String Operation | String in Stack | String in Rodata Section
--- | --- | ---
**Allocating memory for a string**

```c
{ char pcStr[5]; ... }
```

```c
{ ... "hi"... ... }
```

**Initializing a string**

```c
{ char pcStr1[3] = {'h', 'i', '\0'}; char pcStr2[] = {'h', 'i', '\0'}; char pcStr3[3] = "hi"; char pcStr4[] = "hi"; char pcStr5[2] = "hi"; /* truncation */ char pcStr6[20] = "hi"; ... }
```

```c
{ ... "hi"... ... }
```

**Computing the length of a string**

```c
{ char pcStr[20] = "hi"; ... strlen(pcStr) ... /* Evaluates to 2 */ ... sizeof(pcStr) ... /* Evaluates to 20 */ }
```

```c
{ char *pcStr = "hi"; ... strlen("hi") ... ... strlen(pcStr) ... /* Evaluate to 2 */ ... sizeof(pcStr) ... /* Evaluates to 20 */ ... sizeof("hi") ... /* Evaluates to 3 */ }
```

**Changing the characters of a string**

```c
```

```c
{ Runtime error to attempt to change the characters of a string that resides in the rodata section }
```

**Concatenating characters onto a string**

```c
```

```c
{ Runtime error to attempt to change the characters of a string that resides in the rodata section }
```
### Comparing one string with another

```c
{ char pcStr1[] = "hi";
  char pcStr2[] = "bye";
  if (pcStr1 < pcStr2) ...
    /* Legal, but compares pointers!!! */
  if (strcmp(pcStr1, pcStr2) < 0) ...
    /* Compares strings */
}
```

(Same as string in stack)

### Reading a string

```c
{ char pcStr[20];
  iConvCount = scanf("%s", pcStr);
    /* Reads a word as a string.
       Grave danger of memory corruption. */
  iRet = gets(pcStr);
    /* Reads a line as a string,
       removing the \n character.
       Grave danger of memory corruption. */
  iRet = fgets(pcStr, 20, stdin);
    /* Reads a line as a string,
       retaining the \n character. */
}
```

(Runtime error to attempt to change the characters of a string that resides in the rodata section)

### Writing a string

```c
{ char pcStr[] = "hi";
  iCharCount = printf("%s", pcStr);
    /* Writes a string. */
  iSuccessful = puts(pcStr);
    /* Writes a string, appending a \n character. */
  iSuccessful = fputs(pcStr, stdout);
    /* Writes a string. */
}
```

(Same as string in stack)

### Converting a string to another type

```c
{ char pcStr[] = "123";
  int i;
  long l;
  double d;
  iConvCount = sscanf(pcStr, "%d", &i);
  i = atoi(pcStr);
  l = atol(pcStr);
  d = atof(pcStr);
}
```

(Same as string in stack)

### Converting another type to a string

```c
{ char pcStr[20];
  int i = 123;
  iCharCount = sprintf(pcStr, "%d", i);
    /* Danger of memory corruption. */
}
```

(Runtime error to attempt to change the characters of a string that resides in the rodata section)

---

Copyright © 2005 by Robert M. Dondero, Jr.