CMSC 21
Fundamentals of Programming

2nd Semester 2011-2012

Array of Characters

STRINGS

Strings

- Array of characters or a pointer to characters
- Terminated by a special character called the null terminator `'\0'`
- Can hold a maximum number of characters equal to size-1

Declaring Strings

- Strings are declared as:

```
char var_name[size];
```

```
char name[30];
/*allocate an array of 30 characters
in the memory*/
```

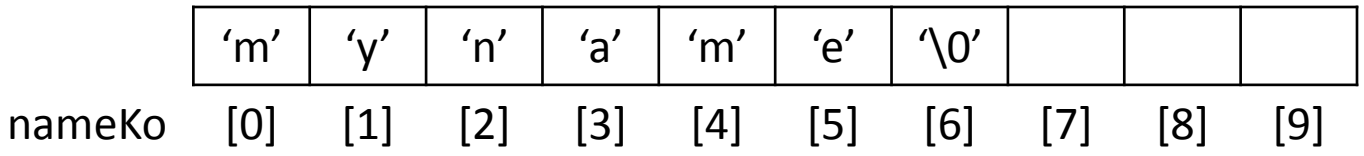
Declaring Strings

```
char name[30];
/*This string can hold up to 29
characters. The last space is
reserved for the null terminator
'\0'*/
```

Initializing Strings

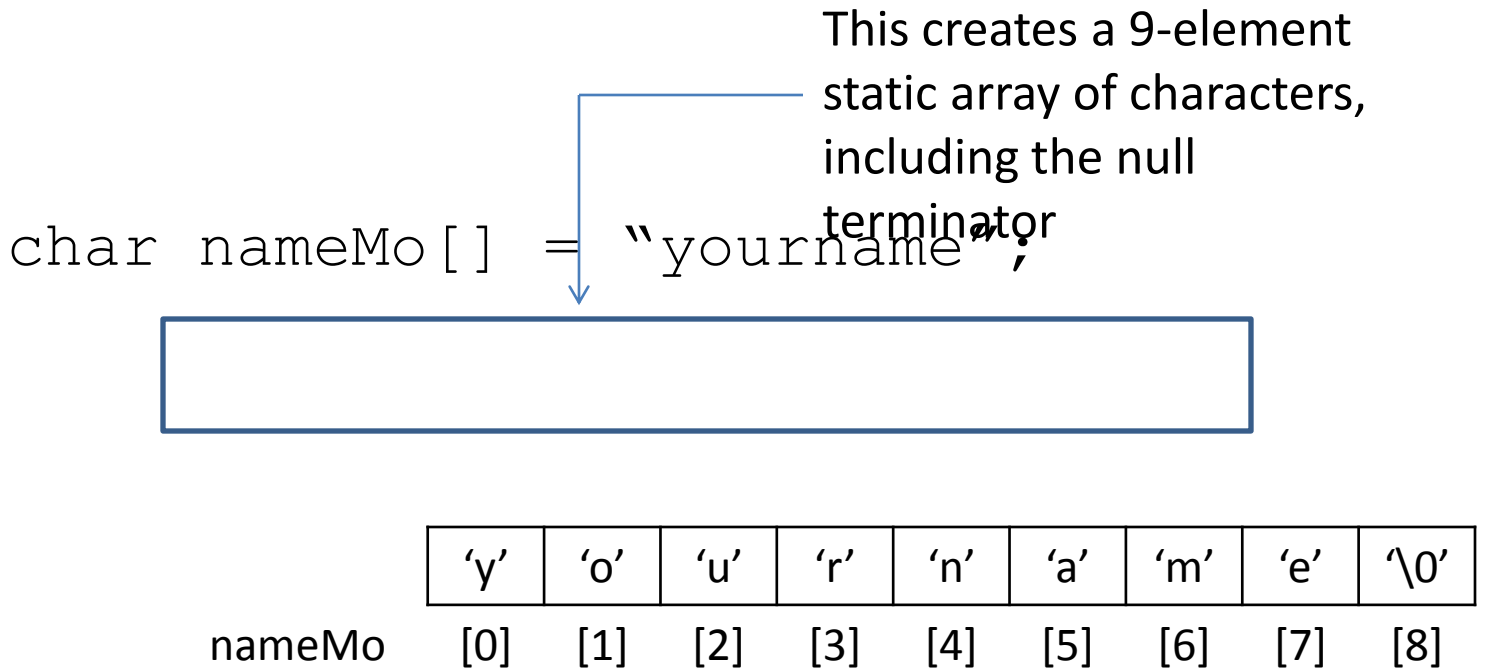
- Strings can be initialized upon declaration

```
char nameKo[10] = "myname";
```



Initializing Strings

- Strings can be initialized upon declaration



Initializing Strings

- Strings can be initialized upon declaration

This creates a 9-element static array of characters, including the null terminator

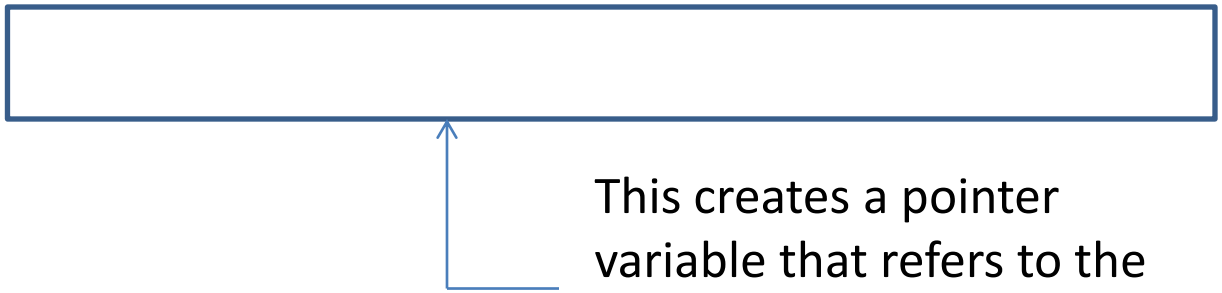
```
char nameMo1[] = {'y', 'o', 'u', 'r', 'n',
                  'a', 'm', 'e', '\0'};
```

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]
	'y'	'o'	'u'	'r'	'n'	'a'	'm'	'e'	'\0'
nameMo1									

Initializing Strings

- Strings can be initialized upon declaration

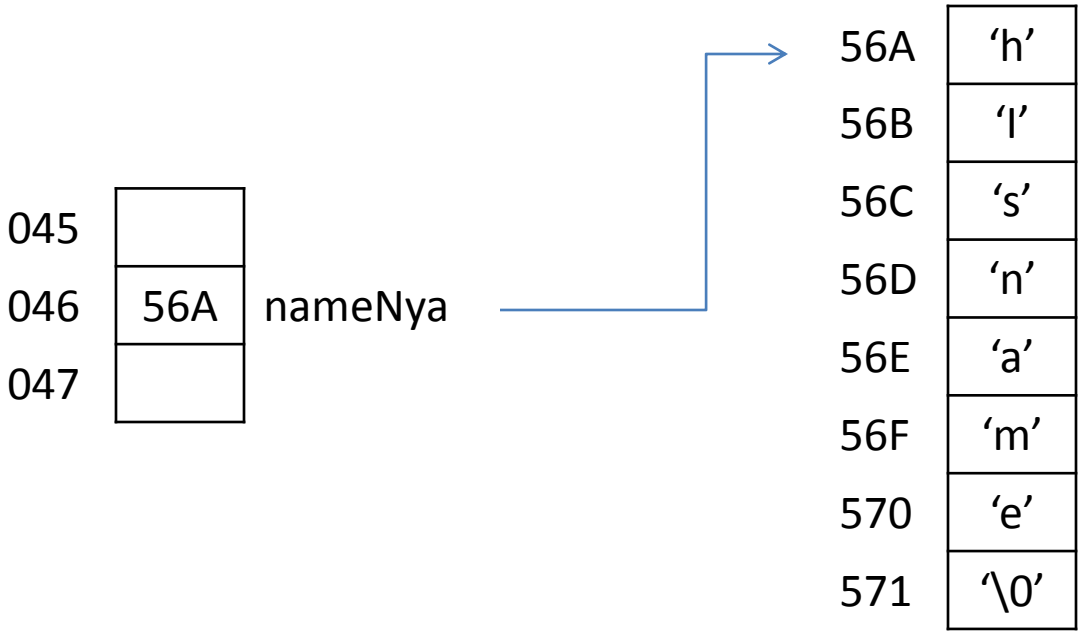
```
char *nameNya = "hisname";
```



Initializing Strings

- Strings can be initialized upon declaration

```
char *nameNya = "hisname";
```



Notes

- Given the following declarations:

```
char s[20];
char *sp;
char s1[] = “hello”;
```

Notes

```
s = “hello world”;   //not valid since s and s1 =
“hi world”;    //s1 are constant
         //pointers

strcpy (s, “hello world”); //valid
strcpy (s1, “hi world”);   //valid
```

Notes

```
sp = “hello world”;  //valid
sp = s;              //valid

strcpy (sp, “hello word”);
//not valid because sp is not a string, it is
just //a pointer to a character
```