C-STRINGS

C-Style / Character Strings

Delroy A. Brinkerhoff

C-STRINGS

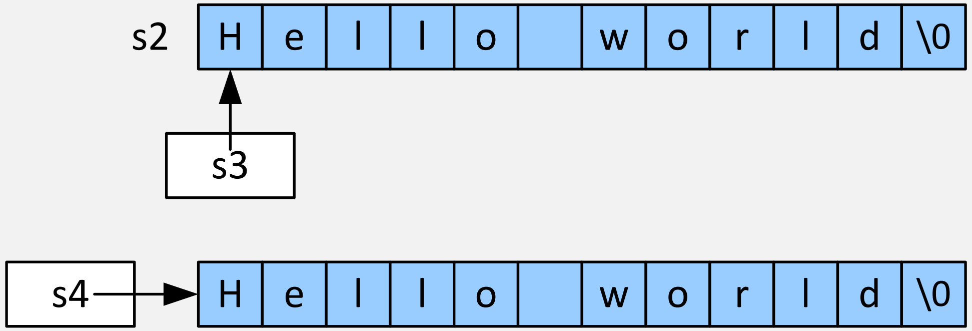
- C-Strings consist of a null-terminated character array

s1	H	e	l	l	o		w	o	r	l	d	\0
s2	H	e	l	l	o		w	o	r	l	d	\0

```
char s1[] = { 'H', 'e', 'l', 'l', 'o', ' ', 'w', 'o', 'r', 'l', 'd', '\0' };
char s2[] = "Hello world";
```

C-STRINGS, PART 2

- C-Strings are often manipulated as a character pointer

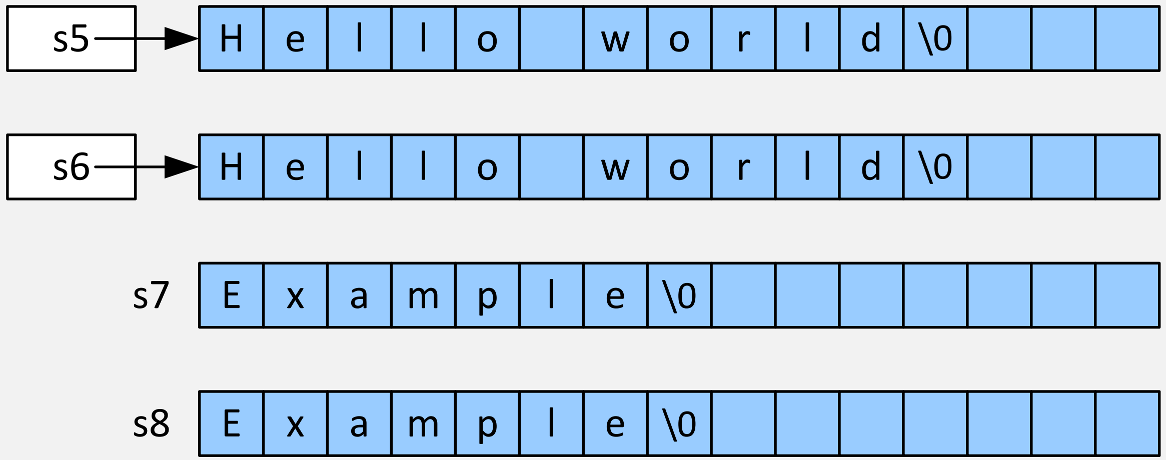


s2	H	e	l	l	o		w	o	r	l	d	\0
s4	H	e	l	l	o		w	o	r	l	d	\0

```
char s2[] = "Hello world";
char* s3 = s2;
const char* s4 = "Hello world";
```

C-STRINGS, PART 3

- C-Strings may be shorter than the storing array



s5	H	e	l	l	o		w	o	r	l	d	\0			
s6	H	e	l	l	o		w	o	r	l	d	\0			
s7	E	x	a	m	p	l	e	\0							
s8	E	x	a	m	p	l	e	\0							

```
char* s5 = new char[15] { 'H', 'e', 'l', 'l', 'o', ' ', 'w', 'o', 'r', 'l', 'd’ };
char* s6 = new char[15] { "Hello world" };
char s7[15] = { 'E', 'x', 'a', 'm', 'p', 'l', 'e' };
char s8[15] = { "Example" };
```

UNINITIALIZED CHARACTER ARRAYS

- Possible to create a character array without initializing it
  - Not a C-string yet: no null-termination character
  - Most C-string functions will fail
  - `char s9[100];`
  - `char* s10 = new char[100];`