

## typedef

A **typedef** is a keyword that is used to assign alternative names to existing datatypes. We use typedef with user defined datatypes, when names of the datatypes become slightly complicated to use in programs. Typedefs can be used to:

- Provide the clarity in the code
- it makes easier to change the underlying data types that you use
- Typedefs makes the code more clear and easier to modify.

Following is the syntax for using typedef,

```
typedef <previous_name> <alias_name>
```

In the above syntax, '**previous\_name**' is the name of an already existing variable while '**alias\_name**' is another name given to the existing variable.

For example, suppose we want to create a variable of type **unsigned long**, then it becomes a time taking task if we want to declare multiple variables of this type. To overcome this problem, we use a **typedef** keyword.

```
typedef unsigned long ul;
```

The above example defines a term ul for an unsigned long datatype. Now this ul identifier can be used to define unsigned long type variables.

```
ul a, b, c;
```

Let's see an example.

```
#include <stdio.h>
int main() {
    typedef unsigned long ul;
    ul a = 5, b = 8;
    printf("a = %d\n", a);
    printf("b = %d\n", b);
    return 0;
```

There are **various applications of typedef**. The following are the applications of the typedef.

- Typedef can be used with an array mostly with multi-dimensional array. It will increase the readability.

- As we know, the typedef can be implemented for defining a user-defined data type with a specific name and type. We can also use a typedef with structures of C language.

```
typedef struct
{
    structure element1;
    structure element2;
    structure element3;
} name_of_type;
```

Here name\_of\_type can be implemented by declaring a variable of this structure type.

```
name_of_type type1, type2;
```

- typedef can be used for providing a pseudo name to pointer variables as well.

```
typedef int* ptr
ptr a, b, c;
```

## Advantages of typedef

- Typedef increases the readability of the code. If we are using structure and function pointer in our code, it will increase the readability of code.
- With the help of typedef, we can use the same name for the different types in different scopes.
- In the case of structure, if we use the typedef then we do not require to write struct keyword at the time of variable declaration.
- Typedef increases the portability of the code.