

int array
int arr[5];

Syntax:- char arr_name[size]; ⇒ char array

Example:- create a char array to store 5 characters.

```
char arr[5]; // Definition
// size 5 bytes, each char takes 1 byte
```

How to initialize char array?

```
1. char arr[5] = { 'a', 'b', 'c', 'd', 'e' }; // Initialization of Array.
```

^{5th} ^{1st} ^{2nd} ^{3rd} ^{4th}
'a' 'b' 'c' 'd' 'e'
 1 byte

How to print char array?

manually print each element

```
printf("%c", arr[0]); ⇒ a
printf("%c", arr[1]); ⇒ b
...
printf("%c", arr[4]); ⇒ e
```

Use for loop to print it
(for, while, do-while)

printing array using for loop

```
for (int i=0; i<5; i++) 0 1 2 3 4
    printf("%c", arr[i]);
```

$i=0; 0 < 5 \text{ T } arr[0] \Rightarrow a$
 $i=1; 1 < 5 \text{ T } arr[1] \Rightarrow b$
 \vdots
 $i=5; 5 < 5 \text{ False}$

if I don't initialize char array and try to print it?

```
# char arr[5];
```

| | | | |
 By default it has some garbage value.

```
for (int i=0; i<5; i++)
    printf("%c", arr[i]);
```

1. Garbage value? or null characters
2. crash?

char arr[2] = { 'a', 'b' } OR char arr[2] = { 'a', 'b' };

↑
 compiler will calculate size for us.
 It's good practice to keep it empty when initializing arr
 char arr[] = { 'a', 'b' }
 char arr[]; ⇒ Error (How much memory it will take)

Q. What if we try to print 5 characters if char array has only 2 characters.

```
char arr[2] = { 'a', 'b' };
```

⁰ ¹
'a' 'b' | | |

```
for (int i=0; i; i<5; i++) 0 - 4
```

```

char arr[2] = {'a', 'b'};
for (int i=0; i; i<5; i++)
{
    printf("%c", arr[i]);
}

```

0 - 4

arr[0] → a
arr[1] → b
arr[2] : ?
arr[3] : ?
arr[4] : ?

Undefined behavior
↳ crash
↳ null characters
↳ garbage value

"this is not good practice"

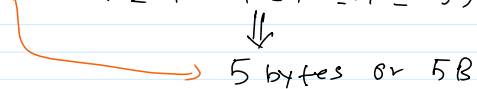
How much memory char array consumes/takes??

'a' → 1 byte

```

char arr[5] = {'a', 'b', 'c', 'd', 'e'}; // 5 elements or 5 characters

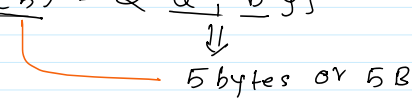
```



```

char arr[5] = {'a', 'b'};

```



##

```

char arr[5] = {}; // empty array.

```

```

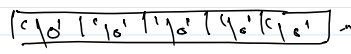
char arr[5] = {0}; // memory - 5 byte

```

```

char arr[5] = {'\0'};

```



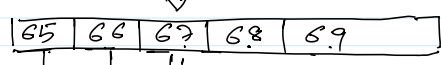
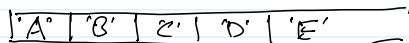
1. Garbage
2. null character.

What it will print? ⇒ Blank because null char is not printable.

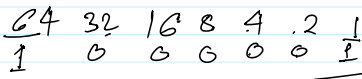
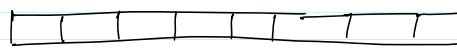
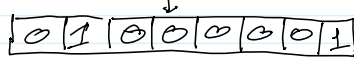
```

char arr[5] = {'A', 'B', 'C', 'D', 'E'};

```



compiler will replace characters by their ASCII value.



compiler will convert ASCII values to binary no system.