C12 File I/O

1. types of mode(can use multiple mode at the same time by using plus sign(+)):
    - r(read):can only read the file content
    - w(write):will empty the file if the file is not empty and it is start at the begining of the file and write over it. If the file is not exist, it will create it.
    - a(append): if the file already has the content, it will start at the end of th econtent and continue wrting without empty the file
    - t(text):default mode when r,w and a
    - b(binary)

```
def main():
    #open the file named lines.txt in read mode only
    f = open('lines.txt', 'r')
    for line in f:
        #rstrip: will return a copy of the string with trailing whitespace removed.
        print(line.rstrip())

if __name__ == '__main__': main()
```

Text file

1. the file extension is .txt
2. text file is a default mode when user read, write and append the file

```
def main():
    infile = open('lines.txt', 'rt')
    outfile = open('lines-copy.txt', 'wt')
    for line in infile:
        #first method
        outfile.writelines(line)
        #second method
        #using print method is good to use because able to strips these line endings (able to format the line endings that foll
        print(line.rstrip(), file=outfile)
        print('.', end='', flush=True)
    outfile.close()
    print('\ndone.')

if __name__ == '__main__': main()
```

Binary file

```
#copy a image file
def main():
    infile = open('berlin.jpg', 'rb')
    outfile = open('berlin-copy.jpg', 'wb')
    while True:
        #we don't want read the whole file all together because we don't know how much memory is available in our system
        #by using 10240, we can limit the size of the file when we read and copy it
        buf = infile.read(10240) #10kbytes(very small in the system)
        if buf:
            outfile.write(buf)
            print('.', end='', flush=True)
        else: break
    outfile.close()
    print('\ndone.')

if __name__ == '__main__': main()
```