

概念

语句规则

变量

语句

代码块

注释

行注释

多行注释

在Python中使用帮助

基础语法

输入

输出

作业

概念

Python编写的代码，从粗细可以分为模块文件、代码块、代码三部分。

在代码模块中又有“函数”和“类”的概念。

语句规则

变量

1. 变量的命名

可以使用字符、数字和下划线的组合，首字母可以是字母或者下划线(**不能使用数字开头**)。

```
1 Code
2 _data
3 _name_
4 tt
```

语句

在Python解释器中，源代码是一行一行被解释执行的，行与行之间通过换行符来区分。默认是一行一条语句，如果在一行中有多个语句来进行执行，我们需要通过 `;` 来进行区分。

```
1 A=6
2 B=7;C=8
```

代码块

通过缩进来表述代码块的，即同一个代码块中的语句具有相同的缩进格式。缩进可以是：使用空格(四个)或者是TAB键来实现的。

```
1 def fun():#定义一个函数名字叫做fun
2     print "hello"#该语句被缩进，表示fun函数的函数体，只有执行fun函数时才
   被执行
3
4 print "Go"#该语句没有缩进，直接就执行了
5 fun()#该语句没有缩进，会执行其函数体内的语句
```

注释

行注释

英文字符 `#` 用来代表注释，他的生效范围是“行”，也就是说在一行代码中 `#` 之后的内容将不会被Python解释器处理。

注释一般用来说明代码的含义。

多行注释

使用的是 `'''` (三个单引号)或者 `"""` (三个双引号)将代码变成字符串，程序同样不会执行。

```

1  '''
2  def fun():#定义一个函数名字叫做fun
3      print "hello"#该语句被缩进，表示fun函数的函数体，只有执行fun函数时才
      被执行
4
5  print "Go"#该语句没有缩进，直接就执行了
6  fun()#该语句没有缩进，会执行其函数体内的语句
7  '''

```

在Python中使用帮助

在使用Python的过程中可以使用 `help` 来获取一些帮助信息，他可以查找关于Python的基础函数，类型，常用库的信息。

```

1  >>> help()
2  help> print
3  The "print" statement
4  *****
5
6      print_stmt ::= "print" ([expression ("," expression)* [","]]
7                      | ">>" expression [("," expression)+ [","]])
8
9  "print" evaluates each expression in turn and writes the
10 resulting
11 object to standard output (see below). If an object is not a
12 string,
13 it is first converted to a string using the rules for string
14 conversions. The (resulting or original) string is then written.
15 A
16 space is written before each object is (converted and) written,
17 unless
18 the output system believes it is positioned at the beginning of a
19 line. This is the case (1) when no characters have yet been
20 written
21 to standard output, (2) when the last character written to
22 standard
23 output is a whitespace character except "' '", or (3) when the
24 last

```

18 write operation on standard output was not a `"print"` statement.
(In
19 some cases it may be functional to write an empty string to
standard
20 output `for this` reason.)
21
22 Note: Objects which act like file objects but which are not the
23 built-in file objects often `do` not properly emulate `this` aspect
of
24 the file object's behavior, so it is best not to rely on this.
25
26 A `"'\n'"` character is written at the end, unless the `"print"`
statement
27 ends `with` a comma. This is the only action `if` the statement
contains
28 just the keyword `"print"`.
29
30 Standard output is defined as the file object named `"stdout"` in
the
31 built-in module `"sys"`. If no such object exists, or `if` it does
not
32 have a `"write()"` method, a `"RuntimeError"` exception is raised.
33
34 `"print"` also has an extended form, defined by the second portion
of
35 the syntax described above. This form is sometimes referred to as
36 `"print chevron."` In `this` form, the first expression after the
`">>"`
37 must evaluate to a `"file-like"` object, specifically an object
that has
38 a `"write()"` method as described above. With `this` extended form,
the
39 subsequent expressions are printed to `this` file object. If the
first
40 expression evaluates to `"None"`, then `"sys.stdout"` is used as the
file
41 `for` output.
42
43 `help>`
44
45 You are now leaving help and returning to the Python interpreter.

```
46 | If you want to ask for help on a particular object directly from  
    | the  
47 | interpreter, you can type "help(object)". Executing  
    | "help('string')"  
48 | has the same effect as typing a particular string at the help>  
    | prompt.
```

如上操作等同于 `help('print')`

基础语法

输入

在python2中，获取输入是使用的 `raw_input` 函数

```
year = raw_input()
```

输出

在Python2中，输出语句为: `print`

```
print year
```

作业

完成如下的操作。

tips:使用print和raw_input完成。

```
C:\Python27\python.exe C:/Users/98165/Desktop/main.py
```

你叫什么呀？

吴小岑

你多少岁了鸭？

3

我是吴小岑小朋友，今年3岁了！

注：绿色为输入的内容