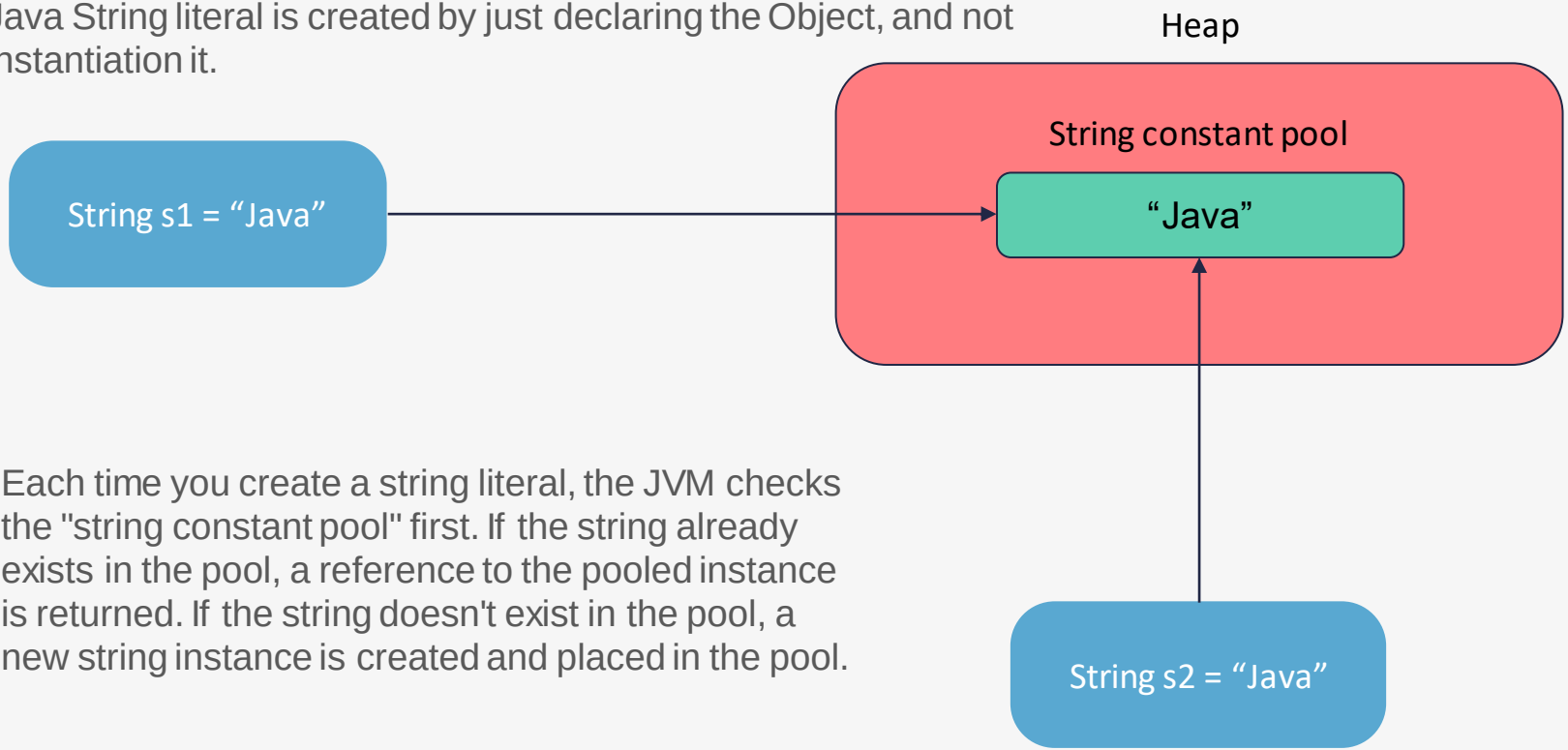
String

TestLeaf
Always Ahead

Initialization & Memory

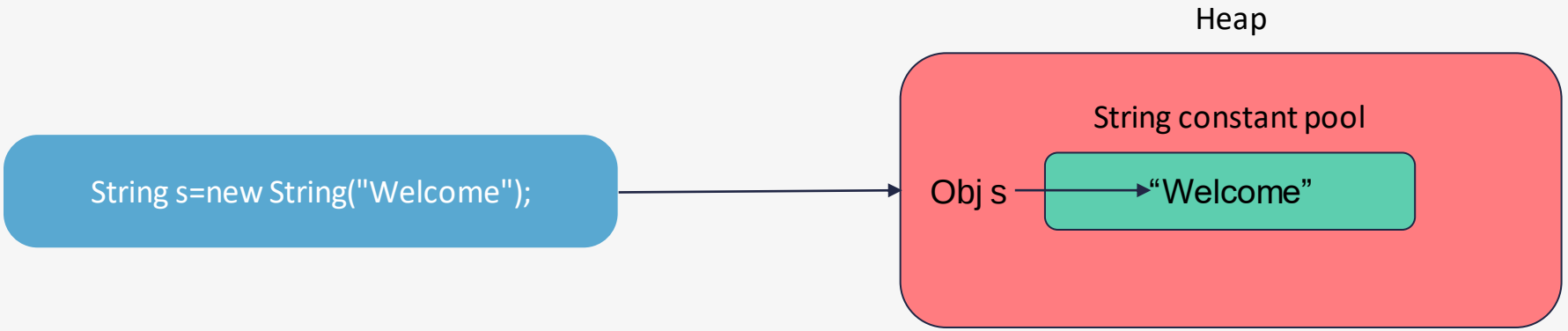
String Literals

Java String literal is created by just declaring the Object, and not instantiation it.



Each time you create a string literal, the JVM checks the "string constant pool" first. If the string already exists in the pool, a reference to the pooled instance is returned. If the string doesn't exist in the pool, a new string instance is created and placed in the pool.

By “**new**” Keyword



JVM will create a new string object in normal (non-pool) heap memory, and the literal "Welcome" will be placed in the string constant pool. The variable s will refer to the object in a heap (non-pool).

Methods

- toCharArray();
- intern()
- codePointAt(int index)
- charAt(int index)
- contains(String value)
- concat(String value)
- endWith(String value), and startsWith(String value)
- equals(String value), and equalsIgnoreCase(String value)
- length(), and trim(), and isEmpty()
- indexOf(String value), and lastIndexOf(String value)
- valueOf(T arr[])
- split(String value)
- subString(String value)
- replace(char oldChar, char newChar), and replaceAll(String old,String new)