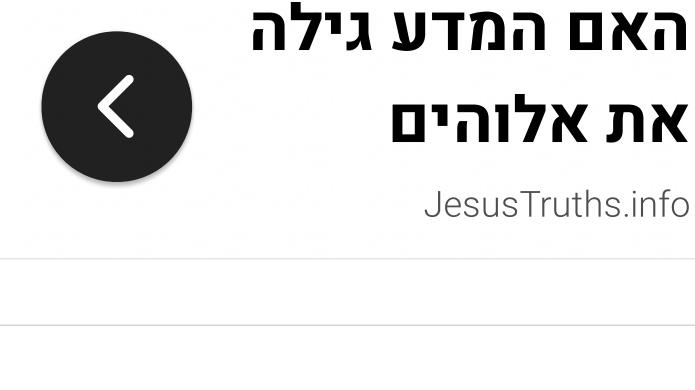
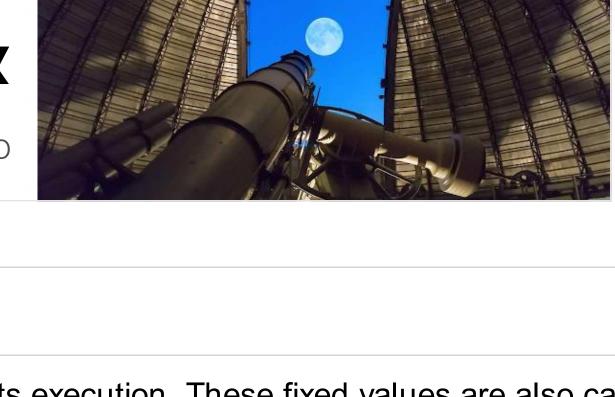


Advertisements

C - Constants and Literals





Constants can be of any of the basic data types like an integer constant, a floating constant, a character constant, or a string literal. There are enumeration constants as well.

Integer Literals

octal, and nothing for decimal.

uppercase or lowercase and can be in any order.

215u

301

30ul

510E

210f

.e55

Previous Page

Here are some examples of integer literals – /* Legal */ 212

/* Illegal: 8 is not an octal digit */ 078 /* Illegal: cannot repeat a suffix */ 032UU

/* Legal */

/* long */

/* unsigned long */

literals either in decimal form or exponential form.

0xFeeL /* Legal */

```
Following are other examples of various types of integer literals –
          /* decimal */
85
     /* octal */
0213
     /* hexadecimal */
0x4b
30
    /* int */
    /* unsigned int */
30u
```

Floating-point Literals

/* Illegal: incomplete exponent */

Following is the example to show a few escape sequence characters –

When the above code is compiled and executed, it produces the following result -

/* Illegal: no decimal or exponent */

/* Illegal: missing integer or fraction */

```
While representing decimal form, you must include the decimal point, the exponent, or both; and while representing exponential form,
you must include the integer part, the fractional part, or both. The signed exponent is introduced by e or E.
Here are some examples of floating-point literals –
3.14159
         /* Legal */
314159E-5L /* Legal */
```

A floating-point literal has an integer part, a decimal point, a fractional part, and an exponent part. You can represent floating point

Character Constants

```
printf("Hello\tWorld\n\n");
return 0;
```

"hello, " "d" "ear"

Defining Constants

#include <stdio.h>

#define LENGTH 10

#define NEWLINE '\n'

#define WIDTH 5

There are two simple ways in C to define constants -

Using #define preprocessor.

"hello, dear"

"hello, \

dear"

Hello World

The following example explains it in detail -

int main() { int area;

```
The following example explains it in detail -
#include <stdio.h>
int main() {
   const int LENGTH = 10;
   const int WIDTH = 5;
```



Live Demo

Live Demo

Next Page **⊘**

* Refund Policy About us

tutorialspoint ::: Categories -

Constants are treated just like regular variables except that their values cannot be modified after their definition.

Constants refer to fixed values that the program may not alter during its execution. These fixed values are also called literals.

Next Page **⊘**

An integer literal can be a decimal, octal, or hexadecimal constant. A prefix specifies the base or radix: 0x or 0X for hexadecimal, 0 for An integer literal can also have a suffix that is a combination of U and L, for unsigned and long, respectively. The suffix can be

Character literals are enclosed in single quotes, e.g., 'x' can be stored in a simple variable of char type. A character literal can be a plain character (e.g., 'x'), an escape sequence (e.g., '\t'), or a universal character (e.g., '\u00bb00000). There are certain characters in C that represent special meaning when preceded by a backslash for example, newline (\n) or tab (\t). Here, you have a list of such escape sequence codes -

Live Demo #include <stdio.h> int main() {

```
String Literals
String literals or constants are enclosed in double quotes "". A string contains characters that are similar to character literals: plain
characters, escape sequences, and universal characters.
You can break a long line into multiple lines using string literals and separating them using white spaces.
Here are some examples of string literals. All the three forms are identical strings.
```

Using **const** keyword. The #define Preprocessor Given below is the form to use #define preprocessor to define a constant -#define identifier value

```
area = LENGTH * WIDTH;
   printf("value of area : %d", area);
   printf("%c", NEWLINE);
   return 0;
When the above code is compiled and executed, it produces the following result -
value of area: 50
The const Keyword
You can use const prefix to declare constants with a specific type as follows –
const type variable = value;
```

return 0;

Advertisements

האם המדע גילה את אלוהים מהן אותן תגליות מופלאות שגרמו למדענים לדבר פתאום על אלוהים JesusTruths.info

const char NEWLINE = '\n'; int area; area = LENGTH * WIDTH; printf("value of area : %d", area); printf("%c", NEWLINE);

When the above code is compiled and executed, it produces the following result value of area: 50 Note that it is a good programming practice to define constants in CAPITALS.

(i) (i)

