

TYPDEF

The typedef keyword allows us to create a new data type name from an existing data type. It does not create a new data type but introduces a new name for existing type. A typedef declaration does not reserve storage. We cannot use the typedef specifier inside a function definition.

Syntax

```
typedef datatype newdatatype;
```

Example:

```
typedef int INTEGER;  
INTEGER num=29;
```

Here, INTEGER is the new name of data type int.

Program

```
#include<stdio.h>  
#include<conio.h>  
typedef char  uchar;  
void main()  
{  
    uchar ch = 'a';  
    printf("ch: %c\n", ch);  
    getch();  
}
```

Output

```
ch : a
```

typedef in structure

We can use typedef with a structure. If the typedef keyword is given before the keyword struct, then the struct becomes a new type. For example,

```
typedef struct student  
{  
    int regno;  
    char name[20];
```

```
};
```

Now, student is the new data type. The variables of the structure student can be declared as follows:

```
student stud;
```

Program

```
#include <stdio.h>
#include <string.h>
typedef struct student
{
    int id;
    char name[20];
    float percentage;
} status;
void main()
{
    status record;
    record.id=1;
    strcpy(record.name, "Raju");
    record.percentage = 86.5;
    printf(" Id is: %d \n", record.id);
    printf(" Name is: %s \n", record.name);
    printf(" Percentage is: %f \n", record.percentage);
    getch();
}
```

Output

```
Id is: 1
```

```
Name is: Raju
```

```
Percentage is: 86.500000
```