

Strings

Part Two

H e l l o !

H	e	l	l	o	!
---	---	---	---	---	---

0

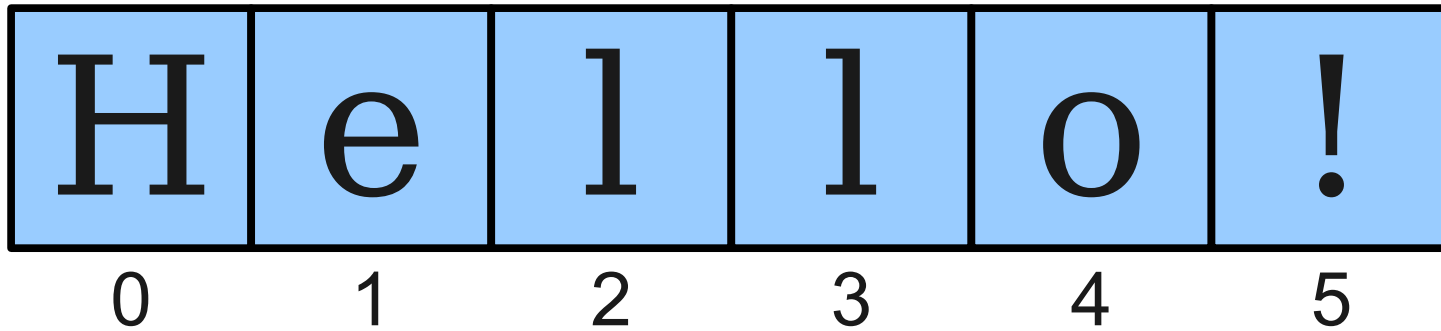
1

2

3

4

5



string.charAt (***index***)

The Data Type **char**

- The primitive type **char** represents a single character or glyph.
- Some examples:

```
char letterA = 'A';
```

```
char plus = '+'
```

```
char zero = '0';
```

Escape Sequences

- An **escape sequence** is a sequence of characters in a program's source code that represents a single logical character.
- Examples:
 - `\t`: Horizontal tab
 - `\n`: Newline
 - `\'`: Single quote
 - `\"`: Double quote

Testing Properties of Characters

boolean Character.isDigit(char ch)

Determines if the specified character is a digit.

boolean Character.isLetter(char ch)

Determines if the specified character is a letter.

boolean Character.isLetterOrDigit(char ch)

Determines if the specified character is a letter or a digit.

boolean Character.isLowerCase(char ch)

Determines if the specified character is a lowercase letter.

boolean Character.isUpperCase(char ch)

Determines if the specified character is an uppercase letter.

boolean Character.isWhitespace(char ch)

Determines if the specified character is **whitespace** (spaces and tabs).

char Character.toLowerCase(char ch)

Converts **ch** to its lowercase equivalent, if any. If not, **ch** is returned unchanged.

char Character.toUpperCase(char ch)

Converts **ch** to its uppercase equivalent, if any. If not, **ch** is returned unchanged.

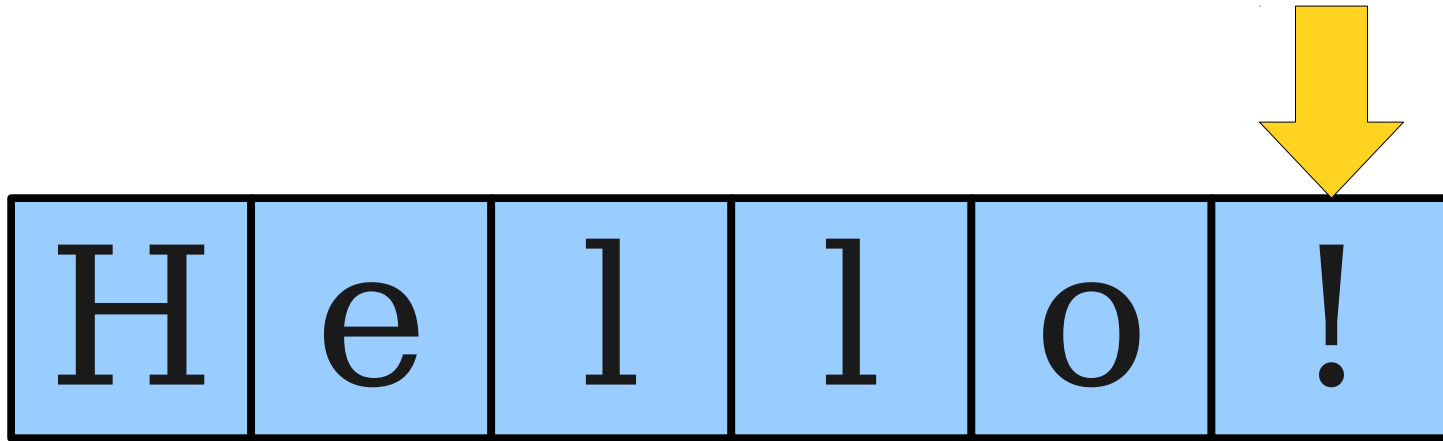


http://deathandtaxesmag.wpengine.netdna-cdn.com/wp-content/uploads/2013/01/quokka_1.jpg

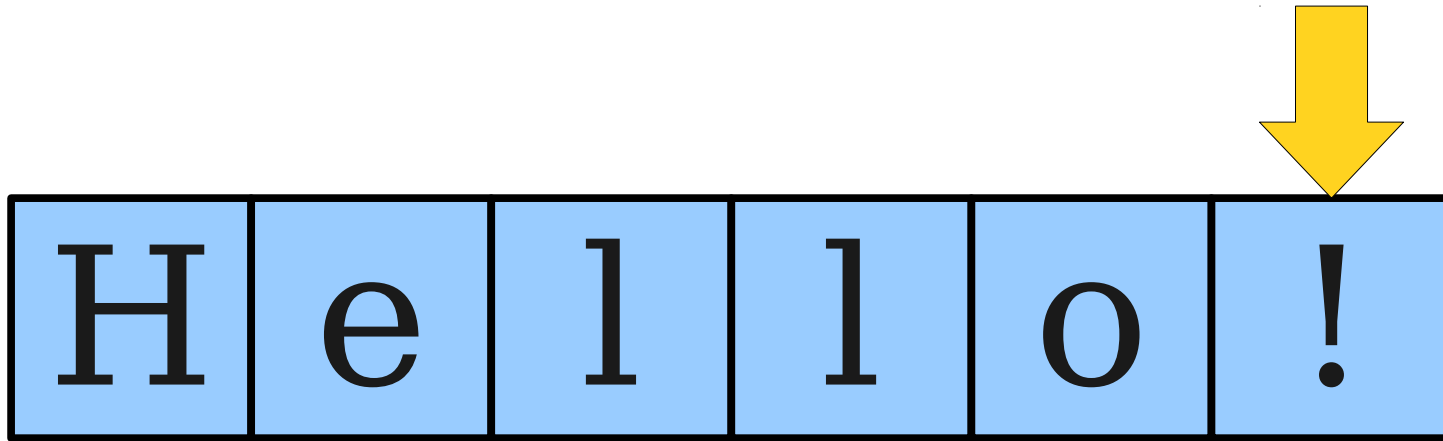
Strings are Immutable

- Java strings are **immutable**: once a string has been created, its contents cannot change.
- To change a string:
 - Create a new string holding the new value you want it to have.
 - Reassign the String variable to hold the new value.
- **Important consequence**: if you pass a String into a method, that method cannot modify that string.

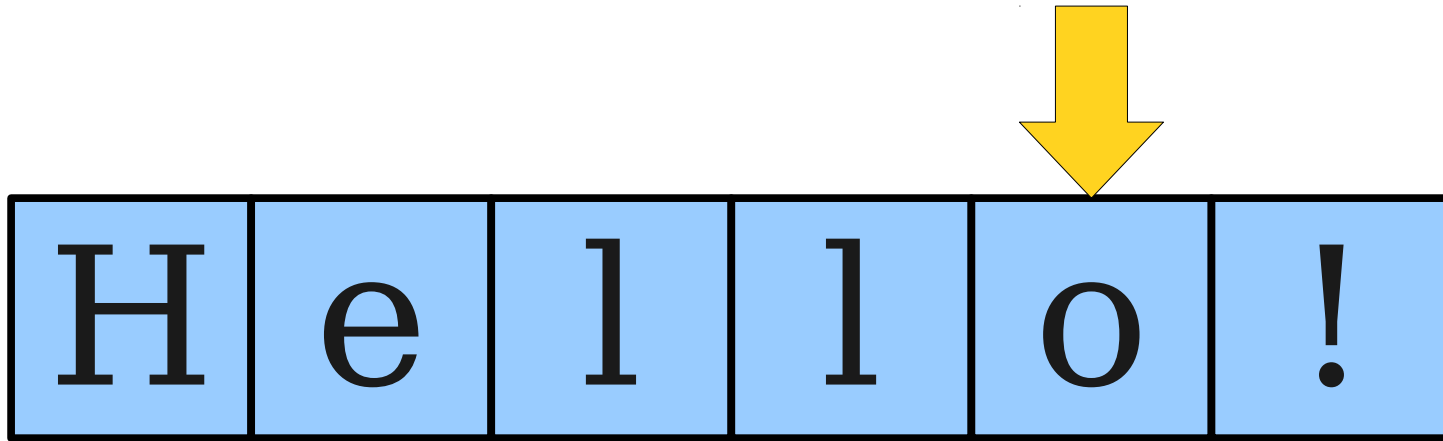
Reversing a String



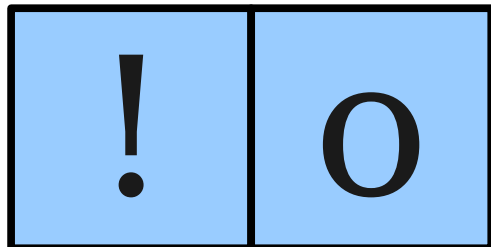
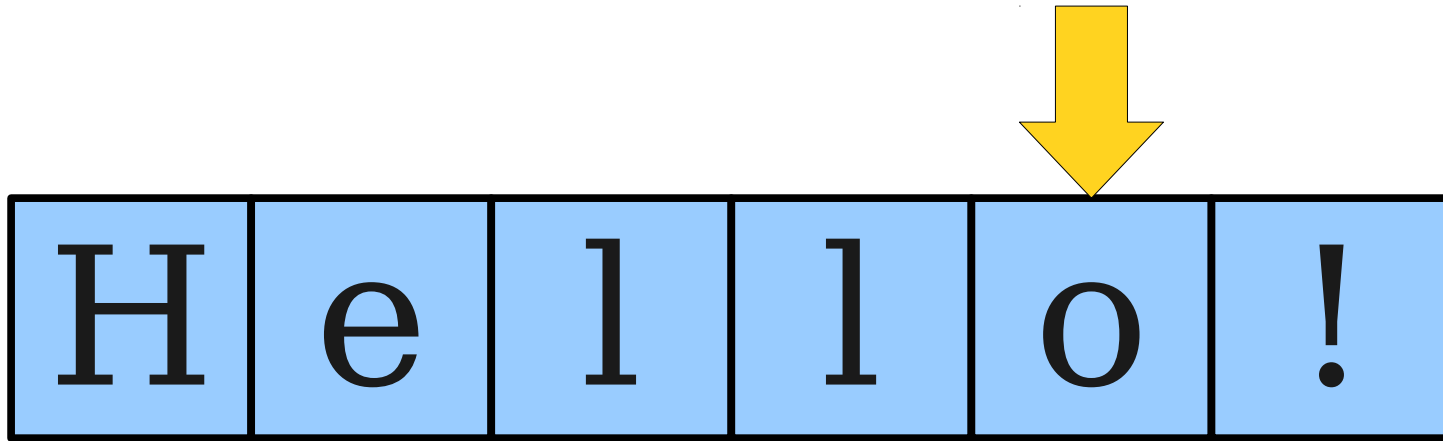
Reversing a String



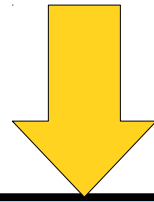
Reversing a String



Reversing a String



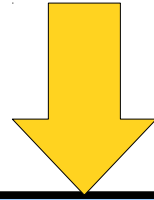
Reversing a String



H	e	l	l	o	!
---	---	---	---	---	---

!	o
---	---

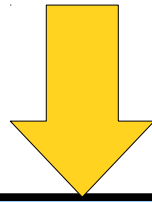
Reversing a String



H	e	l	l	o	!
---	---	---	---	---	---

!	o	l
---	---	---

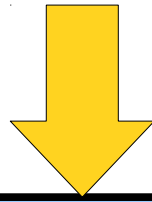
Reversing a String



H	e	l	l	o	!
---	---	---	---	---	---

!	o	l
---	---	---

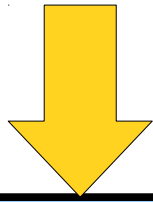
Reversing a String



H	e	l	l	o	!
---	---	---	---	---	---

!	o	l	l
---	---	---	---

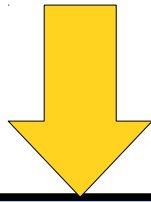
Reversing a String



H	e	l	l	o	!
---	---	---	---	---	---

!	o	l	l
---	---	---	---

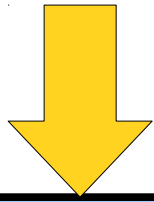
Reversing a String



H	e	l	l	o	!
---	---	---	---	---	---

!	o	l	l	e
---	---	---	---	---

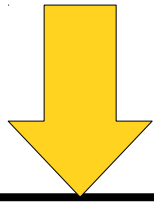
Reversing a String



H	e	l	l	o	!
---	---	---	---	---	---

!	o	l	l	e
---	---	---	---	---

Reversing a String



H	e	l	l	o	!
---	---	---	---	---	---

!	o	l	l	e	H
---	---	---	---	---	---

Reversing a String

H	e	l	l	o	!
---	---	---	---	---	---

!	o	l	l	e	H
---	---	---	---	---	---

Time-Out for Announcements!

Assignment 3

- Assignment 3 due on Monday at 3:15PM.
- **Recommendation:** Try to get the game completed by Friday so you have time to test and add extensions by Monday.
- Have questions?
 - Stop by the LaIR!
 - Stop by office hours!
 - Email your section leader!
 - Ask on QuestionHut!

Midterm Information

- Review session: **Sunday, 1PM - 4PM** in **Hewlett 200**.
 - Come with questions, leave with answers!
- Solutions to first practice exam released.
- Second practice exam released.
 - Actual exam from Winter 2011-2012.
 - Solutions released on Friday.
- Feel free to ask questions!

Back to CS106A!

Palindromes

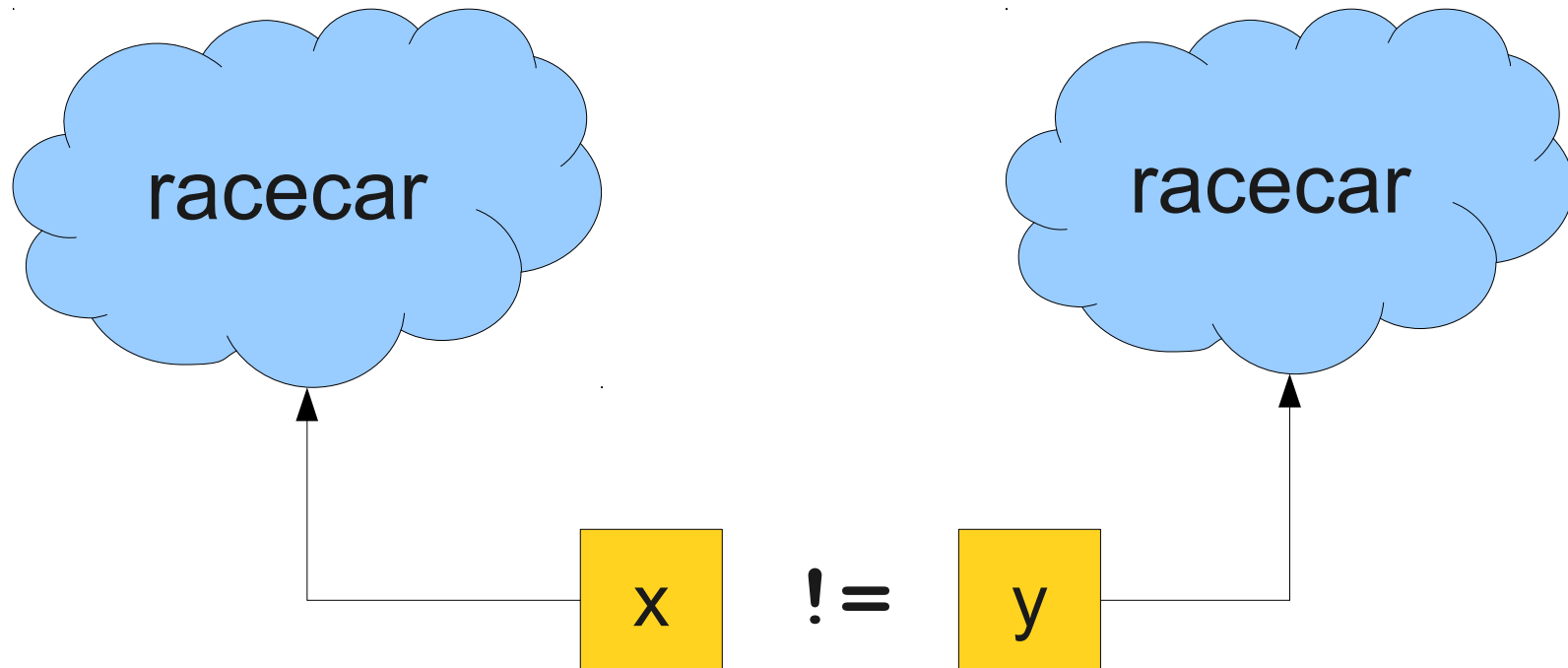
- A **palindrome** is a string that reads the same forwards and backwards.
- For example:
 - Racecar
 - Kayak
 - Mr. Owl ate my metal worm.
 - Go hang a salami! I'm a lasagna hog.

Checking for Palindromes

What Went Wrong?

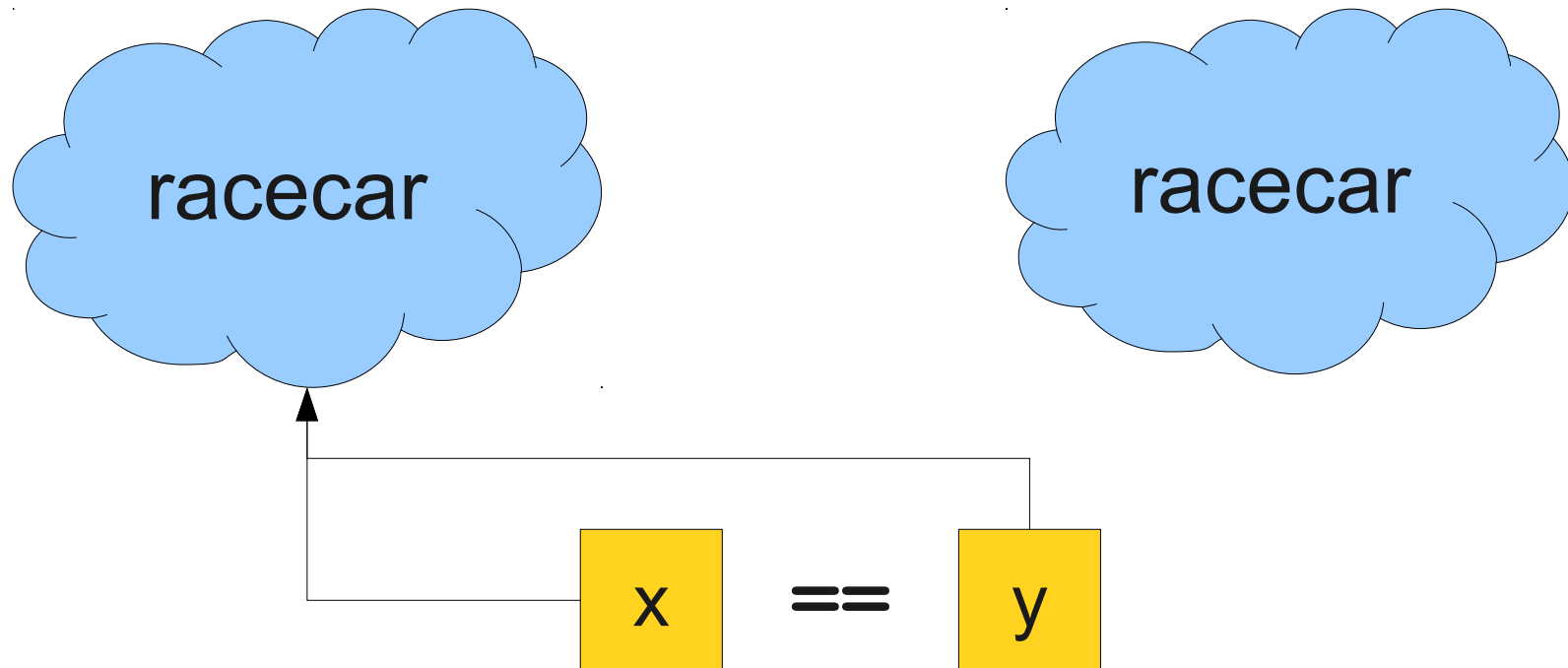
The == Operator

- When applied to objects, the == operator reports whether the two objects are the same object, not whether the *values* of those objects are equal.



The == Operator

- When applied to objects, the == operator reports whether the two objects are the same object, not whether the *values* of those objects are equal.



Comparing Strings for Equality

- To determine if two strings are equal, use the `.equals()` method:

```
String s1 = "racecar";  
String s2 = reverseString(s1);  
if (s1.equals(s2)) {  
    /* ... s1 and s2 are equal ... */  
}
```