

Programlamaya Giriş

Dizeler

Hüseyin Ahmetođlu

Strings

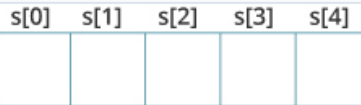
- ▶ C programlamasında, bir dizi, null karakter \0 ile sonlandırılmış bir karakter dizisidir.
- ▶ Derleyici, çift tırnak işareti içine alınmış bir dizi karakterle karşılaştığında, varsayılan olarak sonuna bir \0 karakteri ekler.

```
char c[] = "c string";
```

c		s	t	r	i	n	g	\0
---	--	---	---	---	---	---	---	----

Bir dizge nasıl bildirilir?

```
char s[5];
```



Dizeler nasıl başlatılır?

```
char c[] = "abcd";
```

```
char c[50] = "abcd";
```

```
char c[] = {'a', 'b', 'c', 'd', '\0'};
```

```
char c[5] = {'a', 'b', 'c', 'd', '\0'};
```

c[0]	c[1]	c[2]	c[3]	c[4]
a	b	c	d	\0

Dizelere Değer Atama

```
char c[100];  
c = "C programming"; // Error! array type is not assignable.
```

Kullanıcıdan String alma

- ▶ Bir dizeyi okumak için scanf () fonksiyonunu kullanabilirsiniz.
- ▶ Scanf () fonksiyonu, boşlukla (boşluk, satırsonu, sekme, vb.) karşılaşılan kadar karakter dizisini okur.

```
#include <stdio.h>
int main()
{
    char name[20];
    printf("Enter name: ");
    scanf("%s", name);
    printf("Your name is %s.", name);
    return 0;
}
```

Output

```
Enter name: Dennis Ritchie
Your name is Dennis.
```

Bir metin satırı nasıl okunur?

- ▶ Dennis Ritchie yukarıdaki programa girilmiş olsa da, ad dizisinde sadece "Dennis" saklanıyordu. Dennis'den sonra boşluk olduğu için.
- ▶ Bir dizge satırını okumak için `fgets ()` fonksiyonunu kullanabilirsiniz. Ve dizeyi görüntülemek için `puts ()` kullanabilirsiniz.

```
#include <stdio.h>
int main()
{
    char name[30];
    printf("Enter name: ");
    fgets(name, sizeof(name), stdin); // read string
    printf("Name: ");
    puts(name); // display string
    return 0;
}
```

Output

```
Enter name: Tom Hanks
Name: Tom Hanks
```

Dizeleri Fonksiyonlara parametre olarak atama

```
#include <stdio.h>
void displayString(char str[]);

int main()
{
    char str[50];
    printf("Enter string: ");
    fgets(str, sizeof(str), stdin);
    displayString(str);    // Passing string to a function.
    return 0;
}
void displayString(char str[])
{
    printf("String Output: ");
    puts(str);
}
```


String Fonksiyonları - strlen()

```
#include <stdio.h>
#include <string.h>
int main()
{
    char a[20]="Program";
    char b[20]={'P','r','o','g','r','a','m','\0'};

    // using the %zu format specifier to print size_t
    printf("Length of string a = %zu \n",strlen(a));
    printf("Length of string b = %zu \n",strlen(b));

    return 0;
}
```

Output

```
Length of string a = 7
Length of string b = 7
```

strcpy()

```
char* strcpy(char* destination, const char* source);
```

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

int main() {
    char str1[20] = "C programming";
    char str2[20];

    // copying str1 to str2
    strcpy(str2, str1);

    puts(str2); // C programming

    return 0;
}
```

Output

```
C programming
```

strcat()

```
char *strcat(char *destination, const char *source)
```

```
#include <stdio.h>
#include <string.h>
int main() {
    char str1[100] = "This is ", str2[] = "programiz.com";

    // concatenates str1 and str2
    // the resultant string is stored in str1.
    strcat(str1, str2);

    puts(str1);
    puts(str2);

    return 0;
}
```

Output

```
This is programiz.com
programiz.com
```

strcmp()

```
int strcmp (const char* str1, const char* str2);
```

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

int main()
{
    char str1[] = "abcd", str2[] = "abCd", str3[] = "abcd";
    int result;

    // comparing strings str1 and str2
    result = strcmp(str1, str2);
    printf("strcmp(str1, str2) = %d\n", result);

    // comparing strings str1 and str3
    result = strcmp(str1, str3);
    printf("strcmp(str1, str3) = %d\n", result);

    return 0;
}
```

Output

```
strcmp(str1, str2) = 32
strcmp(str1, str3) = 0
```

Return Value

0

negative

positive
integer

<ctype.h>

Functions	Description
isalpha()	checks whether character is alphabetic
isdigit()	checks whether character is digit
isalnum()	Checks whether character is alphanumeric
isspace()	Checks whether character is space
islower()	Checks whether character is lower case
isupper()	Checks whether character is upper case
isxdigit()	Checks whether character is hexadecimal
iscntrl()	Checks whether character is a control character
isprint()	Checks whether character is a printable character
ispunct()	Checks whether character is a punctuation
isgraph()	Checks whether character is a graphical character
tolower()	Checks whether character is alphabetic & converts to lower case
toupper()	Checks whether character is alphabetic & converts to upper case

```

1  #include <stdio.h>
2  int main() {
3      char line[150];
4      int vowels, consonant, digit, space;
5
6      vowels = consonant = digit = space = 0;
7
8      printf("Enter a line of string: ");
9      fgets(line, sizeof(line), stdin);
10
11     for (int i = 0; line[i] != '\0'; ++i) {
12         if (line[i] == 'a' || line[i] == 'e' || line[i] == 'i' ||
13             line[i] == 'o' || line[i] == 'u' || line[i] == 'A' ||
14             line[i] == 'E' || line[i] == 'I' || line[i] == 'O' ||
15             line[i] == 'U') {
16             ++vowels;
17         } else if ((line[i] >= 'a' && line[i] <= 'z') || (line[i] >= 'A' && line[i] <= 'Z')) {
18             ++consonant;
19         } else if (line[i] >= '0' && line[i] <= '9') {
20             ++digit;
21         } else if (line[i] == ' ') {
22             ++space;
23         }
24     }
25
26     printf("Vowels: %d", vowels);
27     printf("\nConsonants: %d", consonant);
28     printf("\nDigits: %d", digit);
29     printf("\nWhite spaces: %d", space);
30     return 0;
31 }

```

```

Enter a line of string: adfslkj34 34lkj343 34lk
Vowels: 1
Consonants: 11
Digits: 9
White spaces: 2

```

KAYNAKLAR

- ▶ Goel, A., & Mittal, A. (2016). Computer Fundamentals and Programming in C (RMK). Pearson Education India. Retrieved from <https://www.oreilly.com/library/view/computer-fundamentals-and/9789332579200>
- ▶ yet another insignificant Programming Notes. (2021, April 08). Retrieved from <https://www3.ntu.edu.sg/home/ehchua/programming/index.html#Cpp>