to work
  - String s = “Big”;
    s.concat(“ Data”) → “Big Data”
- + is a special method for Strings used for concatenation
  - String s = “Data”;
    s + “ Type” → “Data Type”
- += works too
  - String t = “Software”;
    t += “ Engineering”;
    t → “Software Engineering”