

Working with Strings in Java



For-each loops and Strings



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/)
by Christine Alvarado, Mia Minnes, and Leo Porter, 2015.

By the end of this video you will be able to...

- Use for-each loops to manipulate Strings and other arrays

```
public static boolean hasLetter(String word, char letter)
{
    for (char c : word.toCharArray())
    {
        if (c == letter)
        {
            return true;
        }
    }
    return false;
}
```

```
public static String replace(String word,  
                             char gone, char here)  
{  
    char[] cArray = word.toCharArray();  
    for (char c : cArray)  
    {  
        if (c == gone)  
        {  
            c = here;  
        }  
    }  
    return word;  
}
```

```
public static String replace(String word,  
                             char gone, char here)  
{  
    char[] cArray = word.toCharArray();  
    for (char c : cArray)  
    {  
        if (c == gone)  
        {  
            c = here;  
        }  
    }  
    return word;  
}
```

```
// somewhere else:  
replace("a happy", 'a', 'i')  
→ "i hippy" ??
```

```
public static String replace(String word,
                             char gone, char here)
{
    char[] cArray = word.toCharArray();
    for (char c : cArray)
    {
        if (c == gone)
        {
            c = here;
        }
    }
    return word;
}
```

**Does this method
successfully return a
modified word?**

```
public static String replace(String word,
                             char gone, char here)
{
    char[] cArray = word.toCharArray();
    for (char c : cArray)
    {
        if (c == gone)
        {
            c = here;
        }
    }
    return word;
}
```

```
// somewhere else:
replace("a happy", 'a', 'i')
→ "i hippy" ??
```

```
public static String replace(String word,  
                             char gone, char here)  
{  
    char[] cArray = word.toCharArray();  
    for (char c : cArray)  
    {  
        if (c == gone)  
        {  
            c = here;  
        }  
    }  
    return word;  
}
```

**toCharArray() returns a
copy of word's array of
chars**


```
public static String replace(String word,  
                             char gone, char here)  
{  
    char[] cArray = word.toCharArray();  
    for (char c : cArray)  
    {  
        if (c == gone)  
        {  
            c = here;  
        }  
    }  
    return new String(cArray);  
}
```

**Does this method
successfully return a
modified word?**

```
public static String replace(String word,
                             char gone, char here)
{
    char[] cArray = word.toCharArray();
    for (char c : cArray)
    {
        if (c == gone)
        {
            c = here;
        }
    }
    return new String(cArray);
}
```

```
// somewhere else:
replace("a happy", 'a', 'i')
→ "i hippy" ??
```

```
public static String replace(String word,  
                             char gone, char here)  
{  
    char[] cArray = word.toCharArray();  
    for (char c : cArray)  
    {  
        if (c == gone)  
        {  
            c = here;  
        }  
    }  
    return new String(cArray);  
}
```

**c gets a copy of each
value in cArray!**

```
public static String replace(String word,  
                             char gone, char here)  
{  
    char[] cArray = word.toCharArray();  
    char[] cArrayMod = new char[cArray.length],  
    int i = 0;  
    for (char c : cArray) {  
        if (c == gone)  
            cArrayMod[i] = here;  
        else  
            cArrayMod[i] = c;  
        i++;  
    }  
    return new String(cArrayMod);  
}
```

**Does this method
successfully return a
modified word?**

```
public static String replace(String word,
                             char gone, char here)
{
    char[] cArray = word.toCharArray();
    char[] cArrayMod = new char[cArray.length];
    int i = 0;
    for (char c : cArray) {
        if (c == gone)
            cArrayMod[i] = here;
        else
            cArrayMod[i] = c;
        i++;
    }
    return new String(cArrayMod);
}
```

```
// somewhere else:
replace("a happy", 'a', 'i')
    → "i hippy" ??
```

```
public static String replace(String word,
                             char gone, char here)
{
    char[] cArray = word.toCharArray();
    char[] cArrayMod = new char[cArray.length];
    int i = 0;
    for (char c : cArray) {
        if (c == gone)
            cArrayMod[i] = here;
        else
            cArrayMod[i] = c;
        i++;
    }
    return new String(cArrayMod);
}
```

**Does this method
modify the
parameter String,
word?**