Welcome,

PROGRAMMERS

01.

What is Keyword?

Keyword

A **reserved word** which have a **specific meaning**.

C language has total **32 keywords**.

auto	break	case	char
const	continue	default	do
double	else	enum	extern
float	for	goto	if
int	long	register	return
short	signed	sizeof	static
struct	switch	typedef	union
unsigned	void	volatile	while

Total Keywords

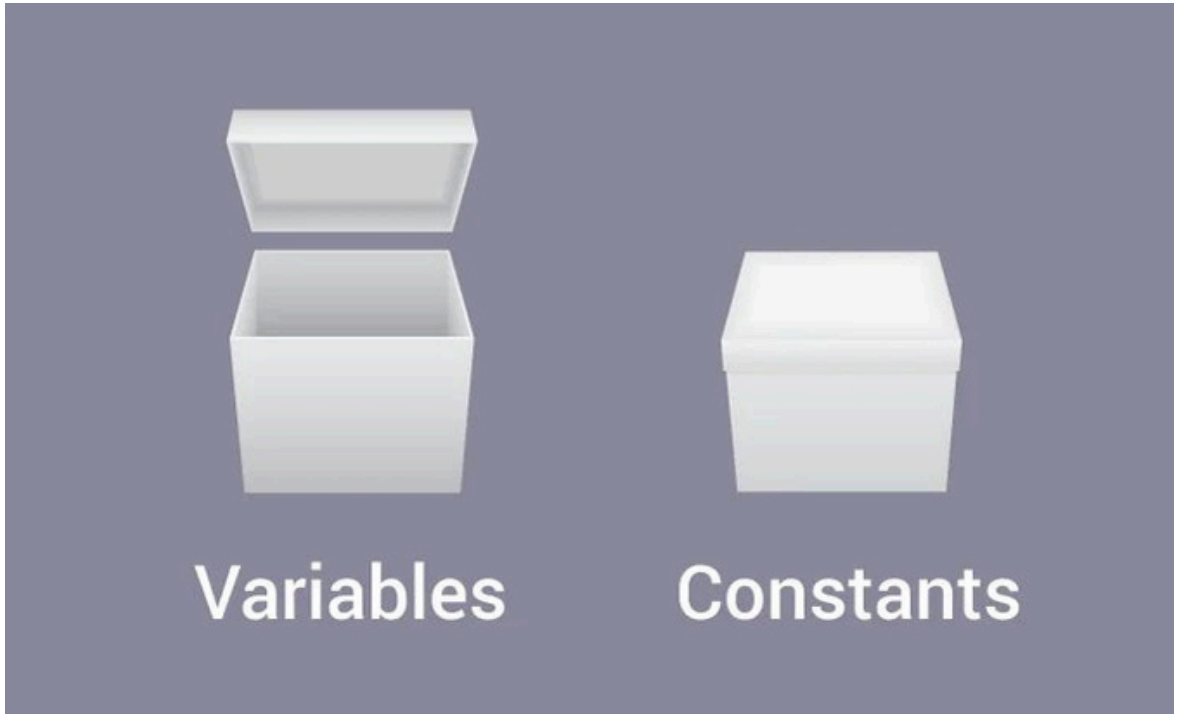
02.

What is Constant?

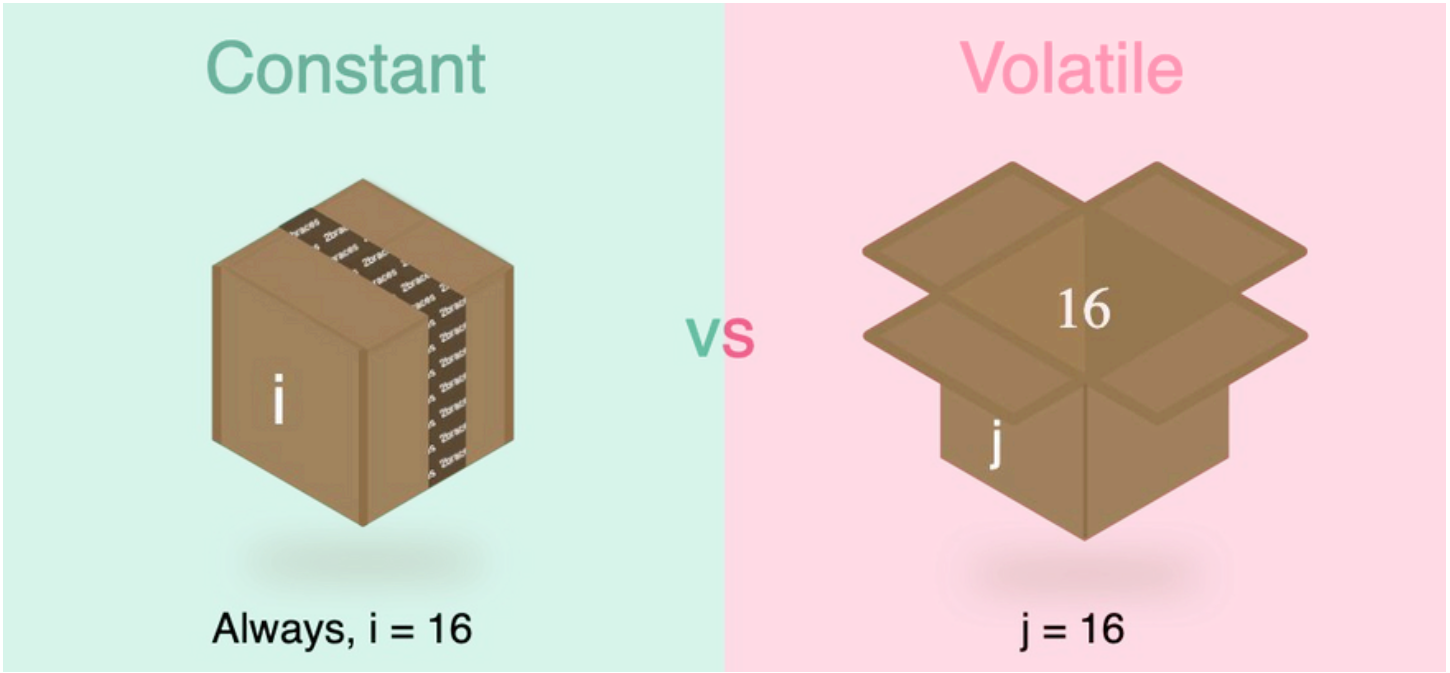
Constant

A **constant** is a kind of **variable** which **value is fixed** after initialization.
Constant always keeps value unchanged.
Constant value **can't be changed**.

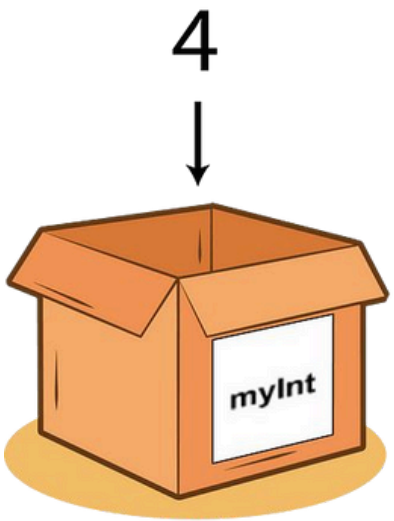
Variable / Constant



Variable / Constant

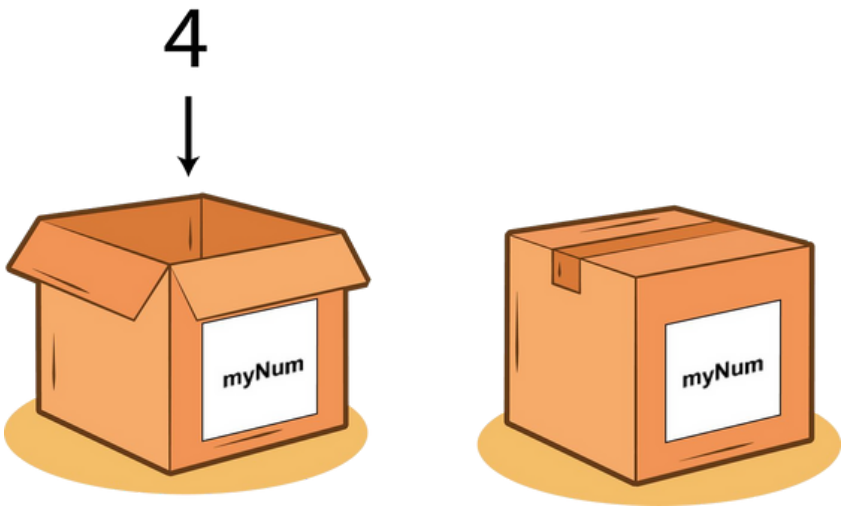


Volatile Integer




```
int myInt = 4;
```

Constant Integer




```
const int myNum = 4;
```

or

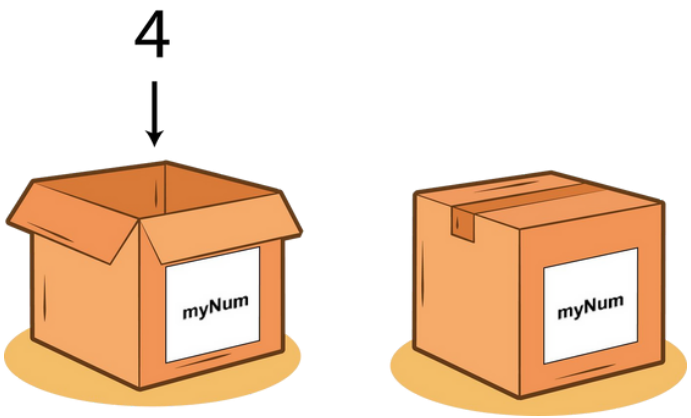
```
#define myNum = 4;
```

How to create Constant?

A **constant** can be created using two ways:

1. Using **const** keyword
2. Using a **macro**

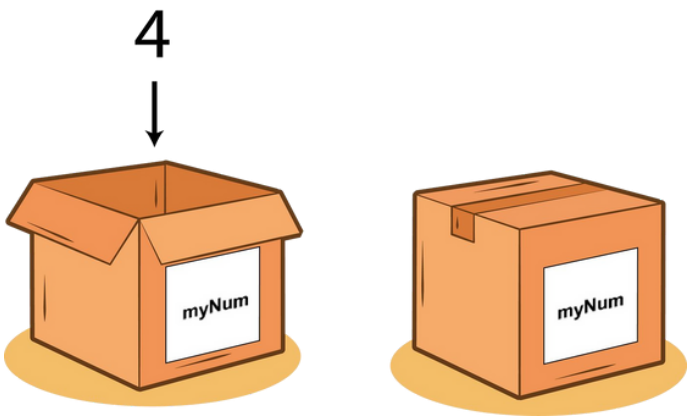
Using const keyword




```
const int myNum = 4;

myNum = 10;  // error, can't be changed
```

Using macro




```
#define myNum 4

myNum = 10;  // error, can't be changed
```

Rules of Constant

Constant **value** must have to **initialized** while **declaration** of variable.

```
const int myNum = 4;
or
#define myNum 4
```

```
const int myNum;
myNum = 4;
or
#define myNum
myNum = 4
```

Rules of Constant

Constant variable name should have **capitalized** as per global convention.

```
const int MAX = 999;
or
#define MAX 999
```

```
const int max = 999;
or
#define max 999
```

Rules of Constant

Macro is a **standard choice** as per global convention rather than using const keyword.

```
#define MAX 999
```

```
const int MAX = 999;
```

03.

What is Format specifier?

format Specifier

Format specifiers is a symbol of **%**.
Format specifiers **tell the compiler** about the **type**

of data that must be given for input and the type of data that must be printed on the screen.

format Specifier

Format Specifier	Type
%c	Used to print a character
%d	Used to print the signed integer
%f	Used to print the float values
%i	Used to print the unsigned integer
%l	Used to print the long integer
%lf	Used to print the double values
%lu	Used to print the unsigned integer or unsigned long integer
%s	Used to print the string
%u	Used to print the unsigned integer

01

Integer

%d

For integer

%d is commonly used for showing integer value.

```
int a = 7;
printf("a=%d", a);
```

Output:

a=7

For integer

A digit specified in between **%d** is used for **acquire total number of position** for integer value including remaining **space in front of actual value**.

```
int a = 7;
printf("a=%3d", a);
```

Output:

a=__7

// _ is whitespace

For integer

A digit specified after dot in between **%d** is used for **showing total number of integer values** including 0 & **actual value**.

```
int a = 7;
printf("a=%3.2d", a);
```

Output:

a=_07

// _ is whitespace

02

float

%f

For Float

%f is commonly used for showing float value.

```
float a = 7;
printf("a=%f", a);
```

Output:

a=7.000000

For float

A digit specified in between **%f** is used for **acquire total number of position** for float value including remaining **space in front of actual value**.

```
float a = 7;
printf("a=%10f", a);
```

Output:

a=__7.000000

// _ is whitespace

For float

A digit specified after dot in between **%f** is used for **showing number of 0s after float value**.

```
float a = 7;
printf("a=%0.2f", a);
```

Output:
a=7.00

```
float a = 7;
printf("a=%10.2f", a);
```

Output:
a=_____7.00

// _ is whitespace

character

%c

For character

%c is commonly used for showing character value.

```
char a = ‘r’;
printf(“a=%c”, a);
```

Output:

a=r

For character

A digit specified in between **%c** is used for **acquire total number of position** for character value including remaining **space in front of actual value**.

```
char a = ‘r’;
printf(“a=%3c”, a);
```

Output:

a=__r

// _ is whitespace

04.

What is scanf()?

scanf()

scanf() is a function used for **take input** from the user **on console**.

Syntax of scanf() is **same** as printf() function.

```
scanf("format_specifier", &variable_name);
```

Rules of **scanf()**

A **whitespace** should **not** be used **after** a format specifier.

```
scanf("%d", &var);
scanf(" %d", &var);
scanf(" %d", &var);
```

```
scanf("%d ", &var);
```

Rules of **scanf()**

A message or any letter **not allowed**.

```
scanf(“%d”, &var);
```

```
scanf(“Hi %d”, &var);
```

Language

Let’s start now...