

0. IDLE 的交互模式和编辑器模式有什么区别？

交互模式

写一个代码，反馈一下

编辑器模式

可以得到一连串代码所得出的反馈

答：交互模式相当于跟 **Python** 的解析器“你一言我一语”的对话模式，你输入一个命令表达式，**Python** 将计算结果立即给你反馈回来。

1. 在课堂上敲过的代码中，除了 `print()` 和 `input()`，你觉得还有哪一个是 **Python** 的 **BIF** 内置函数

`int()`

答：`int()`。

提示：事实上它是一个工厂函数（**Factory Function**），不过你，暂时先把它当作函数用就可以了。

2. 请问 `print()` 和 `Print()` 的功能一样吗？

不一样

`print()` 是 **python** 的内置函数

而 `Print()` 不是

答：不一样，`print()` 是 **Python** 的内置函数，用于打印输出一个对象。由于 **Python** 是区分大小写的，所以如果你输入 `Print("小甲鱼是帅哥吗？")`，**Python** 会报错。

3. 请统计一下 **Python** 一共有多少个 **BIF** 内置函数？

153

答：68 个。

解析：有童鞋可能会说，视频里演示的可不止 68 个呢？！

是的，使用 `dir(builtins)`，显示的内容除了内置函数（**Built-in Function**），还包含了一些内置常量（**Built-in Constants**）和内置异常（**Built-in Exception**）等内容。

所以，内置函数只有以下 68 个：

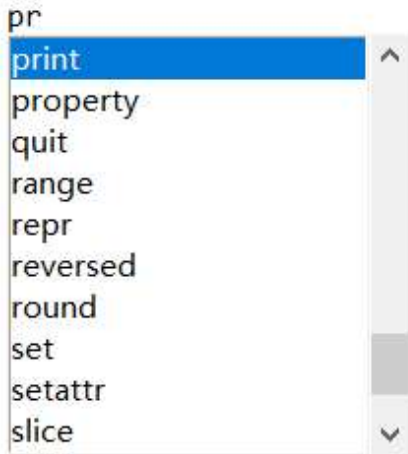
Built-in Functions				
<code>abs()</code>	<code>dict()</code>	<code>help()</code>	<code>min()</code>	<code>setattr()</code>
<code>all()</code>	<code>dir()</code>	<code>hex()</code>	<code>next()</code>	<code>slice()</code>
<code>any()</code>	<code>divmod()</code>	<code>id()</code>	<code>object()</code>	<code>sorted()</code>
<code>ascii()</code>	<code>enumerate()</code>	<code>input()</code>	<code>oct()</code>	<code>staticmethod()</code>
<code>bin()</code>	<code>eval()</code>	<code>int()</code>	<code>open()</code>	<code>str()</code>
<code>bool()</code>	<code>exec()</code>	<code>isinstance()</code>	<code>ord()</code>	<code>sum()</code>
<code>bytearray()</code>	<code>filter()</code>	<code>issubclass()</code>	<code>pow()</code>	<code>super()</code>
<code>bytes()</code>	<code>float()</code>	<code>iter()</code>	<code>print()</code>	<code>tuple()</code>
<code>callable()</code>	<code>format()</code>	<code>len()</code>	<code>property()</code>	<code>type()</code>
<code>chr()</code>	<code>frozenset()</code>	<code>list()</code>	<code>range()</code>	<code>vars()</code>
<code>classmethod()</code>	<code>getattr()</code>	<code>locals()</code>	<code>repr()</code>	<code>zip()</code>
<code>compile()</code>	<code>globals()</code>	<code>map()</code>	<code>reversed()</code>	<code>__import__()</code>
<code>complex()</code>	<code>hasattr()</code>	<code>max()</code>	<code>round()</code>	
<code>delattr()</code>	<code>hash()</code>	<code>memoryview()</code>	<code>set()</code>	

4. Tab 键除了用于缩进，你还发现它在 IDLE 中有什么特殊的功能吗？

可以用tab键查找或者补全BIF函数命令

答：缩进在 **Python** 中，是统一使用 **Tab** 键来实现的，默认它是相当于 4 个空格，这里建议大家不要去修改它。

Tab 键的另一个妙用就是自动填充，比如咱输入 **pr** 两个字符，再摁一下键盘上的 **Tab** 键，会弹出一个提示款，**IDLE** 在猜测你想干啥呢：



这些都是 **Python** 的 **BIF** 内置函数，这样就算你忘了这个函数具体叫啥名，这个功能也能让你想起个大概来，对吧？！

5. 请问下面代码为什么不能正常执行？

```
""" 用Python设计第一个游戏 """
temp = input("不妨猜一下小甲鱼现在心里想的是哪个数字：")
guess = int(temp)

if guess = 8:
    print("你是小甲鱼心里的蛔虫嘛？！")
    print("哼，猜中了也没奖励！")
else:
    print("猜错啦，小甲鱼现在心里想的是8！")

print("游戏结束，不玩啦^_^")
```

If 和 else 没有正确缩进

答：因为粗心！

解析：在 **Python** 中，如果用于判断两数是否相等，应该使用连续的两个等号（**==**），而不是一个。

动手手：版权属于：<https://fishc.com.cn>

0. 请在 **IDLE** 的交互模式中，计算一年有多少秒？

```
>>> 365*24*60*60
31536000
```

答：

```
>>> 365 * 24 * 60 * 60
31536000
```


1. 按下面要求修改课堂中的 `game.py` 代码。

让用户输入这次数学考试的成绩。

如果分数是 100 分，显示：好棒，你离女神又近了一步–

如果分数不是 100 分，显示：小子，想要幸福，就得努力！

程序实现如下：



```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\goodb\Desktop\game.py =====
这次数学考试成绩: 58
小子，想要幸福，就得努力!
游戏结束，不玩啦^_^
>>>
===== RESTART: C:\Users\goodb\Desktop\game.py =====
这次数学考试成绩: 100
好棒，你离女神又近了一步^_^
游戏结束，不玩啦^_^
>>>
```

```
temp=input("这次数学考试成绩:")
```

```
guess=int(temp)
```

```
If guess==100:
```

```
print("好棒，你离女神又近一步–")
```

```
else:
```

```
print("小子，想要幸福，就得努力!")
```

```
print("游戏结束，不玩啦–")
```

答：

```
""" 用Python设计第一个游戏 """

temp = input("这次数学考试成绩:")
guess = int(temp)

if guess == 100:
    print("好棒，你离女神又近了一步^_^")
else:
    print("小子，想要幸福，就得努力！")

print("游戏结束，不玩啦^_^")
```