1.打印输出

使用print() 语句实现在屏幕上打印你需要的输出内容

In [4]:

```
print("hello,world!")
```

```
hello,world!
```

多行输出

In [8]:

```
print("Hello")
print("Here is a Python Program")
print("Python is my favorite programming language!")
print("谢谢！")
```

```
Hello
Here is a Python Program
Python is my favorite programming language!
谢谢！
```

2. 字符串

"hello,wolrd!"这样的文本被成为字符串

字符串规范

In [5]:

```
#引号开头和结尾，如果是像如下这样就会出错
print("hello)
```

```
  File "C:\Users\85204\AppData\Local\Temp\ipykernel_4312\3081832087.p
y", line 2
    print("hello)
                 ^
SyntaxError: EOL while scanning string literal
```

In [15]:

```
#不能跨越多行，如果是像如下这样就会出错
print("hello
      world
      ")
```

```
  File "C:\Users\85204\AppData\Local\Temp\ipykernel_4312\171356250.p
y", line 2
    print("hello
                ^
SyntaxError: EOL while scanning string literal
```

In [16]:

```
#如果实在需要跨行，可以用如下方式
print('''
hello,
world!
''')
```

```
hello,
world!
```

转义字符：用于表示特殊字符

\t:制表符
\n：换行符
\""双引号
\'单引号
\\ 反斜杠

In [10]:

```
#如果我们想打印的文本带有引号？
print("What \"characters\" does this print?")
```

```
What "characters" does this print?
```

In [13]:

```
#如果我们想打印的文本需要换行
print("This \nis \nPython")
```

```
This
is
Python
```

3. 打印复杂图形

In [17]:

```
print('''
   _          _ _ _ _ _ _ _ _           _            _ _ _ _          _
  (_)        (_)(_)(_)(_)(_)(_)(_)       (_)        _(_)(_)(_)(_)_       (_)
  (_)        (_)(_)           (_)       (_)       (_)        (_)      (_)
  (_) _ _ _  (_)(_) _ _        (_)       (_)       (_)        (_)      (_)     _
  (_)(_)(_)(_)(_)(_)(_)(_)     (_)       (_)       (_)        (_)      (_)   _(_)
  (_)        (_)(_)           (_)       (_)       (_)        (_) _ _  (_)  (_)
  (_)        (_)(_) _ _ _ _   (_) _ _ _ (_)       (_) _ _ _ _(_) (_)(_) (_)_(_)
  (_)        (_)(_)(_)(_)(_)(_)(_)(_)(_)(_)(_)(_)(_)(_)(_)(_)(_)(_)(_)(_)   (_)(_)    (_)
''')
```

```
   _          _ _ _ _ _ _ _ _           _            _ _ _ _
_            _    _ _ _ _ _ _ _ _ _    _          _ _ _ _ _     _
  (_)        (_)(_)(_)(_)(_)(_)(_)        (_)         _(_)(_)(_)(_)_
(_)          (_) _(_)(_)(_)(_)_ (_)(_)(_)(_) _ (_)         (_)(_)(_)(_)
(_)
  (_)        (_)(_)           (_)          (_)         (_)         (_)
(_)          (_)(_)         (_)(_)        (_)(_)        (_)      (_)_
(_)
  (_) _ _ _  (_)(_) _ _        (_)          (_)         (_)         (_)
(_)     _    (_)(_)         (_)(_) _ _ _ (_)(_)         (_)       (_)
(_)
  (_)(_)(_)(_)(_)(_)(_)(_)     (_)          (_)         (_)         (_)
(_)   _(_)_   (_)(_)         (_)(_)(_)(_)(_)   (_)         (_)       (_)
(_)
  (_)        (_)(_)           (_)          (_)         (_)         (_) _
_   (_)  (_) (_)  (_)(_)         (_)(_)   (_) _    (_)           (_)      _
(_)
  (_)        (_)(_) _ _ _ _ (_) _ _ _ _ (_) _ _ _ _(_)_ _ _ _(_)
(_)(_)  (_)_(_)   (_)_(_)(_)_  _ _ _(_)(_)     (_) _ (_) _ _ _ _ (_)_ _
(_)    _
  (_)        (_)(_)(_)(_)(_)(_)(_)(_)(_)(_)(_)(_)(_)(_)(_)(_)(_) (_)(_)(_)(_)
(_)(_)    (_)      (_)    (_)(_)(_)(_)  (_)        (_)(_)(_)(_)(_)(_)(_)(_)(_)
(_)    (_)
```

用Python打印一个菱形

In [18]:

```
print("   /\\")
print("  /  \\")
print(" /    \\")
print(" \\    /")
print("  \\  /")
print("   \\/")
```

```
   /\
  /  \
 /    \
 \    /
  \  /
   \/
```

4. 注释、空白、可读性

一个可读性较差的程序

In [24]:

```
print(
"Look at this beautiful program!"
)

print(
"I do belive it is")

print("The best program in the world")

print(".....")
```

```
Look at this beautiful program!
I do belive it is
The best program in the world
.....
```

如何改善程序的可读性

- 将每个语句单独放在一行上
- 使用空行分隔不同
- 功能的语句
- 编写程序注释

程序注释

In [29]:

```
#given an introduction to the user
print("Welcome to Python")

#洗菜
print("...")

#烧水
print("...")

#炒菜
print("...")
```

```
Welcome to Python
...
...
...
```

In [31]:

```
'''
这是一个多行注释
'''
print("hello,world")
```

```
hello,world
```

5.语法错误

尽管人们对口语中的小错误非常宽容，但是Python解释器并不那么宽容。如果我们的程序没有严格地按照语法规则执行，解释器会在将你的程序从Python语言转换到可执行的机器语言的时候报告语法错误。对于一个精密运行的系统而言，一点点微小的错误可能是致命的。

拼写错误

In [33]:

```
prunt("hello,world!")   #尝试理解错误报告
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_4312\3556354443.py in <module>
----> 1 prunt("hello,world!")

NameError: name 'prunt' is not defined
```

In [34]:

```
Print("hello,world!")
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_4312\1254609384.py in <module>
----> 1 Print("hello,world!")

NameError: name 'Print' is not defined
```

In [35]:

```
print("hello,world)
```

```
  File "C:\Users\85204\AppData\Local\Temp\ipykernel_4312\1926936859.p
y", line 1
    print("hello,world)
                       ^

SyntaxError: EOL while scanning string literal
```

In [36]:

```
print(hello,world)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_4312\2595165328.py in <module>
----> 1 print(hello,world)

NameError: name 'hello' is not defined
```

课间练习： 用Python 打印一个简单的图形，并在图形中涉及到字符串

例如： 打印一个火箭

In [ ]:

```

```