13.2 String Variables

Some programming languages provide a special `string` type for declaring string variables. C takes a different tack: any one-dimensional array of characters can be used to store a string, with the understanding that the string is terminated by a null character. This approach is simple. but has significant difficulties. It's sometimes hard to tell whether an array of characters is being used as a string. If we write our own string-handling functions, we've got to be careful that they deal properly with the null character. Also, there's no faster way to determine the length of a string than a character-by-character search for the null character.

Let's say that we need a variable capable of storing a string of up to 80 characters. Since the string will need a null character at the end, we'll declare the variable to be an array of 81 characters:

idiom
```
#define STR_LEN 80
...
char str[STR_LEN+1];
```

We defined `STR_LEN` to be `80` rather than `81`. thus emphasizing the fact that `str` can store strings of no more than 80 characters. and then added 1 to `STR_LEN` in the declaration of `str`. This a common practice among C programmers.

⚠️ When declaring an array of characters that will be used to hold a string, always make the array one character longer than the string, because of the C convention that every string is terminated by a null character. Failing to leave room for the null character may cause unpredictable results when the program is executed, since functions in the C library assume that strings are null-terminated.

Declaring a character array to have length STR_LEN + 1 doesn't mean that it will always contain a string of STR_LEN characters. The length of a string depends on the position of the terminating null character, not on the length of the array in which the string is stored. An array of STR_LEN + 1 characters can hold strings of various lengths, ranging from the empty string to strings of length STR_LEN.

Initializing a String Variable

A string variable can be initialized at the same time it's declared:

```
char date1[8] = "June 14";
```