**Java String Equals and Loops**

* Compare two Strings:
* a.equals(b)
* **Do not** use ==
* Sadly == compiles, but does not work reliably .. a real trap
* In retrospect, an error in the design of Java

**String Equals**

Use the equals() method to check if 2 strings are the same. The equals() method is case-sensitive, meaning that the string "HELLO" is considered to be different from the string "hello". The == operator does not work reliably with strings. Use == to compare primitive values such as int and char. Unfortunately, it's easy to accidentally use == to compare strings, but it will not work reliably. Remember: use equals() to compare strings. There is a variant of equals() called equalsIgnoreCase() that compares two strings, ignoring uppercase/lowercase differences.

String a = "hello";

String b = "there";

if (a.equals("hello")) {

// Correct -- use .equals() to compare Strings

}

if (a == "hello") {

// NO NO NO -- do not use == with Strings

}

// a.equals(b) -> false

// b.equals("there") -> true

// b.equals("There") -> false

// b.equalsIgnoreCase("THERE") -> true

**String For Loop**

* Super-common string for-loop
* Loop to hit each index number once:
* 0, 1, 2, ... length-1
* for (int i = 0; i < str.length(); i++) {

for (int i = 0; i < str.length(); i++) {

// do something at index i

}

* Strategy 1: straight use of standard loop -- great
* Strategy 2: standard loop + some variation

CodingBat Practice> [countXX](http://codingbat.com/prob/p194667)

More practice problems

* CodingBat Practice> [last2](http://codingbat.com/prob/p178318)
* CodingBat Practice> [stringMatch](http://codingbat.com/prob/p198640)

[CodingBat.com](http://codingbat.com/) code practice. Copyright 2012 Nick Parlante.