Introduction

By £ily Lin
The Banana Chariot

All formula include (in order)

Annually compound	$A = P(1+r)^n$
Interest compound k times per year	$A = P\left(1+\frac{r}{k}\right)^{kt}$
Interest compound continuously	$A = Pe^{rt}$
Annuity future value	$FV = R\frac{(1+i)^n - 1}{i}$
Annuity present value	$PV = R\frac{1-(1+i)^{-n}}{i}$

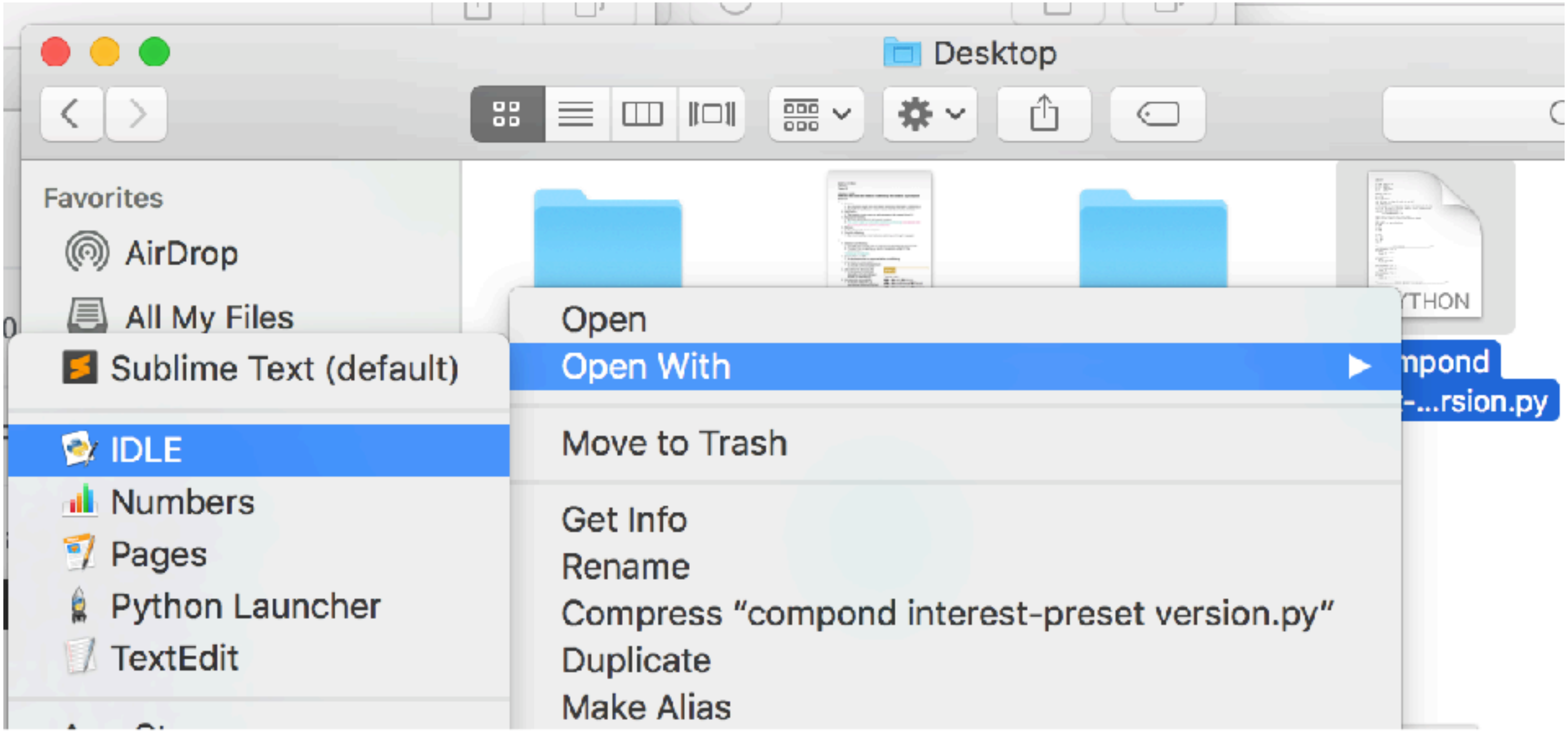
Please also read the things in the file

It’s crude but still functional

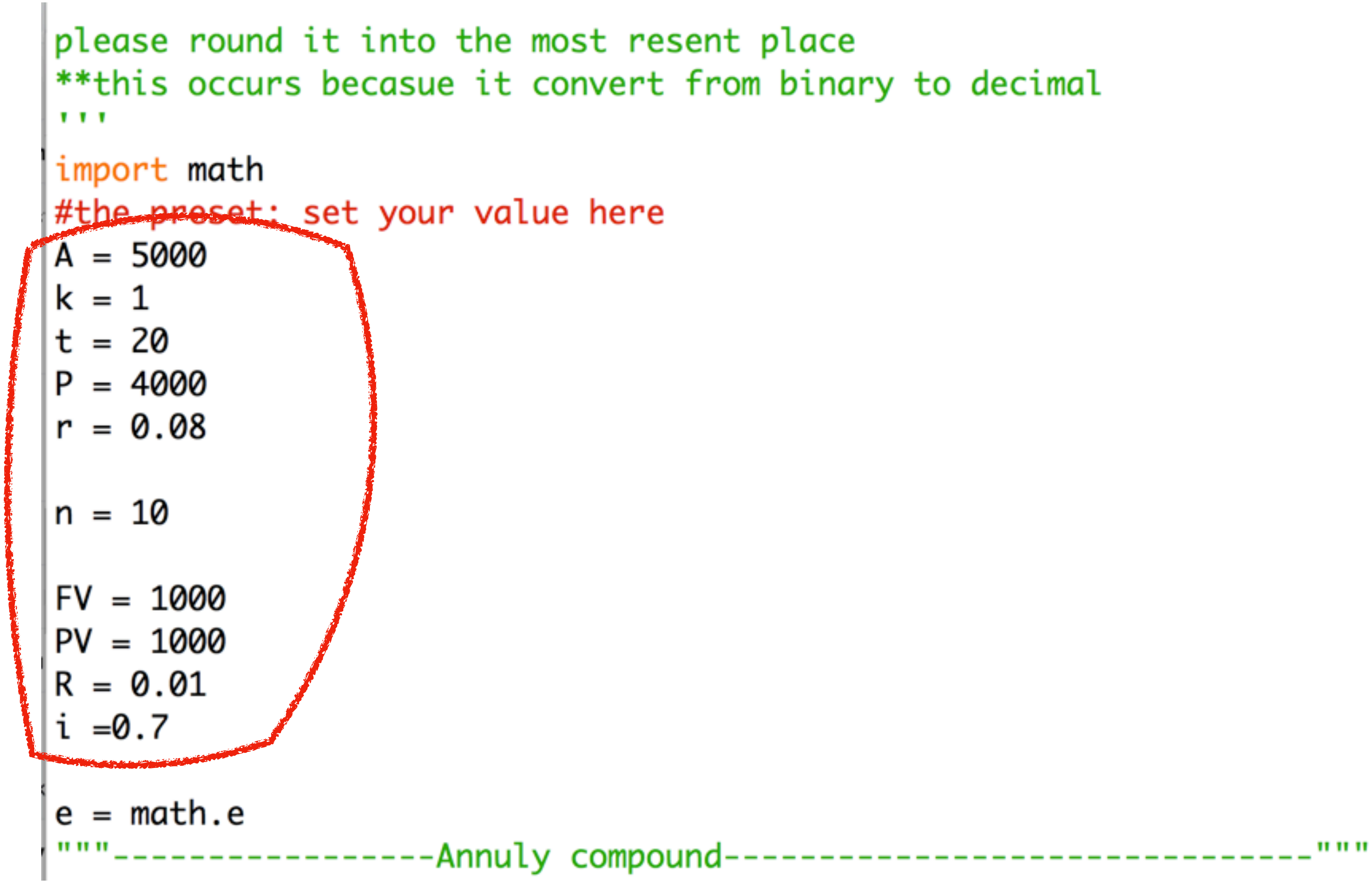
Java is more recommended but I haven’t learn it yet

Preset version

1. Open it with IDLE

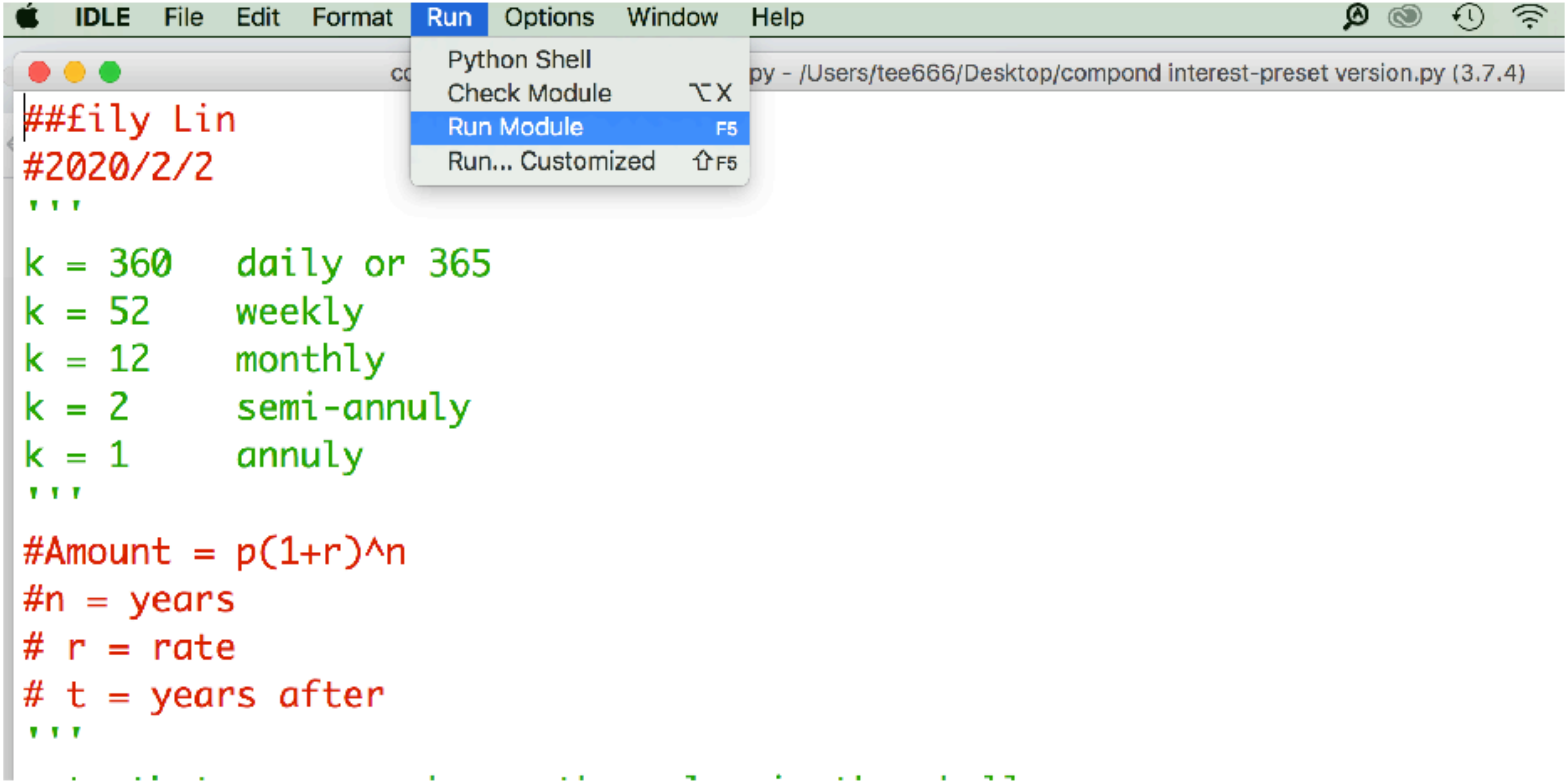


2. Set the number of each Variable *don’t leave it blank



```
please round it into the most resent place
**this occurs becasue it convert from binary to decimal
'''
import math
#the preset: set your value here
A = 5000
k = 1
t = 20
P = 4000
r = 0.08

n = 10

FV = 1000
PV = 1000
R = 0.01
i =0.7

e = math.e
"""-----------------Annuly compound----------------------------------"""
```

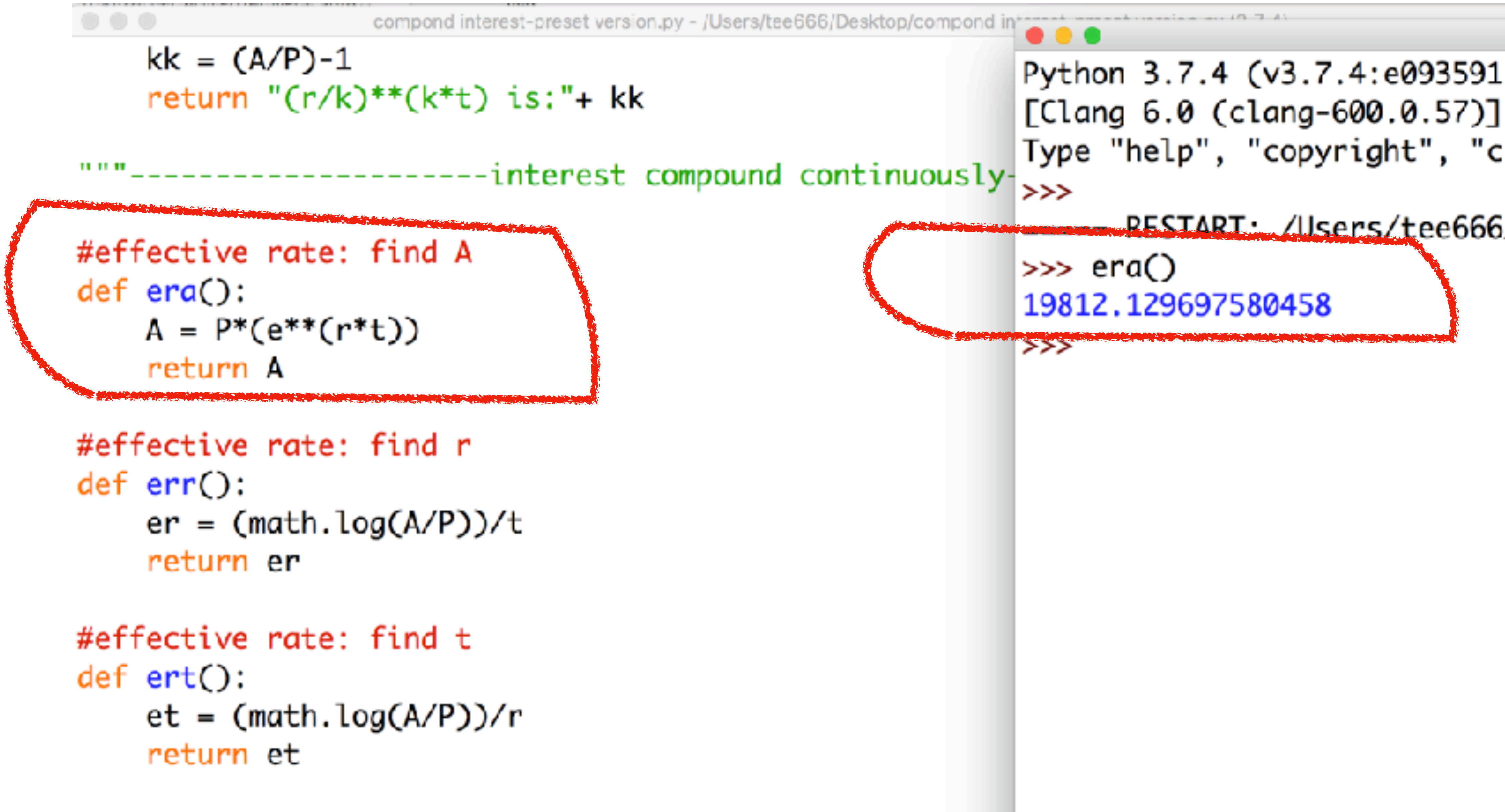
3. Run Module ,or press F5



```
##fily Lin
#2020/2/2
'''
k = 360   daily or 365
k = 52    weekly
k = 12    monthly
k = 2     semi-annuly
k = 1     annuly
'''
#Amount = p(1+r)^n
#n = years
# r = rate
# t = years after
'''
```

4. Find the function and variable you want to find Type the things after def, then press return button *don’t include “:”, no space before the function

Example :



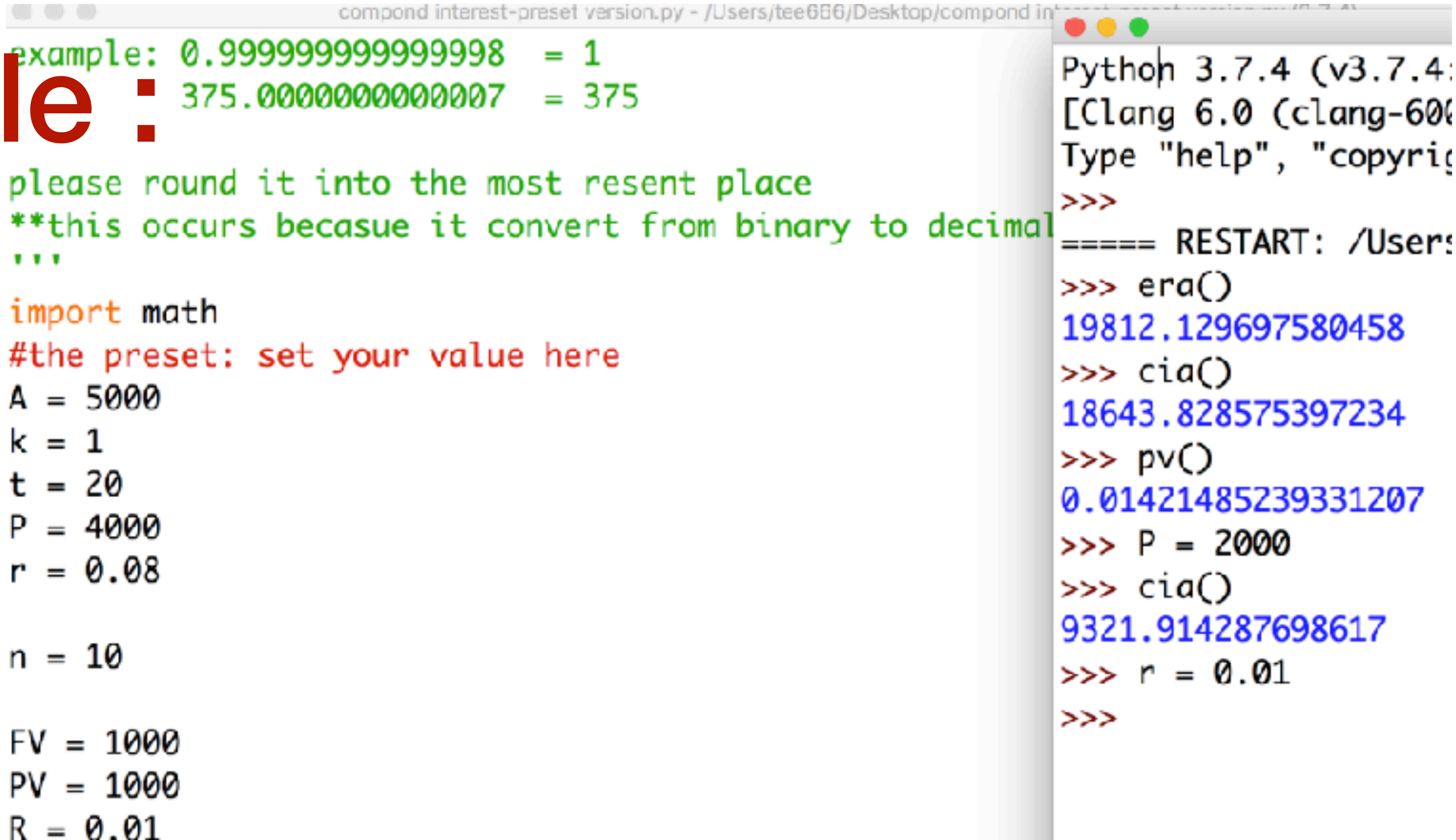
compond interest-preset version.py - /Users/tee666/Desktop/compond in

```
    kk = (A/P)-1
    return "(r/k)**(k*t) is:"+ kk

"""----------------------interest compound continuously-

#effective rate: find A
def era():
    A = P*(e**(r*t))
    return A

#effective rate: find r
def err():
    er = (math.log(A/P))/t
    return er

#effective rate: find t
def ert():
    et = (math.log(A/P))/r
    return et
```

```
Python 3.7.4 (v3.7.4:e093591
[Clang 6.0 (clang-600.0.57)]
Type "help", "copyright", "c
>>>
====== RESTART: /Users/tee666
>>> era()
19812.129697580458
>>>
```

5. You can put other functions or change the preset value by type after >>>
Then press return button

Example :



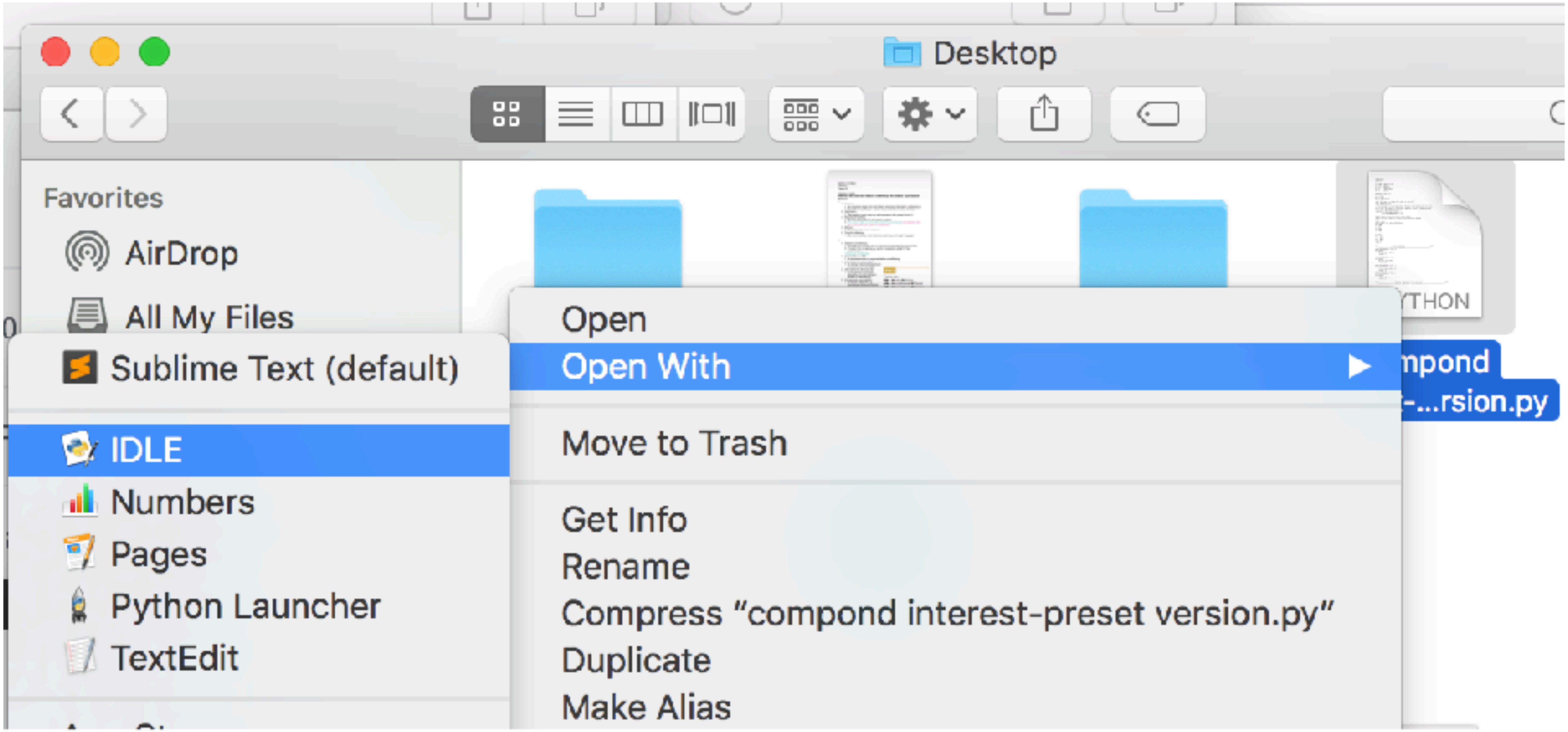
compond interest-preset version.py - /Users/tee666/Desktop/compond in

```
example: 0.9999999999999998  = 1
         375.0000000000007   = 375

please round it into the most resent place
**this occurs becasue it convert from binary to decimal
'''
import math
#the preset: set your value here
A = 5000
k = 1
t = 20
P = 4000
r = 0.08

n = 10

FV = 1000
PV = 1000
R = 0.01
```

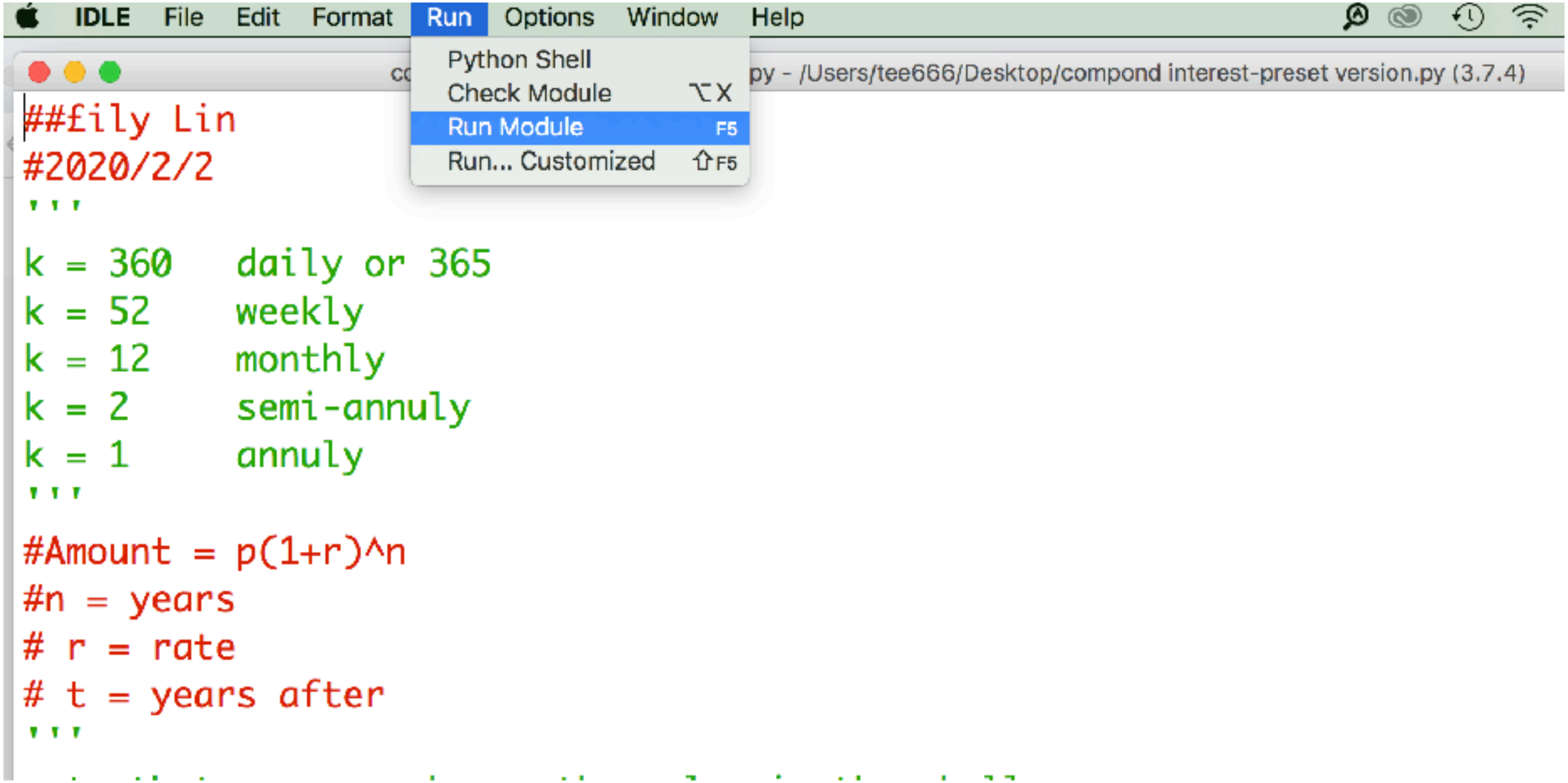
```
Python 3.7.4 (v3.7.4
[Clang 6.0 (clang-600
Type "help", "copyri
>>>
===== RESTART: /User
>>> era()
19812.129697580458
>>> cia()
18643.828575397234
>>> pv()
0.014214852393312O7
>>> P = 2000
>>> cia()
9321.914287698617
>>> r = 0.01
>>>
```

Type variable version

1. Open it with IDLE



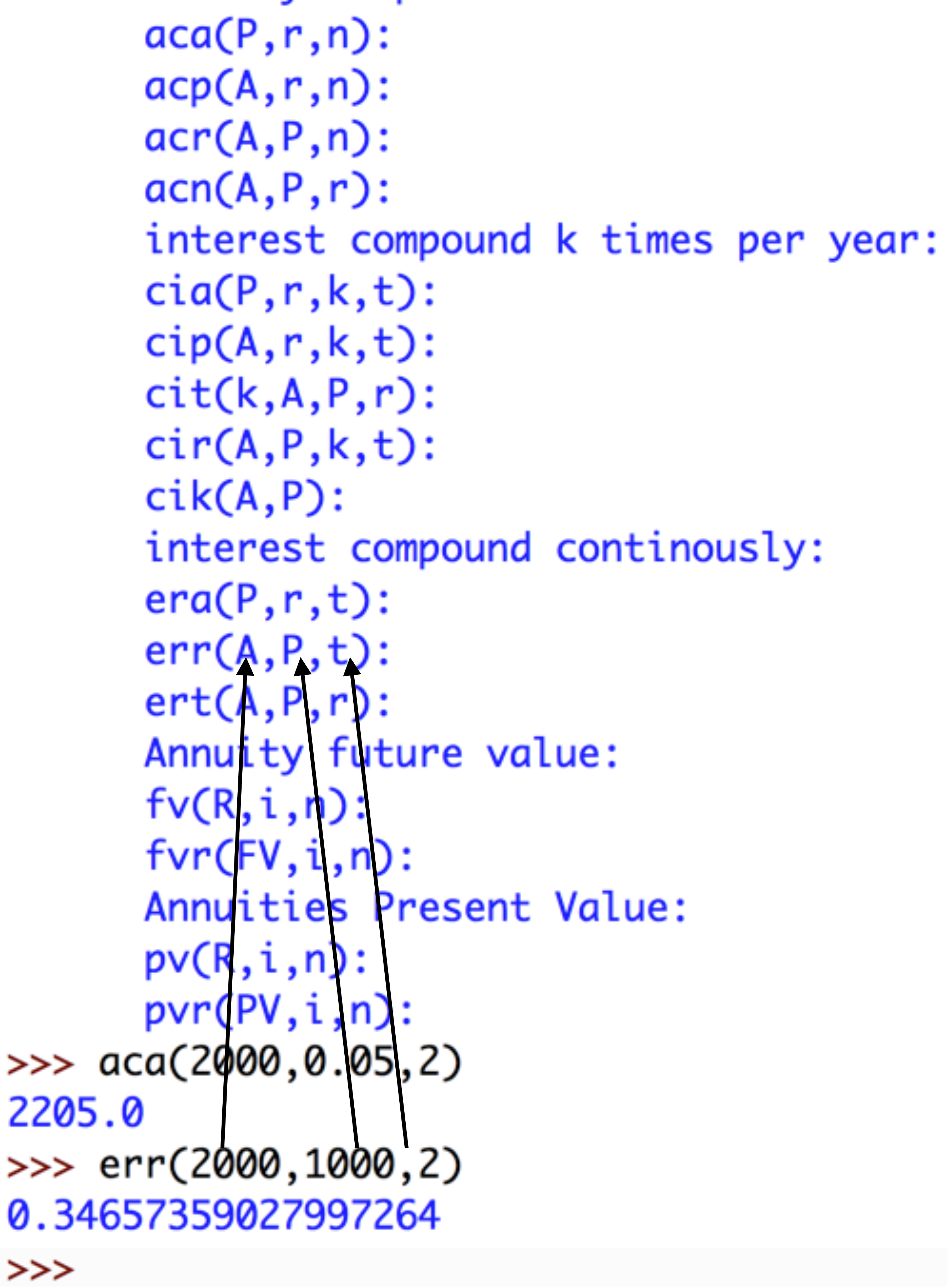
2. Run Module ,or press F5



```
##fily Lin
#2020/2/2
'''
k = 360   daily or 365
k = 52    weekly
k = 12    monthly
k = 2     semi-annuly
k = 1     annuly
'''
#Amount = p(1+r)^n
#n = years
# r = rate
# t = years after
'''
```

3. Find the function and variable you want to find Type the things after def, and put the variables in the parentheses then press return button

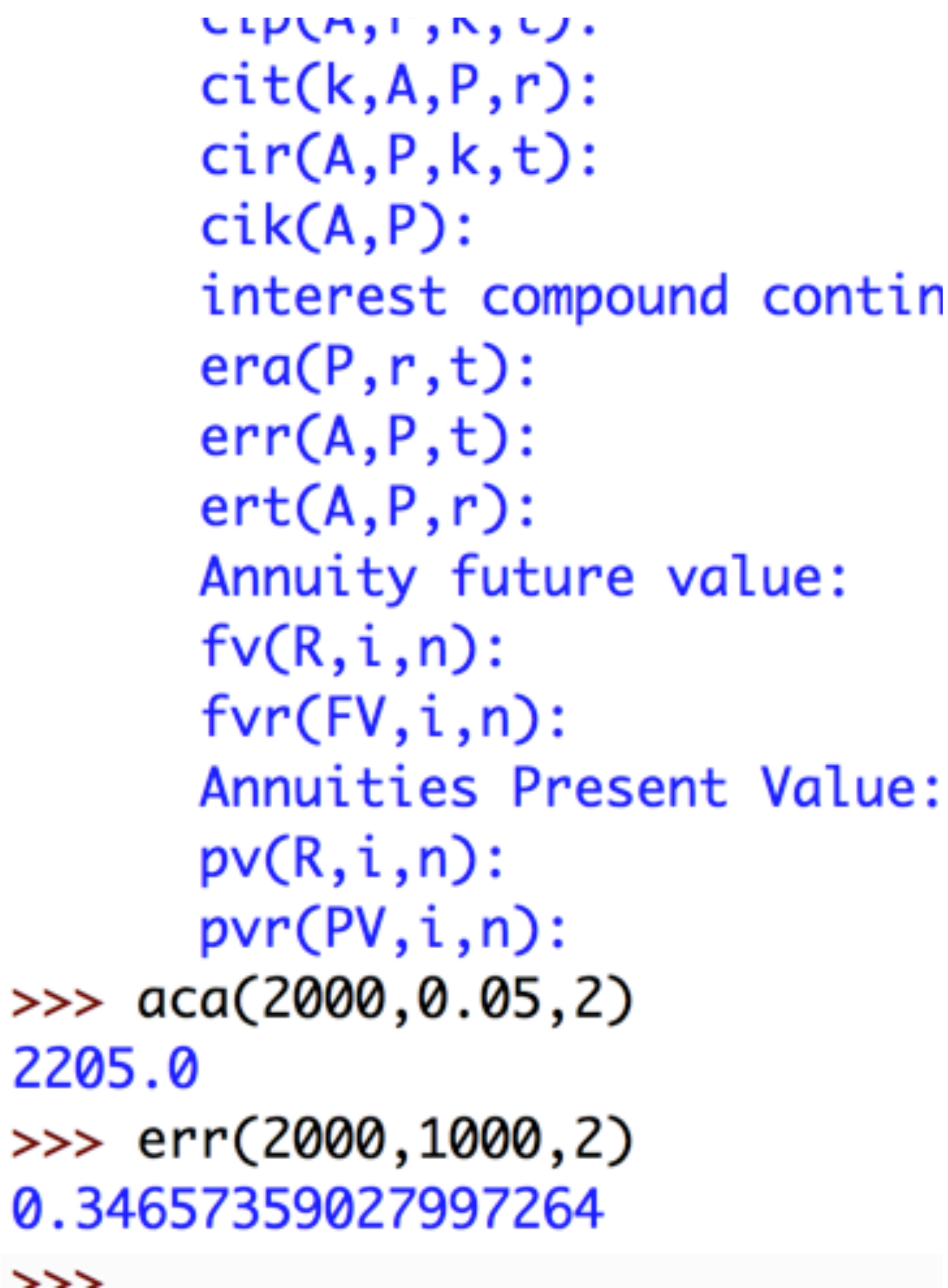
*don’t include “:”



```
        aca(P,r,n):
        acp(A,r,n):
        acr(A,P,n):
        acn(A,P,r):
        interest compound k times per year:
        cia(P,r,k,t):
        cip(A,r,k,t):
        cit(k,A,P,r):
        cir(A,P,k,t):
        cik(A,P):
        interest compound continously:
        era(P,r,t):
        err(A,P,t):
        ert(A,P,r):
        Annuity future value:
        fv(R,i,n):
        fvr(FV,i,n):
        Annuities Present Value:
        pv(R,i,n):
        pvr(PV,i,n):
>>> aca(2000,0.05,2)
2205.0
>>> err(2000,1000,2)
0.34657359027997264
>>>
```

5. You can put other functions by type after >>> Then press return button

Example :



```
        cip(A,r,k,t):
        cit(k,A,P,r):
        cir(A,P,k,t):
        cik(A,P):
        interest compound contin
        era(P,r,t):
        err(A,P,t):
        ert(A,P,r):
        Annuity future value:
        fv(R,i,n):
        fvr(FV,i,n):
        Annuities Present Value:
        pv(R,i,n):
        pvr(PV,i,n):
>>> aca(2000,0.05,2)
2205.0
>>> err(2000,1000,2)
0.34657359027997264
>>>
```