More on Strings

String methods and equality

Produced by: Mairead Meagher
Dr. Siobhán Drohan

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

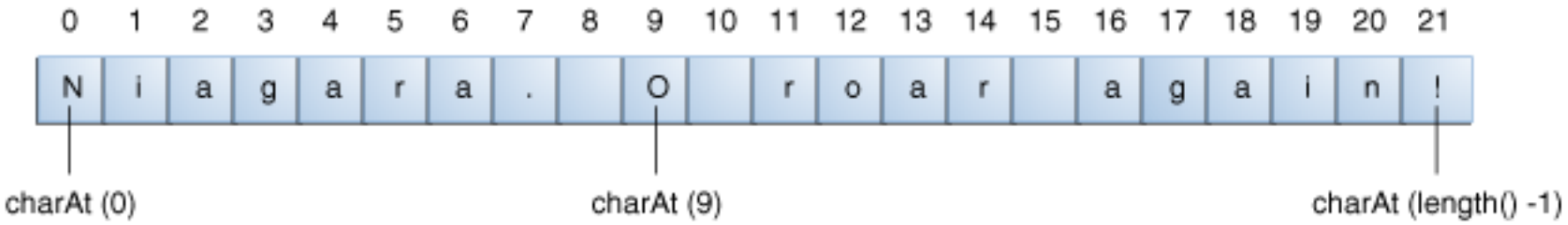
Department of Computing and Mathematics
http://www.wit.ie/

Topics list

- Strings: index of characters
- String methods:
  - charAt(int index)
  - substring(int beginIndex, int endIndex)
  - compareTo(String anotherString)
- Recap: Primitive vs object
- String Identity versus equality
- null
- Escape Sequences

Strings: index of characters

- A String holds a sequence of characters.
- The index of the first character in a String is 0.
- The index of the last character in a String is length()-1.



https://docs.oracle.com/javase/tutorial/java/data/manipstrings.html

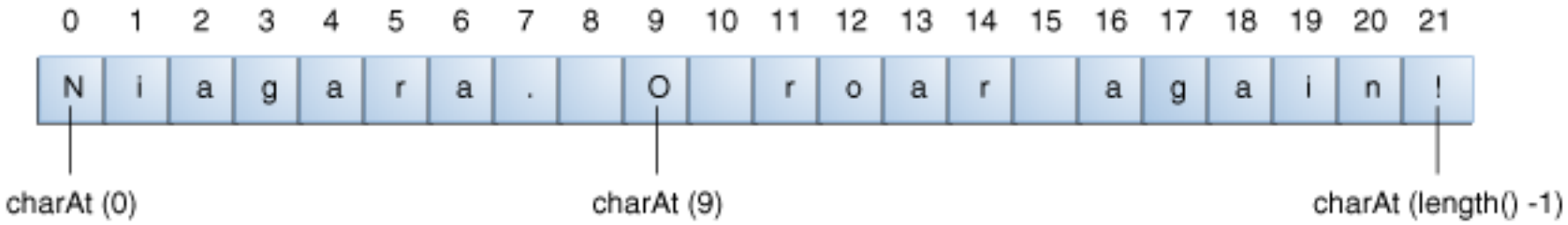
Topics list

- Strings: index of characters
- String methods:
  - charAt(int index)
  - substring(int beginIndex, int endIndex)
  - compareTo(String anotherString)
- Recap: Primitive vs object
- String Identity versus equality
- null
- Escape Sequences

String methods: charAt(int index)

- The following code gets the character at index 9 in a String:

```
String anotherPalindrome = "Niagara. O roar again!";
char aChar = anotherPalindrome.charAt(9);
```

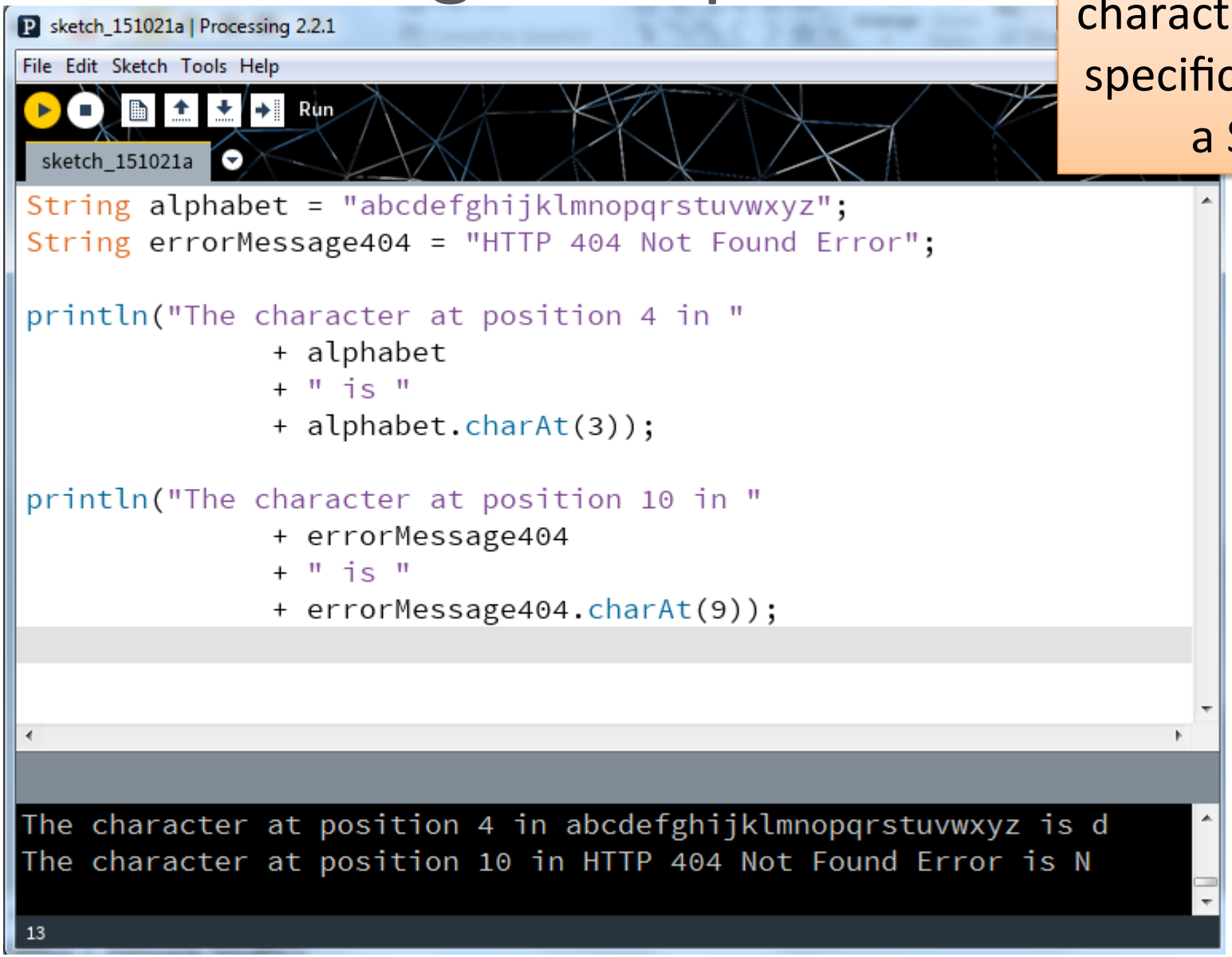


Indices begin at 0, so the character at index 9 is 'O'

https://docs.oracle.com/javase/tutorial/java/data/manipstrings.html

Processing example 7.1

Finding the character located a specific position in a String.



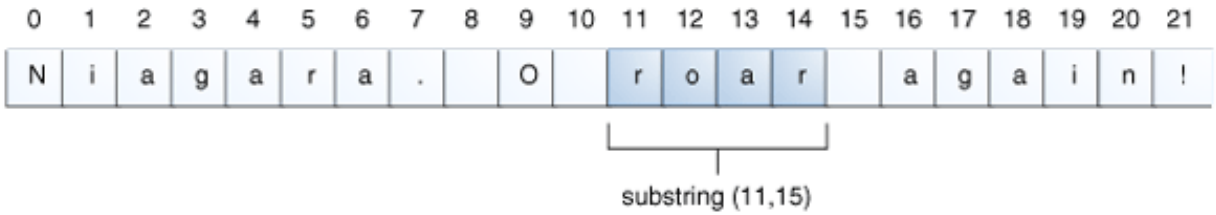
Topics list

- Strings: index of characters
- String methods:
  - charAt(int index)
  - substring(int beginIndex, int endIndex)
  - compareTo(String anotherString)
- Recap: Primitive vs object
- String Identity versus equality
- null
- Escape Sequences

String methods: substring(int beginIndex, int endIndex)

- This method returns a new String that is a substring of this String.
- The substring begins at the specified beginIndex and extends to the character at index endIndex – 1

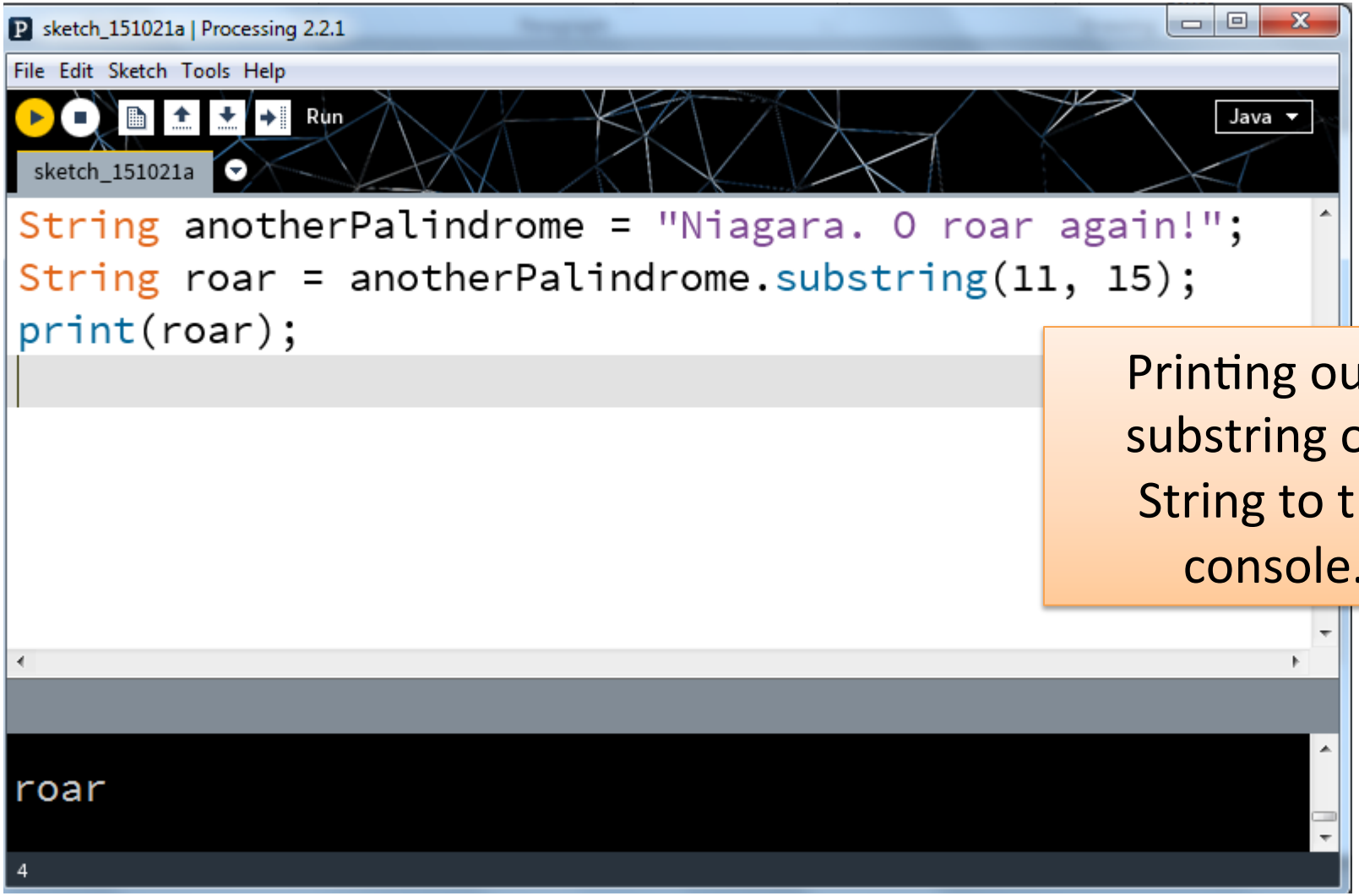
```
String anotherPalindrome = "Niagara. O roar again!";
String roar = anotherPalindrome.substring(11, 15);
```



This code returns a substring (“roar”) from anotherPalindrome. It extends from index 11 up to, but not including, index 15.

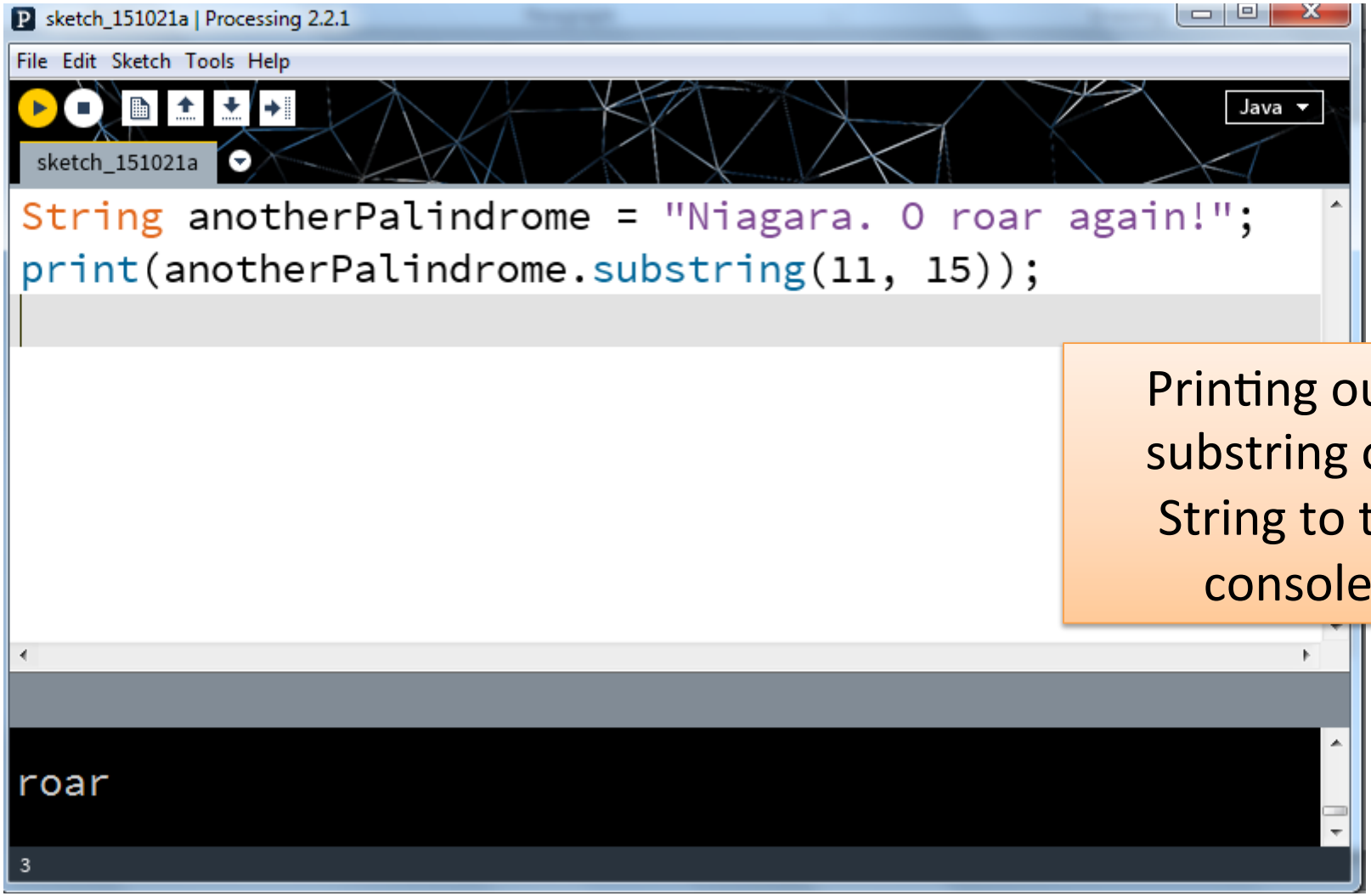
https://docs.oracle.com/javase/tutorial/java/data/manipstrings.html

Processing example 7.2, version 1



Printing out a substring of a String to the console.

Processing example 7.2, version 2



Printing out a substring of a String to the console.

Topics list

- Strings: index of characters
- String methods:
  - charAt(int index)
  - substring(int beginIndex, int endIndex)
  - compareTo(String anotherString)
- Recap: Primitive vs object
- String Identity versus equality
- null
- Escape Sequences

String methods: compareTo

int compareTo (String anotherString)

- This method compares two strings lexicographically i.e. based on the Unicode value of the characters in the String.
- It returns an integer indicating whether this string is:
  - greater than (result is > 0)
  - equal to (result is = 0) or
  - less than (result is < 0) the argument, anotherString.

https://docs.oracle.com/javase/tutorial/java/data/

Processing example 7.3

What will be printed to the console?
Which boolean expression evaluates to true?

```
String str1 = "Dog";
String str2 = "Cat";

if(str1.compareTo(str2) < 0){
    println(str1+" comes before "+ str2 +" in the alphabet");
}
else if(str1.compareTo(str2) > 0){
    println(str2 +" comes before "+ str1+" in the alphabet");
}
else{
   println("The strings are identical");
}
```

Processing example 7.3

```
String str1 = "Dog";
String str2 = "Cat";

if(str1.compareTo(str2) < 0){
    println(str1+" comes before "+ str2 +" in the alphabet");
}
else if(str1.compareTo(str2) > 0){
    println(str2 +" comes before "+ str1+" in the alphabet");
}
else{
   println("The strings are identical");
}
```

str1.compareTo(str2) returns a positive integer as Dog (str1) comes after Cat (str2).

Processing example 7.4

What will be printed to the console?
Which boolean expression evaluates to true?

```
String str1 = “cat";
String str2 = "Cat";

if(str1.compareTo(str2) < 0){
    println(str1+" comes before "+ str2 +" in the alphabet");
}
else if(str1.compareTo(str2) > 0){
    println(str2 +" comes before "+ str1+" in the alphabet");
}
else{
   println("The strings are identical");
}
```

Processing example 7.4

str1.compareTo(str2) returns a positive integer as cat (str1) comes after Cat (str2) in the Unicode character map.

```
String str1 = “cat";
String str2 = "Cat";

if(str1.compareTo(str2) < 0){
    println(str1+" comes before "+ str2 +" in the alphabet");
}
else if(str1.compareTo(str2) > 0){
    println(str2 +" comes before "+ str1+" in the alphabet");
}
else{
   println("The strings are identical");
}
```

Processing example 7.5

What will be printed to the console?
Which boolean expression evaluates to true?

```
String str1 = "Animal";
String str2 = "Cat";

if(str1.compareTo(str2) < 0){
    println(str1+" comes before "+ str2 +" in the alphabet");
}
else if(str1.compareTo(str2) > 0){
    println(str2 +" comes before "+ str1+" in the alphabet");
}
else{
   println("The strings are identical");
}
```

Processing example 7.5

str1.compareTo(str2) returns a negative integer as Animal(str1) comes before Cat (str2) in the Unicode character map.

```
String str1 = "Animal";
String str2 = "Cat";

if(str1.compareTo(str2) < 0){
    println(str1+" comes before "+ str2 +" in the alphabet");
}
else if(str1.compareTo(str2) > 0){
    println(str2 +" comes before "+ str1+" in the alphabet");
}
else{
   println("The strings are identical");
}
```

Processing example 7.6

What will be printed to the console?
Which boolean expression evaluates to true?

```
String str1 = “Cat";
String str2 = "Cat";

if(str1.compareTo(str2) < 0){
    println(str1+" comes before "+ str2 +" in the alphabet");
}
else if(str1.compareTo(str2) > 0){
    println(str2 +" comes before "+ str1+" in the alphabet");
}
else{
   println("The strings are identical");
}
```

Processing example 7.6

str1.compareTo(str2) returns 0 as Cat (str1) is identical to Cat (str2).

```
String str1 = “Cat";
String str2 = "Cat";

if(str1.compareTo(str2) < 0){
    println(str1+" comes before "+ str2 +" in the alphabet");
}
else if(str1.compareTo(str2) > 0){
    println(str2 +" comes before "+ str1+" in the alphabet");
}
else{
   println("The strings are identical");
}
```

Topics list

- Strings: index of characters
- String methods:
  - charAt(int index)
  - substring(int beginIndex, int endIndex)
  - compareTo(String anotherString)
- Recap: Primitive vs object
- String Identity versus equality
- null
- Escape Sequences

Recap: Object types e.g. String

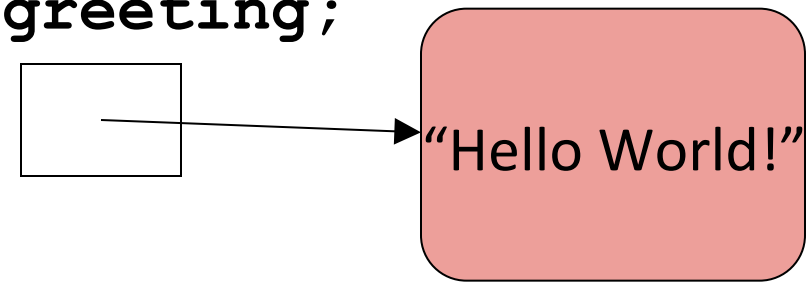
- Strings are a sequence of characters enclosed by double quotes (“ ”).
- String is an object type.
- The Java API provides information the String class and lists methods that can be used on Strings (https://docs.oracle.com/javase/8/docs/api/java/lang/String.html).
- The most direct way to create a String is to write:

```
String greeting = "Hello world!";
```

http://www.tutorialspoint.com/java/pdf/java_strings.pdf

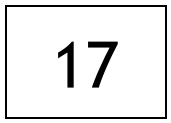
Recap: Primitive types vs. object types

```
String greeting;
```



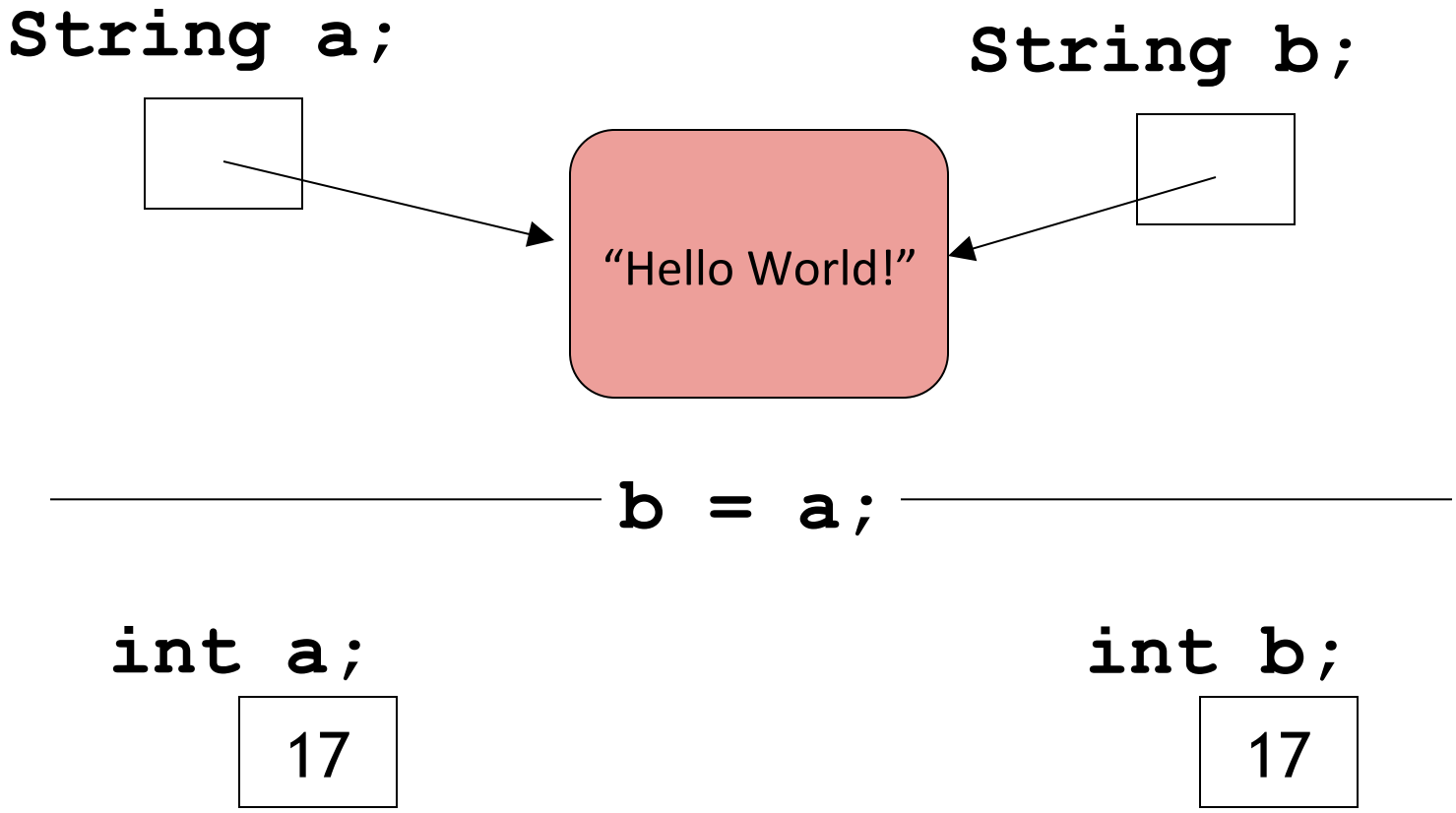
String is an object type. The **greeting** variable contains a reference to where the String is stored in memory.

```
int i;
```



primitive type

Recap: Primitive types vs. object types



Topics list

- Strings: index of characters
- String methods:
  - charAt(int index)
  - substring(int beginIndex, int endIndex)
  - compareTo(String anotherString)
- Recap: Primitive vs object
- String Identity versus equality
- null
- Escape Sequences

String: Identity vs Equality (1)

```
if(input == "bye") {
    ...
}

if(input.equals("bye")) {
    ...
}
```

tests identity i.e. the reference

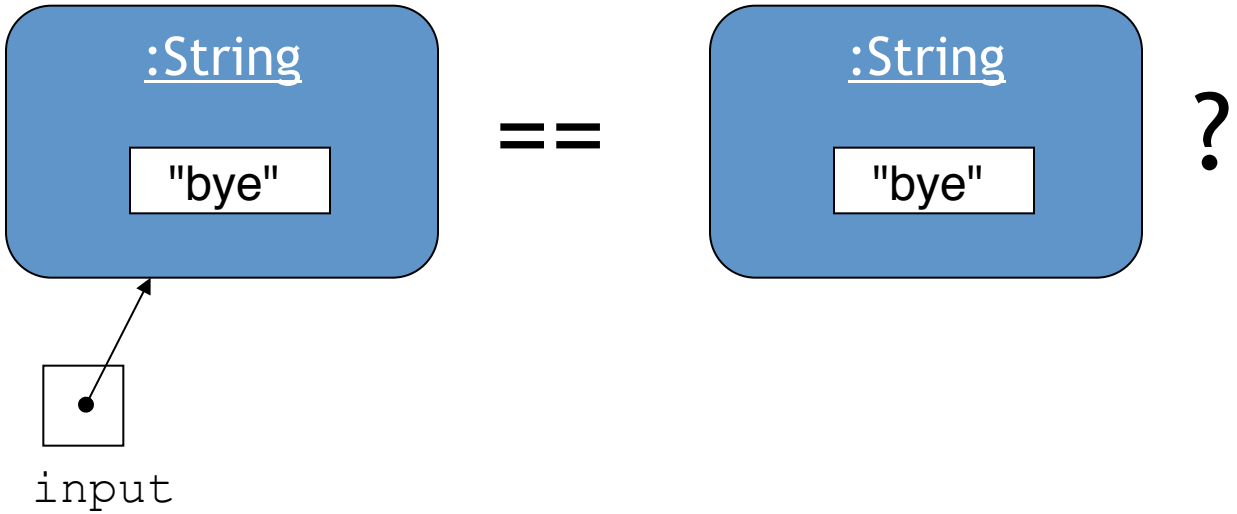
tests equality

Strings should always be compared using the `.equals` method

String: Identity vs Equality (2)

```
String input = "bye";
if(input == "bye") {
    ...
}
```

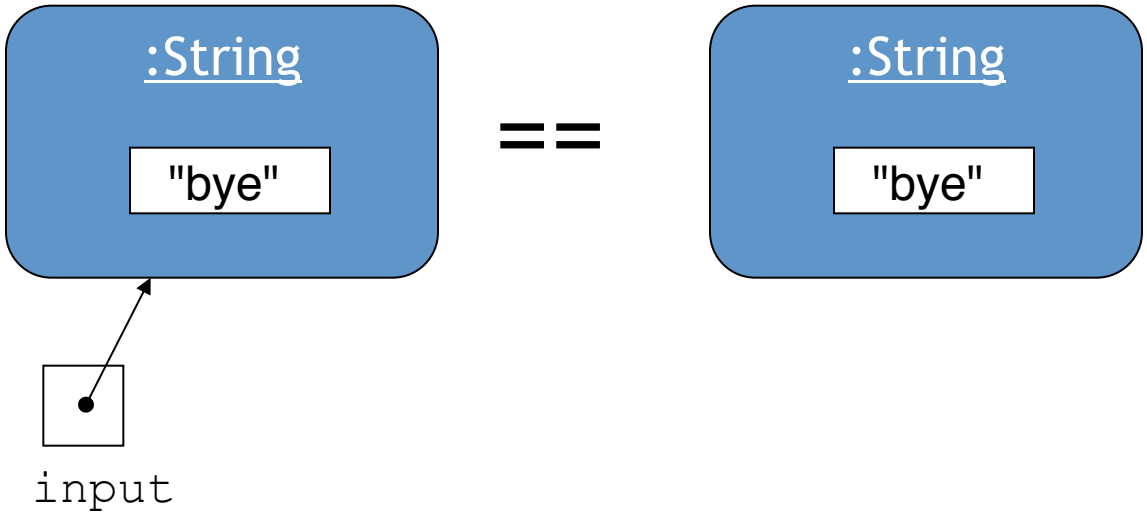
== tests identity



String: Identity vs Equality (2)

```
String input = "bye";
if(input == "bye") {
    ...
}
```

== tests identity

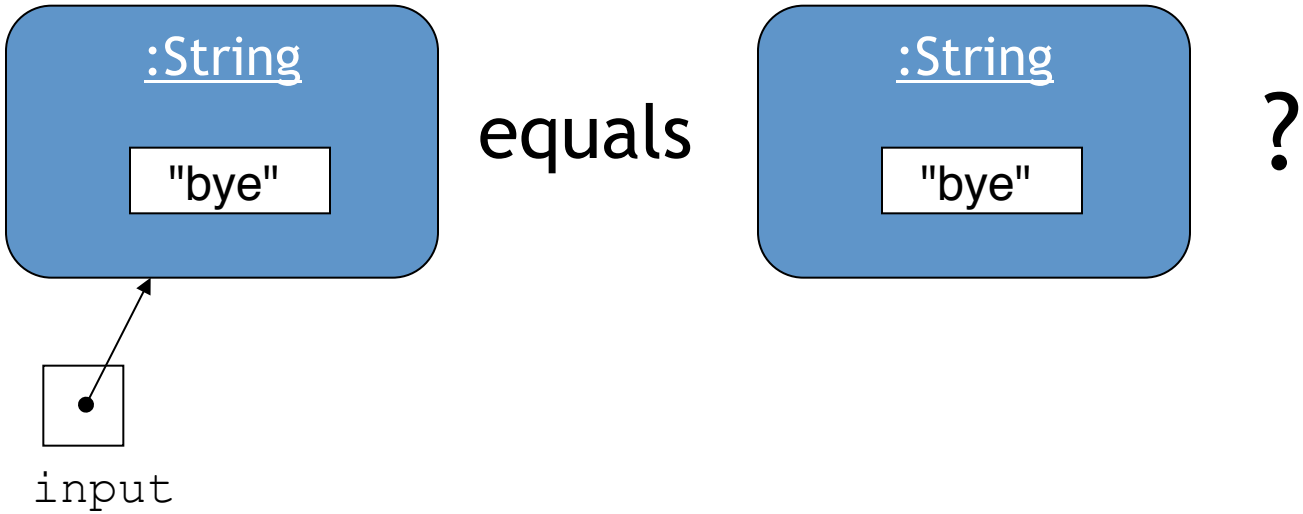


➠ (may be) false!

String: Identity vs Equality (3)

```
String input = "bye";
if(input.equals("bye")) {
    ...
}
```

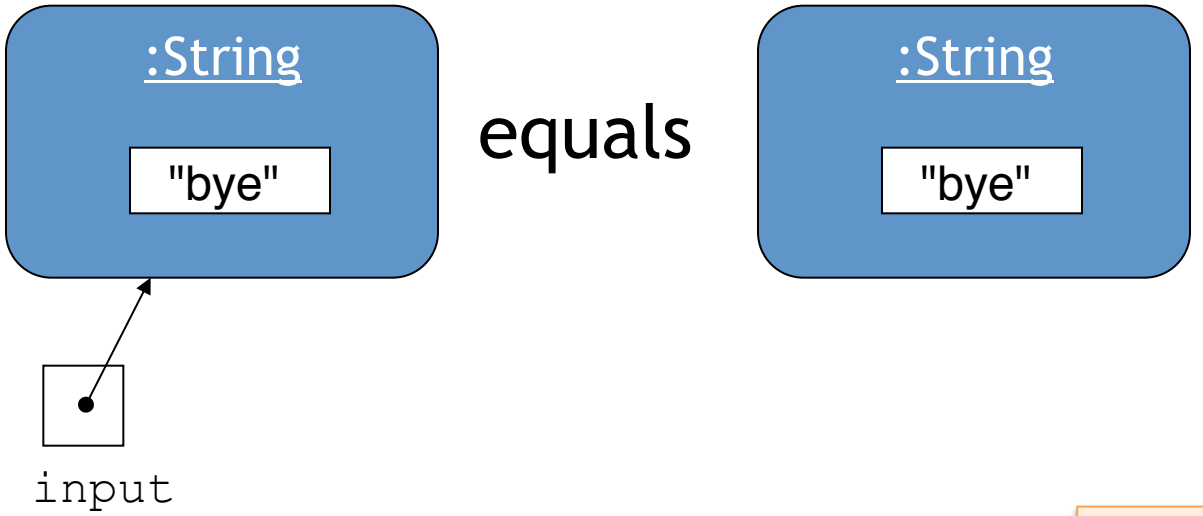
equals tests equality



String: Identity vs Equality (3)

```
String input = "bye";
if(input.equals("bye")) {
    ...
}
```

equals tests equality



➠ true!

Some common errors with Strings...

What's wrong here?

```
  void anyMethod()
  {
        String str1 = "a";
        String str2 = "b";

        if(str1 == str2)
        {
             println(str1+" is the same as
"+ str2);
        }
        else
        {
             println(str1+" is NOT same as
"+ str2);
        }
  }
```

Strings need to use the .equals method

```
void anyMethod()
{
        String str1 = "a";
        String str2 = "b";

        if(str1 == str2)
        {
              println(str1+" is the same as
"+ str2);
        }
        else
        {
              println(str1+" is NOT same as
"+ str2);
        }
}
```

What's wrong here?

```
  public void anyMethod()
  {
        int num1 = 1;
        int num2 = 2;

        if(num1 = num2)
             println(num1+" is the same as "+
num2);
        else
             println(num1+" is NOT same as "+
num2);
  }
```

You need two equals for equality

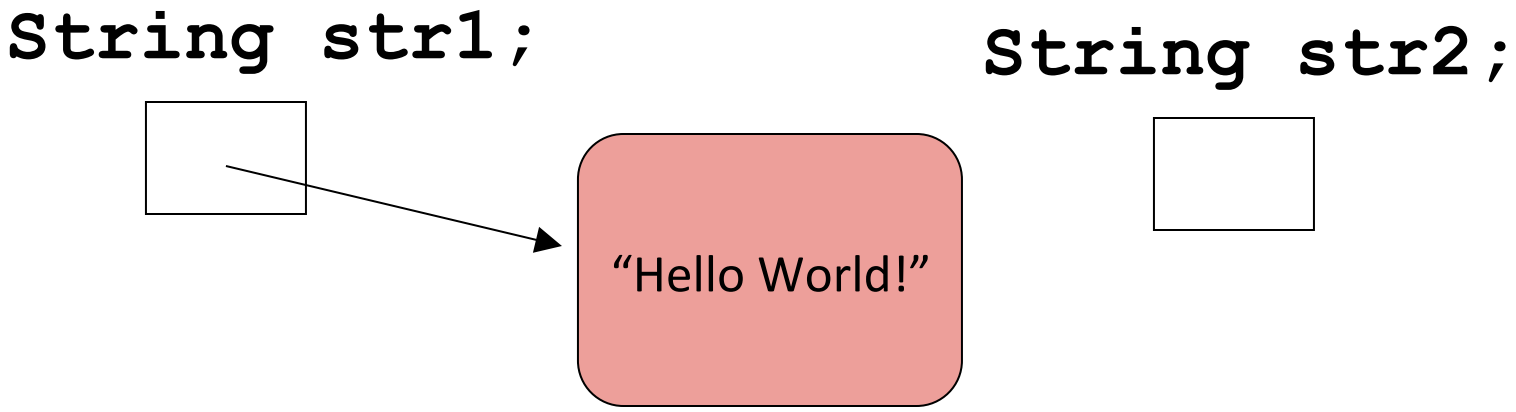
```
  public void anyMethod()
  {
        int num1 = 1;
        int num2 = 2;

        if(num1 = num2)
            println(num1+" is the same as "+
num2);
        else
            println(num1+" is NOT same as "+
num2);
  }
```

Topics list

- Strings: index of characters
- String methods:
  - charAt(int index)
  - substring(int beginIndex, int endIndex)
  - compareTo(String anotherString)
- Recap: Primitive vs object
- String Identity versus equality
- null
- Escape Sequences

null

- `null` is a special value in Java.
- All object variables are initialised to `null`.
- null means that the object variable does not have a reference e.g. str2 below.



null

- `null` is a special value in Java
- All object variables are initialised to `null`.
- You can assign and test for `null`:

```
private NumberDisplay hours;

if(hours == null) { ... }

hours = null;
```

Topics list

- Strings: index of characters
- String methods:
  - charAt(int index)
  - substring(int beginIndex, int endIndex)
  - compareTo(String anotherString)
- Recap: Primitive vs object
- String Identity versus equality
- null
- Escape Sequences

Escape sequences

- When a String is printed, certain single characters that follow a backslash (\) have special meaning…
- …and the compiler interprets them accordingly.

http://docs.oracle.com/javase/tutorial/java/data/characters.html

Java escape sequences

Escape Sequence	Description
\t	Insert a tab in the text at this point.
\b	Insert a backspace in the text at this point.
\n	Insert a newline in the text at this point.
\r	Insert a carriage return in the text at this point.
\f	Insert a formfeed in the text at this point.
\'	Insert a single quote character in the text at this point.
\"	Insert a double quote character in the text at this point.
\\	Insert a backslash character in the text at this point.

http://docs.oracle.com/javase/tutorial/java/data/characters.html

Examples of escape sequences

print(“Java\n”);

is the exact same as:

println(“Java”);

println(“ Java”);

is similar to:

println(“\tJava”);

http://docs.oracle.com/javase/tutorial/java/data/characters.html

Questions?

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics
http://www.wit.ie/