

The String Class

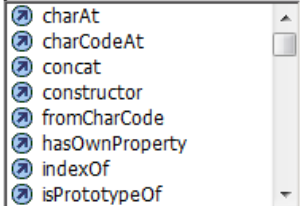
The String class is built-in to Flash. Just as there are many operations we can do with numbers, such as adding and subtracting, there are many operations we can do with Strings.

- Start a new ActionScript 3 file.
- In the ActionScript window declare a String variable with an initial value:

```
var s:String="ABCDEFGHIJKLMNOPQRSTUVWXYZ";
trace(s.
```

When you type s. the list of methods will appear.

```
var s:String="ABCDEFGHIJKLMNOPQRSTUVWXYZ";
trace(s.
```



Type `trace(s.charAt(2));`

You will see the letter C in the output window. Like most things in computers, numbering starts with 0. Thus, `s.charAt(0)` is the letter A.

Try each of these commands, retyping the dot each time so that you can pick from the list.

Command	Output
<code>trace(s.indexOf("C"));</code>	2
<code>trace(s.length);</code>	26
<code>trace(s.replace("BC", "***"));</code>	A**DEFGHIJKLMNOPQRSTUVWXYZ
<code>trace (s.substr(3,2));</code> <i>(substring of s beginning in 3 for a length of 2)</i>	DE
<code>trace(s.substr(20));</code> <i>(substring of s beginning in 20 for the rest of the string)</i>	UVWXYZ
<code>trace(s.toLowerCase());</code>	abcdefghijklmnopqrstuvwxyz

The `split` method is used to create an array. If we type `trace(s.split(""))`; we will see the alphabet with commas between: A,B,C,D, etc. It would usually be used to create an array:

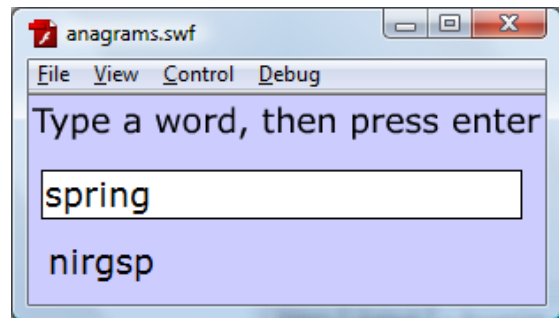
```
var a:Array=s.split("");
trace(a[2]); //displays C
```

This can be useful if we have a name as last,first:

```
var s:String="Banks,Robin";
var a:Array=s.split(",");
trace(a[0]); //displays Banks, a[1] is Robin
```

Anagrams

String functions can be used to create a number of games. An anagram is a jumble of the letters in a word. An anagram for SPRING is GRIPNS. We will write a program that lets the user type in a word and press enter to see some anagrams for the word. As you will see, we use many of the string methods in this program.



Add three text boxes to the movie:

The first is just static text with instructions.

The second is Input Text named **txtInput**. Select a border around the text

The third is Dynamic Text named **txtOutput**.

The code is shown below:

```
stage.addEventListener(KeyboardEvent.KEY_DOWN,scramble);
function scramble(e:KeyboardEvent):void{
    if(e.keyCode==13) { //do this block only when the user presses enter
        var s:String=txtInput.text; //get the word from the input box
        var letters:Array=s.split(""); //make an array of the letters
        var i, j:int;
        var temp:String; //when we swap two letters, we store one of them temporarily
        for(i=0;i<letters.length;i++) { //loop through the letters
            j=Math.random()*letters.length; //random letter
            //swap letters i and j
            temp=letters[i]; //store letter[i]temporarily so it isn't erased in the next step
            letters[i]=letters[j]; //replace letter[i] with letter[j]
            letters[j]=temp; //put former letter [i]from temp into letter[j]
        } //loop to scramble
        var s2:String="";
        for(i=0;i<letters.length;i++) { //loop to make string2 from letters in array
            s2=s2+letters[i]; //concatenate with the +
        } //loop to concatenate the scrambled letters
        txtOutput.text=s2; //display the result
    } //pressed enter
} //scramble
```

Experiment 1: If the user types “Spring” the anagram will also have a capital “S”: “gpiSrn”, for instance. You can avoid this by changing the word to upper case.

Experiment 2: When the user presses enter, show several anagrams. If you make txtOutput multiline you can create a new line with “\n”.

txtOutput.text="abc"+"\\n"+"def"; will display **abc** on one line, **def** on the next.