In [1]:
```
output_file = open('hello.txt','w')
print('Hello world', file = output_file)
output_file.close()
```

In [3]:
```
output_file = open('hello.txt','w')
output_file.write('Hello world!\n')
output_file.close()
```

Mode	What it does
a	Create file if does not exist, open file, append contents to the end
w	Create file if does not exist, open file, write contents to the beginning of file
r	Open file, permit reading only

In [4]:
```
input_file = open("test.py", "r")
contents = input_file.read()
print(contents)
```

```
import math

def newspeed(cur, ideal, traction):
    diff = ideal - cur
    if diff < traction:
        return ideal
    else:
        return cur + traction + 0.5*(diff - traction)

actual = 0
total = 0
speed = 0

#rabbit
traction = 0
circum = 2*math.pi*0.5
max_speed = 1000
accel = 0.1

#hare
traction = 100
circum = 2*math.pi*0.1
max_speed = 500
accel = 1

#detailed
traction = 10
circum = 2
max_speed = 100
accel = 0.5

for i in range(10):
    total += actual
    speed = speed + accel * (max_speed - speed)
    ideal = speed * circum
    actual = newspeed(actual, ideal, traction)
    #print('{:.2f}'.format(total))
    print('{:.2f}\t{:.2f}\t{:.2f}\t{:.2f}'.format(total, actual, ideal, speed))
```

Files Using *with*

In [10]:
```
filename = input('Enter filename: ')
with open(filename, 'r') as file:
    for line in file:
        print(line, end = '')
```

```
Enter filename: test.py
import math

def newspeed(cur, ideal, traction):
    diff = ideal - cur
    if diff < traction:
        return ideal
    else:
        return cur + traction + 0.5*(diff - traction)

actual = 0
total = 0
speed = 0

#rabbit
traction = 0
circum = 2*math.pi*0.5
max_speed = 1000
accel = 0.1

#hare
traction = 100
circum = 2*math.pi*0.1
max_speed = 500
accel = 1

#detailed
traction = 10
circum = 2
max_speed = 100
accel = 0.5

for i in range(10):
    total += actual
    speed = speed + accel * (max_speed - speed)
    ideal = speed * circum
    actual = newspeed(actual, ideal, traction)
    #print('{:.2f}'.format(total))
    print('{:.2f}\t{:.2f}\t{:.2f}\t{:.2f}'.format(total, actual, ideal, speed))
```

Exercise 1

Write a program that takes in a filename, then takes in a series of lines of input until a blank line is entered, writing each line to the file with the given name. After the blank line is entered, properly close the file before ending the program.

Exercise 2

Write a program that takes in a filename and string as input. Then print how many times that string appears inside the chosen file. If the file does not exist, continue asking for a filename until one is given that exists. Use your source code file as test input.

Option 1

Task 1. The input file contains two integers, each on a separate line. Output their sum to the output file.

Task 2. The input file contains a single text string, possibly containing spaces. Print this line in reverse order. The line in the input file ends with the end of line character '\n'.

Option 2

Task 1 .The input file contains two integers that can be separated by spaces and line ends. Output their sum to the output file.

Note. Read the entire file into a string variable using the read() method and split it into parts using the split () method.

Task 2. Print all the lines in this file in reverse order. To do this, read the list of all rows using the read () lines method. The last line of the input file must end with the character '\n'.

INPUT

```
Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
```

OUTPUT

```
Complex is better than complicated.
Simple is better than complex.
Explicit is better than implicit.
Beautiful is better than ugly.
```

In [ ]: