



What is variable

- a box to hold datas
- how to create this box?
 - in python, we do not create, just use
- how to put datas?
 - use equal mark "="
 - when prigramming, this does not represents "equal", just mean put a data into that box, we call this "Assignment".
 - For example

```
n = 10  
print(n)
```

Some types of variable

int

An integer is a number without a decimal part, such as `1`, `88888888888888`, `59603`.

float

In programming, usually we use "float" to represent those numbers with decimal point, such as `0.1`, `569.789`, `10000000.000001`

string

A collection of characters is a String. The string can contain letters, punctuation, special symbols, Chinese, Japanese, and all the world's characters.

A string in Python must be surrounded by either double or single quotation marks, in the form of:

```
"abc"
```

or

```
'abc'
```

There is no difference between double and single quotes in Python strings.

Deal with string in quotation marks

When quotation marks appear in the string content, we need to do something special, otherwise Python will parse error, such as:

```
'I'm a great coder!'
```

Because the above string contains single quotes, Python pairs the single quotes in the string with the first single quotation mark, which treats the `'I'` as a string, followed by the `m a great coder!'` becomes superfluous and leads to syntax errors.

There are two ways to deal with this situation:

1. Escape quotation marks

You can escape the quotation marks by adding a backslash `\` to them, allowing Python to treat them as plain text, such as

```
'I\'m a great coder!'
```

2. Surround the string with different quotation marks

If a single quotation mark appears in the string content, we can enclose the string with double quotation marks and vice versa. For example:

```
"I'm a great coder!"
```

bool

Python provides a bool type to represent true or false. For example, the $5 > 3$ comparison, which is correct, and python uses `True` to represent it, and the $4 > 20$ comparison, which is incorrect, so python uses `False` to represent it.

Bool types can be treated as integers, where `True` equals an integer value of 1 and `False` equals an integer value of 0

Python is a weakly typed language

Weakly typed languages have two characteristics:

Variables can be assigned directly without having to declare them. Assigning a value to a variable that does not exist is equivalent to defining a new variable.

The data type of a variable can change over time. For example, the same variable can be assigned as an integer or a string.

Note that weak typing is not the same as no typing! Weak typing means you don't have to worry about typing when you write code, but it still has typing inside a programming language.

```
num = 10
type(num)
num = 15.8
type(num)
```

arithmetic operators in python

operators	description	examples	results
+	plus	1.3 + 3	4.3
-	minus	1.3 - 0.3	1
*	times	5 * 3.6	18.0
/	division (as in mathematics)	7/2	3.5
//	integer division (only the integer part of the quotient is retained)	7//2	3
%	mod(returns the remainder of a division)	7%2	1
**	power Operation (returns x to the y power)	2**4	16, aka 2 ⁴

how to debug(find where is wrong)

- read the error notice!!
- with print
- with breakpoint