Pointers, Arrays & Strings

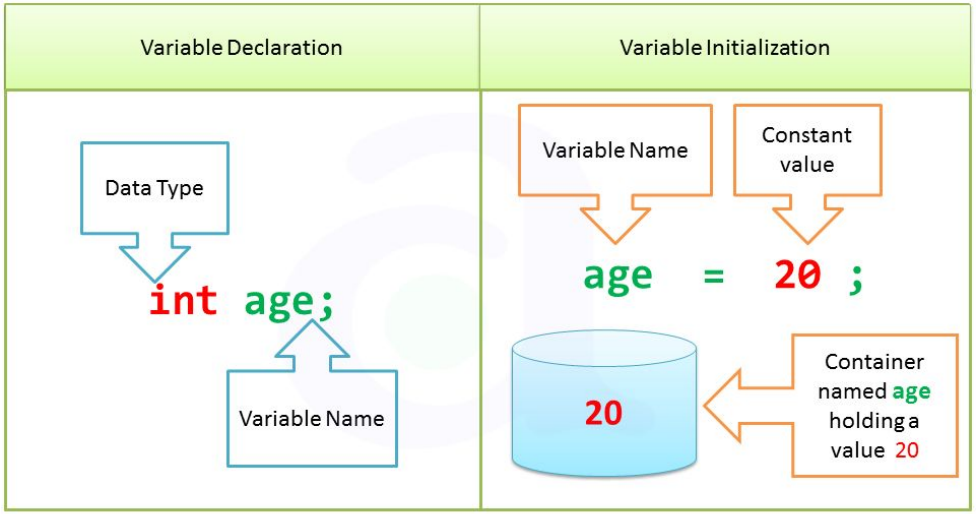
Agenda

Pointers	20
Arrays	20
Strings	20

Pointers

Declaring?? Initializing??

Declaration tells the compiler about the existence of an entity in the program and its location.
Initialization is the process of assigning a value to the Variable.
Every programming language has its own method of initializing the variable.



Intro to C pointers

- Pointers in C are easy and fun to learn.
- Some C programming tasks are performed more easily with pointers. So it becomes necessary to learn pointers to become a perfect C programmer. Let's start learning them in simple and easy steps.
- As you know, every **variable** has a memory location and **every memory location** has its address defined which can be accessed using ampersand (**&**) operator, which denotes an address in memory.

What are pointers?

- A pointer is a variable whose value is the address of another variable, i.e., direct address of the memory location.
- Like any variable or constant, you **must** declare a pointer before you can use it to store any variable address. The general form of a pointer variable declaration is:

```
type *var_name;
```

- Here, `type` is the pointer's base type; it must be a valid C data type and `var_name` is the name of the pointer variable. The asterisk ***** you used to declare a pointer is the same asterisk that you use for multiplication.

```
int  *ip;          /* pointer to an integer */
double *dp;        /* pointer to a double */
float  *fp;        /* pointer to a float */
char  *ch ;        /* pointer to a character */
```

Memory addresses in C

- Whenever a variable is defined then you can access the memory address of that variable.
- Suppose, you define a variable named **var** then if you use **&var**, it will give the variable var's address in memory.
- In C, you can get the memory address of a variable using **&** symbol.

Pointers in C

- The main purpose of a pointer is to get the memory address of the variable which is defined in the program code.
- Pointers are special variables that holds the memory address of another variable of the same data type.
- Without the help of a pointer, you cannot perform tasks such as dynamic memory allocation and many tasks in the C programming language.

How to use pointers in C

- First, you should define a pointer variable.
- Then, assign the address of a variable to that pointer using **&** symbol. It will return the address of that variable.
- You can access the value stored in that address by using *(asterisk) symbol.

We associate data type to a pointer because it knows how much bytes of data can it store.

DATA TYPE	SIZE (IN BYTE)
char	1
short int	2
int	2
long int	4
float	4
double	8
long double	10
void	MEANING LESS
enum	2

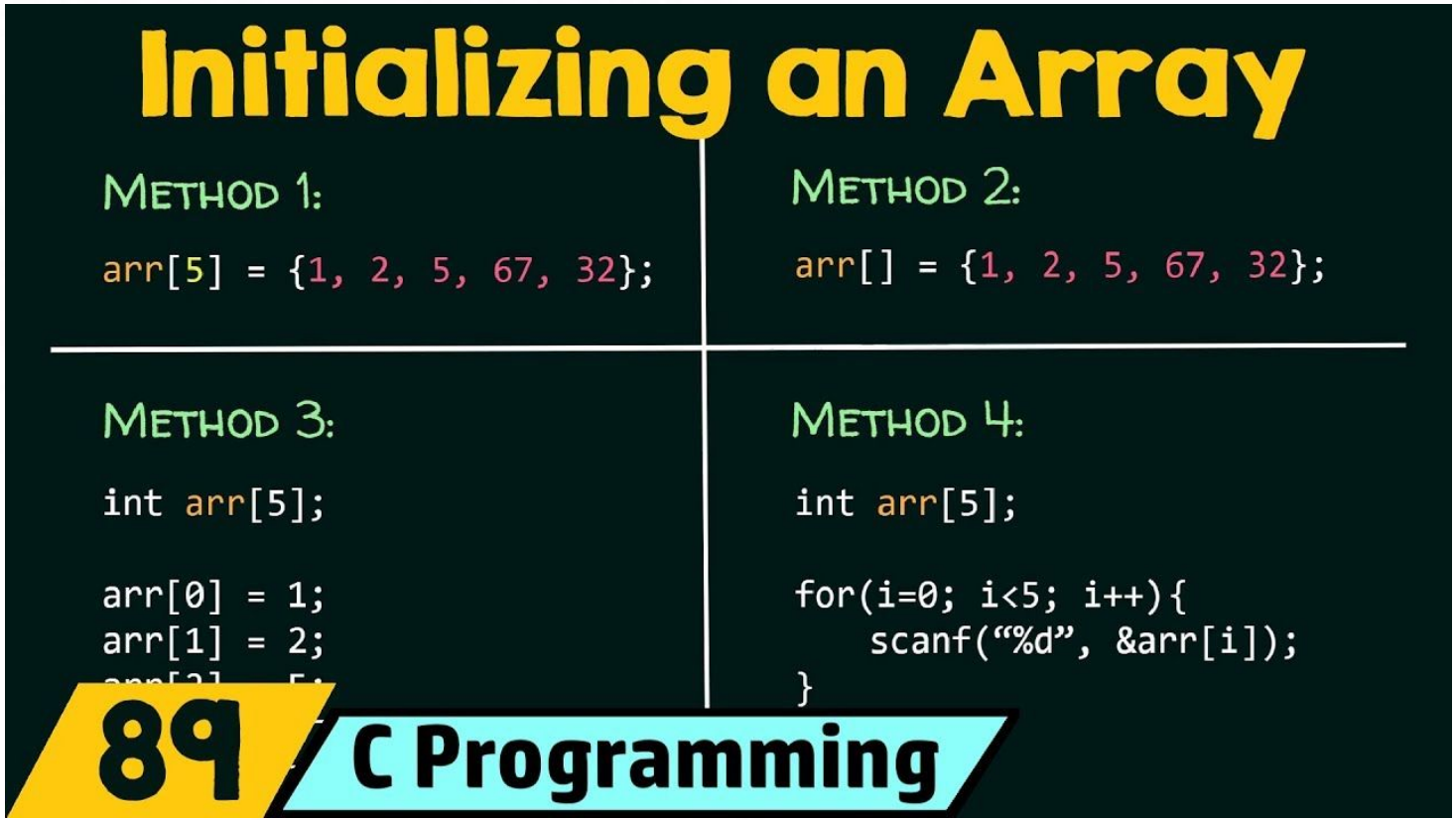
Arrays

Intro to arrays in C

- Arrays is a data structure used to store multiple values in a single variable, instead of declaring separate variables for each value.
- To create an array, define the data type and specify the name of the array followed by square brackets **[ ]**.

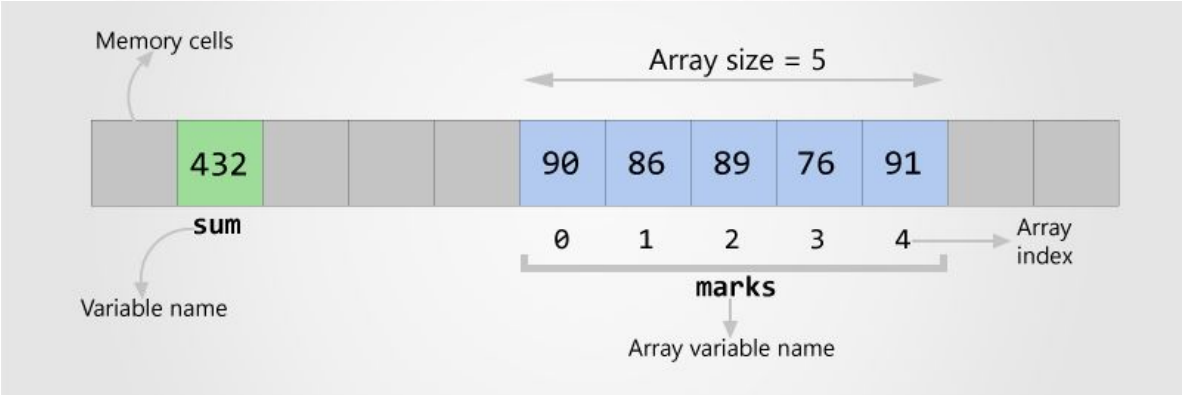
```
int myNumbers[] = {25, 50, 75, 100};
```

Ways to initialize an array in C



Accessing elements of an array

- To access an array element, refer to its index number.
- Array indexes start with 0: **[0]** is the first element. **[1]** is the second element, etc.



Changing an array element

- To change the value of a specific element, refer to the index number

Looping through an array

- You can loop through the array elements with the **for** loop.

Strings

Introduction to strings

- In C programming, a string is a sequence of characters terminated with a null character **\0**

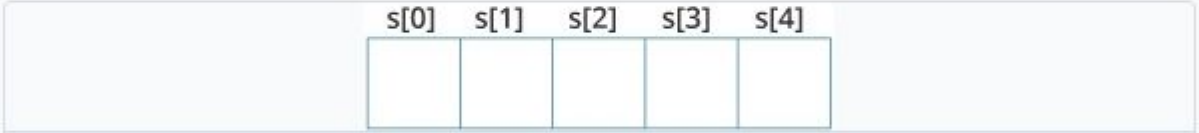
```
char c[] = "c string";
```

- When the compiler encounters a sequence of characters enclosed in the double quotation marks, it appends a null character **\0** at the end by default.

Declaring strings

- Declaring a string is as simple as declaring a one-dimensional array.

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



String Declaration in C

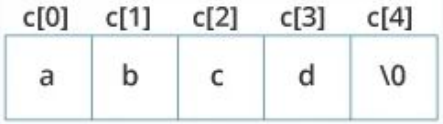
Initializing strings

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

char c[50] = "abcd";

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

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



String Initialization in C

Assigning Values to Strings

- Arrays and strings are second-class citizens in C; they do not support the assignment operator once it is declared.

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

Reading strings from stdin

- You can use the **scanf()** function to read a string.
- The **scanf()** function reads the sequence of characters until it encounters whitespace (space, newline, tab, etc.).

Resources

1. Pointers presentation
2. More on pointers
3. About arrays
4. More on initializing arrays
5. pointers
6.

See you at the next session!