

C String

C String

► #include<string.h>

"ABCDEFGHI"									
'A'	'B'	'C'	'D'	'E'	'F'	'G'	'H'	'I'	'\0'
65	66	67	68	69	70	71	72	73	0

fgets(str, len, stdin)

- ▶ Read from stdin until a new line is found, at end-of-file or error.
 - ▶ '\n' is included
 - ▶ return NULL if error or nothing read before end-of-file reached.

fgets(str, len, stdin)

Input: "MM FF FM"									
'M'	'M'	' '	'F'	'F'	' '	'F'	'M'	'\n'	'\0'
77	77	32	70	70	32	70	77	10	0

strlen

► #include<string.h>

strlen("ABCDEFGHI") returns 9									
'A'	'B'	'C'	'D'	'E'	'F'	'G'	'H'	'I'	'\0'
65	66	67	68	69	70	71	72	73	0

strlen

```
size_t  
strlen(str)  
    const char *str;  
{  
    register const char *s;  
  
    for (s = str; *s; ++s);  
    return(s - str);  
}
```

Note: source code from Apple

strcpy

```
char *  
strcpy(char *s1, const char *s2)  
{  
    char *s = s1;  
    while ((*s++ = *s2++) != 0)  
        ;  
    return (s1);  
}
```

Note: source code from Apple

strcmp

```
int
strcmp(const char *s1, const char *s2)
{
    for ( ; *s1 == *s2; s1++, s2++)
        if (*s1 == '\0')
            return 0;
    return ((*s1 < *s2) ? -1 : +1);
}
```

Note: source code from Apple

strtok

- ▶ `strtok(str, delim)` returns the first token
 - ▶ `strcpy(p, "__ooxx_XXoo_");`
`strtok(p, "_")` returns `p+2` & set `p[6]=0`
- ▶ `strtok(NULL, delim)` returns the next token
 - ▶ `strcpy(p, "__ooxx_XXoo_");`
`strtok(p, "_")` returns `p+2`
`strtok(NULL, "_")` returns `p+7`
`strtok(NULL, "_")` returns `NULL`
- ▶ `strtok` is **destructive**

"ooxx"
"XXoo"

strtok

```
strcpy(p, "__AA__B__");
```

'_'	'_'	'A'	'A'	'_'	'_'	'B'	'_'	'_'	'\0'
95	95	65	65	95	95	66	95	95	0

strtok

strtok(p, "_");									
'_'	'_'	'A'	'A'	'\0'	'_'	'B'	'_'	'_'	'\0'
95	95	65	65	0	95	66	95	95	0

strtok

strtok(NULL, "_");									
'_'	'_'	'A'	'A'	'\0'	'_'	'B'	'\0'	'_'	'\0'
95	95	65	65	0	95	66	0	95	0

strtok

strtok(NULL, "_");									
'_'	'_'	'A'	'A'	'\0'	'_'	'B'	'\0'	'_'	'\0'
95	95	65	65	0	95	66	0	95	0

```

char *strtok(s, delim)
    register char *s;
    register const char *delim;
{
    register char *spanp;
    register int c, sc;
    char *tok;
    static char *last;
    if (s == NULL && (s = last) == NULL)
        return (NULL);
cont:
    c = *s++;
    for (spanp = (char *)delim; (sc = *spanp++) != 0;) {
        if (c == sc)
            goto cont;
    }

    if (c == 0) { /* no non-delimiter characters */
        last = NULL;
        return (NULL);
    }
    tok = s - 1;
    for (;;) {
        c = *s++;
        spanp = (char *)delim;
        do {
            if ((sc = *spanp++) == c) {
                if (c == 0)
                    s = NULL;
                else
                    s[-1] = 0;
                last = s;
                return (tok);
            }
        } while (sc != 0);
    }
}

```

Note: source code from Apple