

For循环

确定次数循环

一个整蛊程序

In [2]:

```
import tkinter
import tkinter.messagebox
import webbrowser

if __name__ == '__main__':

    for i in range(5):
        tkinter.messagebox.showerror("Windows错误！")

    for i in range(5):
        webbrowser.open("www.baidu.com")
```

编程常常涉及到重复的任务，而循环结构则恰好用于避免这种规律性的重复任务。
例如，打印10遍hello,world!

In []:

确定性for循环的语法结构：

In [1]:

```
'''
for 变量 in range(次数):
    循环体代码块
'''
```

Out[1]:

```
'\nfor 变量 in range(次数):\n    循环体代码块\n'
```

练习：设计一个重复的消息弹窗

In []:

In []:

```
import tkinter as tk
import random
import webbrowser

def boom():
    window = tk.Tk()
    width = window.winfo_screenwidth()
    height = window.winfo_screenheight()
    a = random.randrange(0, width-20)
    b = random.randrange(0, height-20)
    window.title('警告')
    window.geometry("300x50" + "+" + str(a) + "+" + str(b))
    tk.Label(window, text='Windows系统出现错误!', bg='red',
             font=('宋体', 17), width=20, height=4).pack()
    window.mainloop()

def openWeb():
    webbrowser.open("www.baidu.com")

if __name__ == '__main__':
    for i in range(5):
        boom()

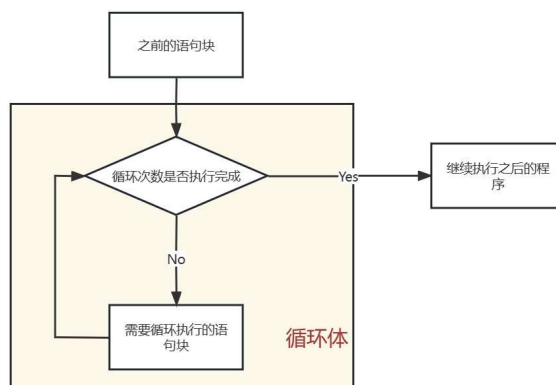
    for i in range(5):
        openWeb()
```

如何将.py程序打包为可执行文件.exe?

- 1.命令行: pip install pyinstaller
- 2.命令行: cd (xxx.py文件所在路径)
- 3.命令行: pyinstaller -F xxx.py

参考教程: <https://blog.csdn.net/libaineu2004/article/details/112612421>
(<https://blog.csdn.net/libaineu2004/article/details/112612421>)

利用循环变量



In []:

用for循环实现累加：

如何实现1+2+3+4+...+10？

In []:

用for循环遍历列表

In []:

小练习：For循环在数值计算中的应用

(1) 用于验证公式：已知 $\frac{\pi}{4} = \frac{1}{1} - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots$ ，请写一个程序，用于计算圆周率 π ：

In []:

(2) 用于求解方程：用数值方法求解方程 $\cos x = x$ (提示：不动点迭代)

In []:

(3) 制作查询程序：已知在某个存放了偶数的列表中混入了一些奇数，请你写一个程序，找到其中奇数元素在列表中的位置。例如：

N=[4,6,8,46,88,25,48,47,60,59]

奇数元素在列表中的位置为:第5,7,9个元素

In []:

多重(嵌套)for循环

嵌套for循环是指在一个for循环中嵌套另一个for循环，用于迭代遍历多维数组或列表等复杂数据结构。

例如，在Python中，我们可以使用嵌套for循环来遍历一个二维列表：

In [1]:

```
#建议在spyder的调试模式中运行，观察每一个循环变量的变化情况
matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9], [10, 11, 12]]
```

已知在某个二维偶数列表中混入了一些奇数，请你写一个程序，找到其中奇数元素在列表中的位置。

	0	1	2	3
0	22	6	54	354
1	36	34	45	456
2	48	124	54	546
3	79	38	46	44
4	20	214	453	34
5	124	154	34	86
6	786	46	78	34



奇数坐标：
(3,0) (4,2)

In []: