

Intermission – Strings in C

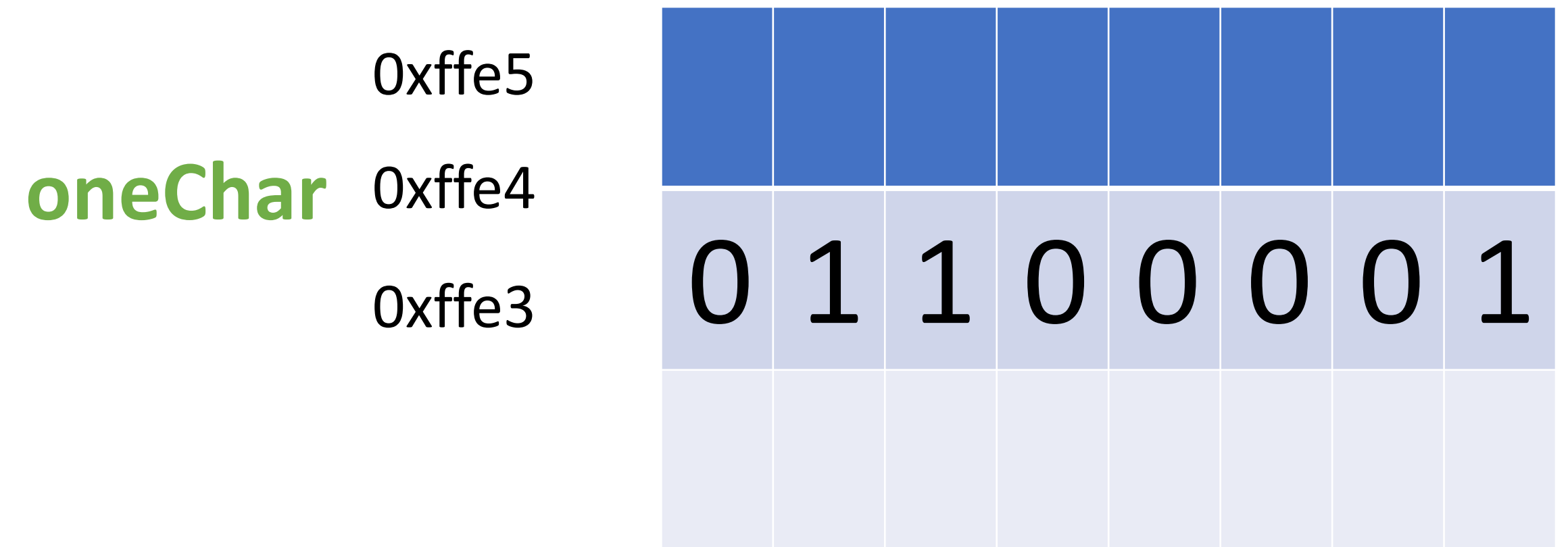
- Normally, we'd cover Strings *after* covering pointers.
- But... we want to get you up and running doing some programming in C and you need to know how C handles Strings to do almost anything in C.
- So....
 - First, a crash course on Strings, then back to our regularly scheduled lesson

Chars

- Char
basic data type (one byte)
- ASCII (UTF-8) character is delimited by **single** quotes (' ')
- Char is just a number, so you can do math on it.

```
char oneChar = 'a';
```

```
char oneChar = 0x61; // same as 'a'
```



```
oneChar = oneChar + 1; // same as 'b'  
                      // same as 0x62
```

C Strings

- C has no dedicated variable type for strings
 - Instead, a string is represented as an **array of characters** with a special ending sentinel with a value `'\0'` (zero)

	<i>index</i>	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>
"Hello"	<i>char</i>	'H'	'e'	'l'	'l'	'o'	'\0'

- `'\0'` is the **null-terminating character** (zero - do not confuse with `'0'`)
 - you always need to allocate one extra space in an array for it
 - a string does not always have `'\n'` (do not depend on `'\n'` being right before the `'\0'`)
- Strings are **not** objects
 - They do not embed additional information (e.g., string length). You must calculate this!
- You can use the C string library **`strlen`** function to calculate string length
 - The null-terminating character does *not* count towards the length.

```
int length = strlen(myStr);           // length = 5
```

Caution: `strlen` is $O(N)$ because it must scan the entire string!
You should save the value if you plan to refer to the length later.

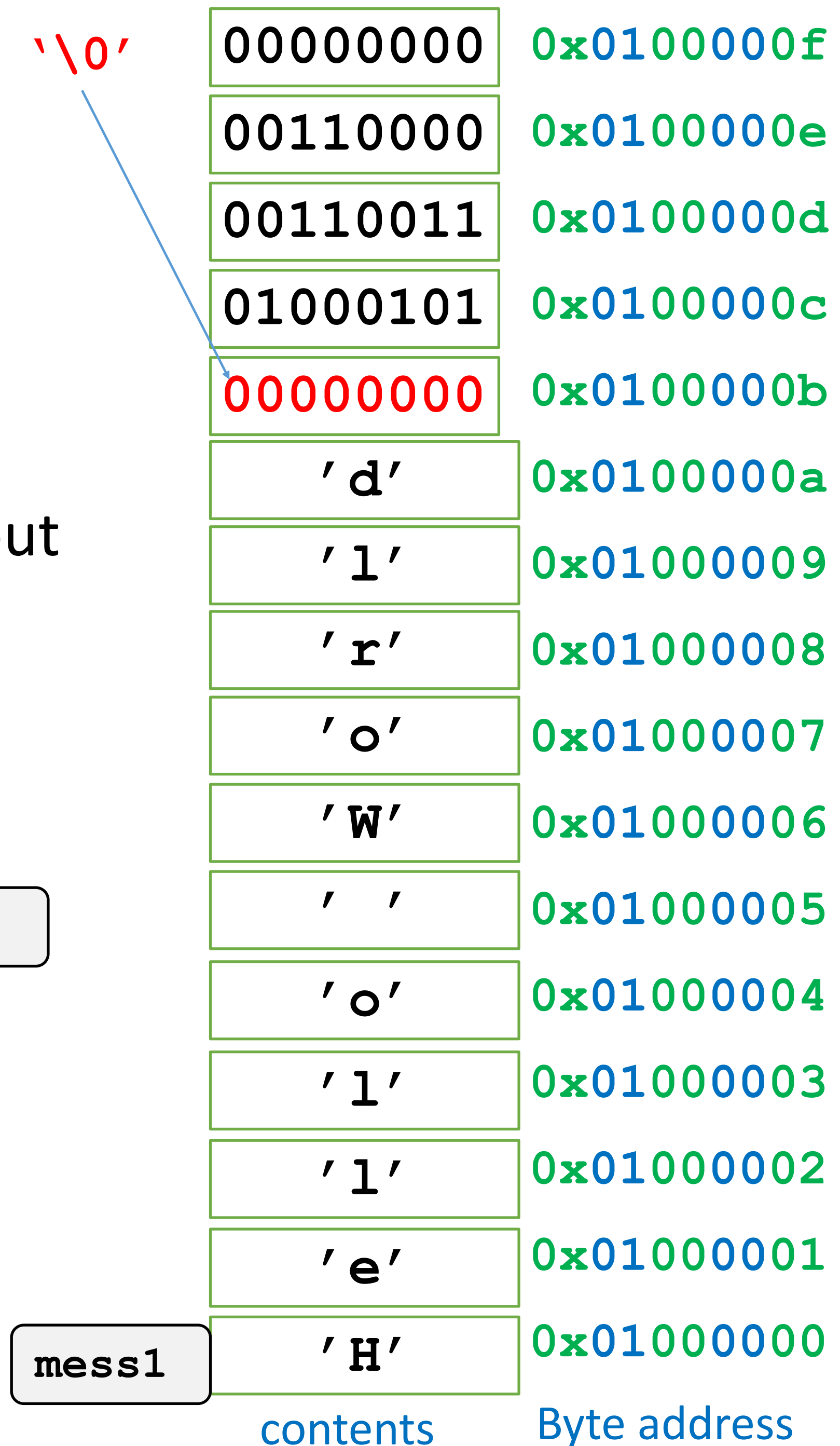
C Strings

- `mess1` is an array with enough space to hold the string + `'\0'`
 - you can change array contents but not what `mess1` points at

```
char mess1[] = "Hello World";
```

- `mess2` is an array with enough space to hold the characters but does not have space for the `'\0'` SO IT IS NOT A VALID STRING
 - Since this is NOT `'\0'` terminated, string library functions will not work properly.

```
char mess2[] = {'H','e','l','l','o',' ','W','o','r','l','d'};
```



C Standard String Library (some useful functions)

- `size_t strlen(const char *s);`
- `char *strcpy(char *s0, const char *s1)`
- `char *strncpy(char *s0, const char *s1, size_t n)`
- `char *strcat(char *s0, const char *s1);`
- `char *strncat(char *s0, const char *s1, size_t n);`
- `int strcmp(const char *s0, const char *s1);`