

Enum in C

An enum (enumeration) in C is a user-defined data type that contains a set of named integer constants. It helps us use meaningful names instead of numeric values in our program.

Example:-

```
enum Day
{
    MONDAY, ✓ → Integer constants 0
    TUESDAY, ✓ → 1
    WEDNESDAY, ✓ → 2
    THURSDAY, ✓ → 3
    FRIDAY, ✓ → 4
    SATURDAY, ✓ → 5
    SUNDAY, ✓ → 6
};
```

```
enum Day today = SUNDAY;
```

```
printf("today is %d", today);
```

~~SUNDAY~~ 6 ✓

```
if (day == 6)
```

```
printf("It's Sunday today");
```

=X=

```
enum Day
{
    MONDAY = 5,
    TUESDAY = 6,
    WEDNESDAY = 7,
    THURSDAY = 8,
    FRIDAY = 9,
    SATURDAY = 10,
    SUNDAY = 11
};
```

5	0
6	1
7	2
8	3
9	4
10	5
11	6
12	7

```
enum Day today = SUNDAY; ✓
```

```
printf("today is %d", today); → 11
```

int today = 11;

=X=

```
enum MachineStatus
```

```
{
    RUNNING, ✓
    STOPPED, ✓
    NOTWORKING ✓
};
```

```
};
```

```
enum MachineStatus status = NOTWORKING; //2
```

```
switch (status)
```

```
{
    ✓ case RUNNING: printf("Machine is Running"); break;
```

```
    ✓ case STOPPED: printf("Machine is stopped"); break;
```

```
case NOTWORKING: printf("Machine not working"); break;
}
```

Annotation :- enum variable can store integer value which is not in Integer constants list.

✓ enum MachineStatus status = 100; // this not error

Note :- Enum names are often written in capital letters, but C doesn't require this

enum DAY ✓

enum Day ✓

enum day ✓

Summary

- ① enum is a user-defined data type
- ② enum constants have integer values
- ③ By default values start at 0 and increment by 1.
- ④ You can assign custom values also. (start from 0 is not mandatory)
- ⑤ Memory is created when we create variable of enum.
- ⑥ In C enum variable can have any ~~any~~ outside value which is not in listed integer constants.