Strings
Part Two
Hello!
Hello!
Hello!

```
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

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>boolean Character.isDigit(char ch)</code></td>
<td>Determines if the specified character is a digit.</td>
</tr>
<tr>
<td><code>boolean Character.isLetter(char ch)</code></td>
<td>Determines if the specified character is a letter.</td>
</tr>
<tr>
<td><code>boolean Character.isLetterOrDigit(char ch)</code></td>
<td>Determines if the specified character is a letter or a digit.</td>
</tr>
<tr>
<td><code>boolean Character.isLowerCase(char ch)</code></td>
<td>Determines if the specified character is a lowercase letter.</td>
</tr>
<tr>
<td><code>boolean Character.isUpperCase(char ch)</code></td>
<td>Determines if the specified character is an uppercase letter.</td>
</tr>
<tr>
<td><code>boolean Character.isWhitespace(char ch)</code></td>
<td>Determines if the specified character is <code>whitespace</code> (spaces and tabs).</td>
</tr>
<tr>
<td><code>char Character.toLowerCase(char ch)</code></td>
<td>Converts <code>ch</code> to its lowercase equivalent, if any. If not, <code>ch</code> is returned unchanged.</td>
</tr>
<tr>
<td><code>char Character.toUpperCase(char ch)</code></td>
<td>Converts <code>ch</code> to its uppercase equivalent, if any. If not, <code>ch</code> is returned unchanged.</td>
</tr>
</tbody>
</table>

Based on slides by Eric Roberts and Mehran Sahami
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

Hello!
Reversing a String

Hello!
Reversing a String

Hello!
Reversing a String

Hello!

! o
Reversing a String

Hello!

! o
Reversing a String

Hello!

! o l
Reversing a String

Hello!

!0l
Reversing a String

Hello!

! o l l l
Reversing a String

Hello!

! o l l
Reversing a String

Hello!

!olleol
Reversing a String

Hello!

!ollel
Reversing a String

Hello!

!olleH
Reversing a String

Hello!

!olleH
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:

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