Loops and Strings

For Loop

```
for (int count = 0; count <= 5; count++)
{
        System.out.println(count);
}
```

While Loop

```
int count = 5;

while (count <= 10)

{

    System.out.println(count);

    count++;

 }
```

String Methods

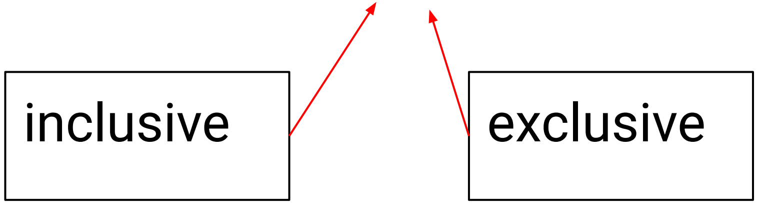
Method	**Use**
name.**substring(int start, int end)**	returns a copy of the string between the two indices excluding the end
name.**substring(int start)**	returns a copy of the string starting at the index, up until the end
name.**equals(Object another)**	return true if the strings have identical contents
name.**length()**	returns the number of characters in str
name.**compareTo(String another)**	for less than/ greater than / equal comparison
name.**charAt(ind index)**	return the character at the index position of the String

How can we traverse Strings?

I want to print each character in a String on a new line using substring

```
String msg = “hello, smile!”;

for (int i = 0; i < msg.length(); i++){
        System.out.println(msg.substring(i, i+1));
}
```



inclusive

exclusive

<= would cause an exception - WHY?

How can we traverse Strings?

I want to print each character in a String on a new line using charAt

```
String msg = “hello, smile!”;

for (int i = 0; i < msg.length(); i++){
        char character = msg.charAt(i);
        System.out.println(character);
}
```

How can we traverse Strings?

I want to print each character in a String on a new line using substring and a while loop

```
String msg = “hello, smile!”;

int i = 0;

while (i < msg.length()){

        System.out.println(msg.substring(i, i +1));

        i++;

}
```

How we can determine how many letters ‘e’ are in the string?

```
public int countLetter(String msg){
    int counter = 0;
    for (int i = 0; i <msg.length(); i++){
        char character = msg.charAt(i);
        if (character == ‘e’){
            counter++;
        }
    }
    System.out.println(counter);
    return counter;
}
```

Some algorithms that use loops and strings

- Build from an empty string: Given a string and a non-negative int n, return a larger string that is n copies of the original string.

  buildString("AP", 2) → "APAP"

- Compare substrings: Count the number of "aa" in the given string. Overlapping is allowed (“aaa" contains 2 "aa")

  countSubstring("aaaa") → 3

Visualizer

You may add print statements when you work with loops and strings to see what is done after each iteration.

You may also use this helpful tool:

https://cscircles.cemc.uwaterloo.ca/java_visualize