String

Md. Saidul Hoque Anik

String

Basically a character array

```
char str[6];
```

String

In place initialization

```
char str[6] = {'H', 'e', 'l', 'l', 'o', '\0'};
```


Null character

String

In place initialization

```
char str[6] = {'H', 'e', 'l', 'l', 'o', '\0'};
```

Null character

We can also write

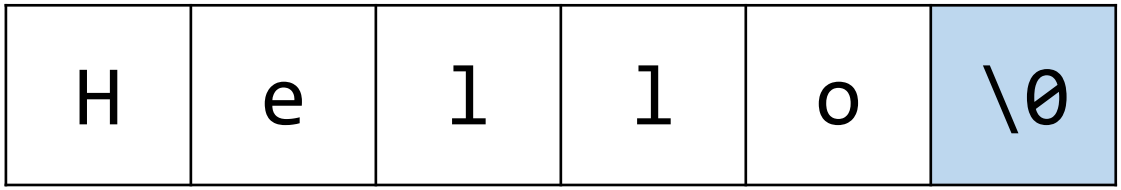
```
char str[6] = "Hello";
```

str:

H	e	l	l	o	\0

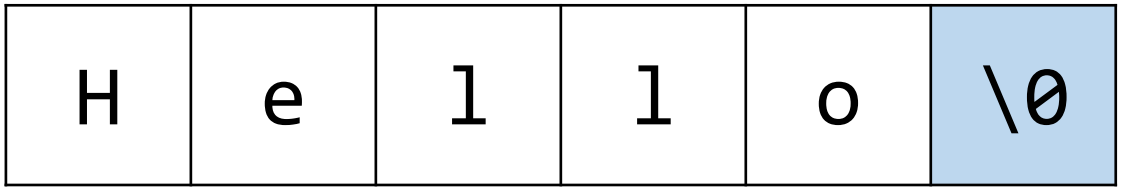
String

Why null character?



String

Why null character?



To mark the end of string

```
char str[6] = "Hello";
```

[C compiler automatically adds null character here]

String

Another more useful way of initialization (without length)

```
char str[] = "Hello";
```

[C compiler automatically assigns required size]

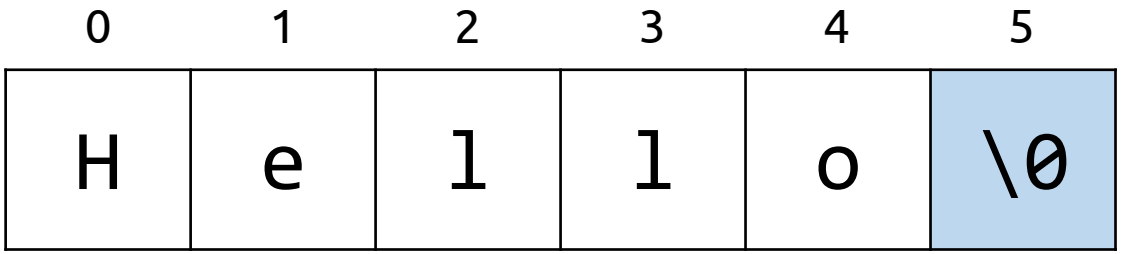
Accessing char by char

```c
#include <stdio.h>

int main()
{
    char str[] = "Hello";

    printf("%c", str[0]); //H
    printf("%c", str[2]); //l
    printf("%c", str[6]); //what will it show?

}
```



Reading word from user

```c
#include <stdio.h>

int main()
{
    char name[20];

    scanf("%s", &name);
    printf("Hello %s", name);

}
```

Reading word from user

```c
#include <stdio.h>

int main()
{
    char name[20];

    scanf("%s", &name);
    printf("Hello %s", name);

}
```

scanf terminates when a whitespace is found.

So we cannot use %s for reading a line

Reading/writing line

Usage of gets() and puts()

```c
#include <stdio.h>

int main()
{
    char name[20];

    gets(name);
    printf("Hello ", name);
    puts(name);

}
```

Finding the length of the string

Is it 20?

```
char name[20] = "Hello world";
printf("%d", sizeof name);   //20
```

Finding the length of the string

How do we know the end of string?

```c
#include <stdio.h>

int main()
{
    char name[20] = "Hello world";









}
```

Finding the length of the string

How do we know the end of string?

```c
#include <stdio.h>

int main()
{
    char name[20] = "Hello world";
    int len = 0;
    int i;









}
```

Finding the length of the string

Why 20?

```c
#include <stdio.h>

int main()
{
    char name[20] = "Hello world";
    int len = 0;
    int i;
    for (i = 0; i<20; i++)
    {




    }

}
```

Finding the length of the string

What will happen if null char is found? Else?

```c
#include <stdio.h>

int main()
{
    char name[20] = "Hello world";
    int len = 0;
    int i;
    for (i = 0; i<20; i++)
    {
        if (name[i] == '\0')

        else

    }

}
```

Finding the length of the string

```c
#include <stdio.h>

int main()
{
    char name[20] = "Hello world";
    int len = 0;
    int i;
    for (i = 0; i<20; i++)
    {
        if (name[i] == '\0')
            break;
        else
            len++;
    }

}
```

Finding the length of the string

```c
#include <stdio.h>

int main()
{
    char name[20] = "Hello world";
    int len = 0;
    int i;
    for (i = 0; i<20; i++)
    {
        if (name[i] == '\0')
            break;
        else
            len++;
    }
    printf("Length of %s is : %d", name, len);
}
```

Task 0: Rewrite the following code using while loop

```c
#include <stdio.h>

int main()
{
    char name[20] = "Hello world";
    int len = 0;
    int i;
    for (i = 0; i<20; i++)
    {
        if (name[i] == '\0')
            break;
        else
            len++;
    }
    printf("Length of %s is : %d", name, len);
}
```

Finding the length of the string

We can also use the library function

```c
#include <stdio.h>
#include <string.h>

int main()
{
    char name[20] = "Hello world";

    int len = strlen(name);

    printf("Length of %s is : %d", name, len);
}
```

Task 1: Search a character in a string

```c
#include <stdio.h>
#include <string.h>

int main()
{
    char str[20];  //take input from user
    char c;          //take input from user

    //print Found, if c is in str
    //print Not found otherwise

}
```

Task 2: Copying one String to another

```c
#include <stdio.h>

int main()
{
    char source[20] = "Hello World";

    char destination[20];
    //copy source to destination
    puts(destination);  //Hello World

}
```

Task 2: Copying one String to another

We can also use the library function

```c
#include <stdio.h>
#include <string.h>

int main()
{
    char source[20] = "Hello World";
    char destination[20];
    strcpy(destination, source);
    puts(destination);
}
```

Task 3: Concatenation (joining)

```c
#include <stdio.h>
#include <string.h>

int main()
{
    char str[20] = "Hello";
    char str2[6] = "World";

    //Write code to append str2 in str

    puts(str);  //HelloWorld
}
```

Task 3: Concatenation (joining)

We can also use the library function

```c
#include <stdio.h>
#include <string.h>

int main()
{
    char str[20] = "Hello";
    char str2[6] = "World";

    strcat(str, str2);

    puts(str);  //HelloWorld

}
```

Task 3: Concatenation (joining)

We can also use the library function

```c
#include <stdio.h>
#include <string.h>

int main()
{
    char str[20] = "Hello";
    char str2[6] = "World";

    strcat(str, str2);

    puts(str);  //HelloWorld
    puts(str2); //What will be the output?
}
```

Task 4: Reversing a string

```c
#include <stdio.h>
#include <string.h>

int main()
{
    char str[20] = "ABCDE";
    int len = 5;

    //Write code to reverse the chars of str

    puts(str);  //EDCBA
}
```

Task 4: Reversing a string

We can also use the library function

```c
#include <stdio.h>
#include <string.h>

int main()
{
    char str[20] = "ABCDE";
    int len = 5;

    strrev(str);

    puts(str);  //EDCBA
}
```

Task 5: Checking is two strings are equal

```c
#include <stdio.h>
#include <string.h>

int main()
{
    char str1[20] = "ABCDE";
    int len1 = 5;
    char str2[20] = "Hello";
    int len2 = 5;

    //Write code to check if str1 and str2 are equal

}
```

Task 5: Checking is two strings are equal

As usual, we can also use the library function

```c
#include <stdio.h>
#include <string.h>

int main()
{
    char str1[20] = "ABCDE";
    int len1 = 5;
    char str2[20] = "ABCDE";
    int len2 = 5;

    int c = strcmp(str1, str2);
    if (c == 0)
        printf("They are equal");
    else
        printf("They are not equal");
}
```

Task 5: Checking is two strings are equal

What will strcmp return if they are not equal?

if Return value < 0 then it indicates str1 is less than str2.

if Return value > 0 then it indicates str2 is less than str1.

if Return value = 0 then it indicates str1 is equal to str2.

Task 5: Checking is two strings are equal

What will strcmp return if they are not equal?

if Return value < 0 then it indicates str1 is less than str2.

if Return value > 0 then it indicates str2 is less than str1.

if Return value = 0 then it indicates str1 is equal to str2.

How can a string be less than another?

Task 5: Checking is two strings are equal

What will strcmp return if they are not equal?

if Return value < 0 then it indicates str1 is less than str2.

if Return value > 0 then it indicates str2 is less than str1.

if Return value = 0 then it indicates str1 is equal to str2.

How can a string be less than another?

A string is smaller than another, if it appears before the other in **lexicographical order**.