

Mrs Goode's copy

day 1

substring
and length
only

Introduction to Java

Unit 5. Notes: Strings are objects!

The String Class. The String class represents a sequence of characters. A character can

be a _____ . Strings are unusual in

that you do NOT have to use the new keyword or a constructor to instantiate

an object of this class. For example: String s1 = "hello, world"

can use String s2 = new String("bye, guys")

A series of characters enclosed in quotes is called a string literal

Each character has a unique location, starting at zero. The location of a character in a String is called its index. For example:

String name = "Abe.jumps"; // A is at index 0, j is at index 4

The **concatenation operator** is the plus sign (+). It is used to concatenate (join) two strings together. For example:

String word = "red";

String sentence = "the " + word + "bird" ; // string concatenation

String x = "3 days?";
 int n = x.length(); 7
 String y = "flower";
 String a = y.substring(4); er
 String b = y.substring(1); lower
 String c = "earth".substring(2); rth
 String d = y.substring(1,2); l
 String e = y.substring(1,3); low
 String f = y.substring(0, 4) flow
 String g = "box".substring(1,3); ox

for (int i = y.length() - 1; i > 2; i--)
 System.out.print(y.substring(i, i+1));

side right first!

The length method returns an int equals to the number of characters in the string.

What does the first line of the first substring method look like?

public String substring (int n)
 ↑ type ↑ name ↑ parameter list

What does the first line of the second substring method look like?

public String substring (int n1, int n2)
 ↑ 2 parameters start and end

length method?
 public int length ()

make clear : String a = y.substring(5);

a "r"

still have y "flower"

y = y.substring(5)

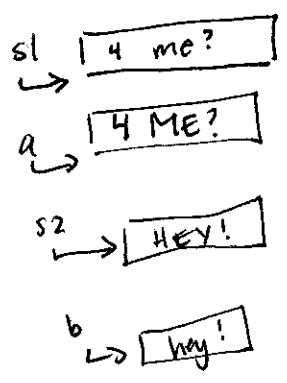
y "r"
you changed y

day 2

What does indexOf method return?

String s = "heater";		
int b = s.length();	<u>6</u>	What does the first line of the indexOf method look like?
int c = s.indexOf("a", 0);	<u>2</u>	public <u>int</u> indexOf (<u>String s, int index</u>)
int d = s.indexOf("heat", 0);	<u>0</u>	
int e = s.indexOf("eat", 1);	<u>1</u>	
int f = s.indexOf("e", 2);	<u>4</u>	
int g = s.indexOf("ice", 0);	<u>-1</u>	What does the indexOf method return if it cannot find what it is looking for?
int h = s.indexOf("a", 4);	<u>-1</u>	<u>-1</u>

String s1 = "4 me?";		What is the return type of the toUpperCase method?
String a = s1.toUpperCase();	<u>"4 ME?"</u>	public <u>String</u> toUpperCase()
String s2 = "HEY!";		What is the return type of the toLowerCase method?
String b = s2.toLowerCase();	<u>"hey!"</u>	public <u>String</u> toLowerCase()
if (b == "hey!")		If you need to check if two strings are equal, do NOT <u>==</u>
System.out.println("ONE");		<u>use .equals instead</u>
else		public boolean equals(Object obj)
System.out.println("TWO");		The equals method is case-sensitive. "A" is not equal to "a"
if (b.equals("hey!"))		
System.out.println("THREE");		
else		
System.out.println("FOUR");		



The Scanner Class. Scanner objects are used to read streams of text. The text could come from the keyboard or a file. The table below describes three methods of the Scanner class.

Return Type	Method Name and Parameters	Description
double	nextDouble()	Ignores all white space and returns the first double it finds. It does not delete the new line character from the input stream.
int	nextInt()	Ignores all white space and returns the first integer it finds.
String	nextLine()	Returns all characters including white space up to, but not including, the new line characters. Deletes the new line character.

Some subtleties of the Scanner class. The Scanner actually deals with "streams" of data. For example, consider the code snippet below.

```
Scanner get = new Scanner(System.in);
System.out.print( "Enter the x and y coordinates " );
int x = get.nextInt();
int y = get.nextInt();
```

If the user types "7 8 9" (one space between each number) and then hits the Enter key, the input stream consists of 6 characters. What happens?

The first call to nextInt() 7 is returned, rest ignored
 The second call to nextInt() 8 is returned, the space ignored as well as newline char /n

```
System.out.print( "Enter your age and name " );
int age = get.nextInt();
String name = get.nextLine();
```

If the user types "16 Moe" and then hits the Enter key, the input stream consists of 7 characters. What happens?

The call to nextInt() 16 returns 16, rest ignored
 The call to nextLine() Moe w/ space & deletes newline char /n

```
String s = "hello";
char c = s.charAt(3); // c holds 'l'
// chars denoted w 'e', strings with "e"
String newString = s.toUpperCase(); // newString holds "HELLO"
// s holds "hello"
String hold = "Hey You".toLowerCase();
hold is _____
```

show Scanner methods & String methods
 java . api .

