程序的开发过程中需要使用到各种数据类型。

字符常量**literal** 就是它本身代表的值。比如25、hello world，他们代表的就是他们自己本身的值。

常量**constant代表了一个字符常量。**常量一般用于：

1. 一个指定的、不会改变的值，它有可能被在程序中使用多次。
2. 为了更好的理解代码（比如PI代表3.14）。

程序中的变量在执行过程中会改变，但是常量在程序的执行过程中始终代表一个值。

**Comparison chart**

| Constant versus Literal comparison chart | | |
| --- | --- | --- |
| [Edit this comparison chart](http://www.diffen.com/difference/Special:EditTable?diffenVal1=Constant&diffenVal2=Literal) | **Constant** | **Literal** |
| **Example** | const PI = 3.14; var radius = 5; var circumference = 2 \* PI \* radius; | var radius = 5; var circumference = 2 \* 3.14 \* radius; |

**Constant vs Literal Data Type - Example**

Suppose we are writing a program to determine which members of a population are eligible to vote, permitted to drink, both or neither.

const DRINKING\_AGE = 21;

const VOTING\_AGE = 18;

18 and 21 are literals. We can use these **literals** in all areas of our program. For example, if(*age* > 18) or if(*age* < 21). But we can make our code more understandable if we use **constants** instead. if(*age* > VOTING\_AGE) is easier to understand. Other benefits of using constants are

* Constants free the programmer from having to remember what each literal should be. Often values that stay constant throughout the program have a business meaning. If there are several such values, the programmer can define them all in the beginning of the program and then work with the easier-to-remember constant names.
* If business requirements dictate that the constant be changed (for example, if the drinking age is lowered to 20 in the future), it is much easier to adapt the program. If we use literals throughout the program, the change will be hard to do and there is a good chance some instances will not be corrected.

**References**

* [wikipedia:Literal (computer programming)](http://en.wikipedia.org/wiki/Literal_%28computer_programming%29)
* [wikipedia:Constant (computer programming)](http://en.wikipedia.org/wiki/Constant_%28computer_programming%29)