Fibonacci+Golden-Ratio

September 28, 2020

1 Python Program to Calculate Fibonacci Numbers' & "Golden Ratio'

Powered by: Dr. Hermann Völlinger, DHBW Stuttgart(Germany); August 2020

See Wikipedia: https://en.wikipedia.org/wiki/Fibonacci_number:

1.1 The Fibonacci Numbers

Fibonacci numbers are named after Italian mathematician Leonardo of Pisa, later known as Fibonacci. In his 1202 book Liber Abaci, Fibonacci introduced the sequence to Western European mathematics,[5] although the sequence had been described earlier in Indian mathematics,[6][7][8] as early as 200 BC in work by Pingala on enumerating possible patterns of Sanskrit poetry formed from syllables of two lengths.

Fibonacci numbers appear unexpectedly often in mathematics, so much so that there is an entire journal dedicated to their study, the Fibonacci Quarterly. Applications of Fibonacci numbers include computer algorithms such as the Fibonacci search technique and the Fibonacci heap data structure, and graphs called Fibonacci cubes used for interconnecting parallel and distributed systems. They also appear in biological settings, such as branching in trees, the arrangement of leaves on a stem, the fruit sprouts of a pineapple, the flowering of an artichoke, an uncurling fern, and the arrangement of a pine cone's bracts.

[1]:

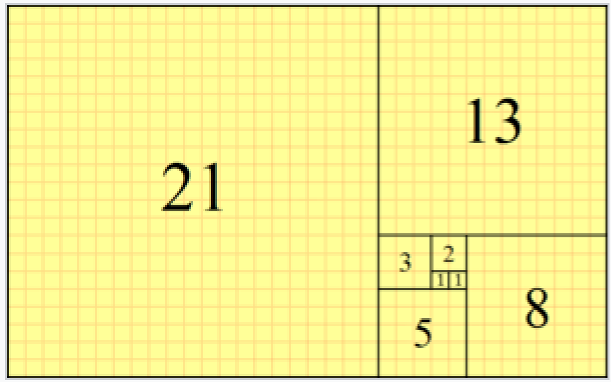
```python
print ("*****************************************************************")
print ("***'Fibonacci Numbers' are shown as squares whose side lengths *****")
print ("*** are successive Fibonacci numbers: 1, 1, 2, 3, 5, 8, 13, 21.. ***")
print ("*****************************************************************")

from IPython.display import Image

Image('fibonacci-nb.png')
```

```
*****************************************************************
***'Fibonacci Numbers' are shown as squares whose side lengths *****
*** are successive Fibonacci numbers: 1, 1, 2, 3, 5, 8, 13, 21.. ***
*****************************************************************
```

[1]:



A tiling with squares whose side lengths are successive Fibonacci numbers: 1, 1, 2, 3, 5, 8, 13 and 21.

[ ]:
```
## 'Fibonacci Spiral'

An approximation of the golden spiral created by drawing circular arcs connecting the opposite corners of squares in
the Fibonacci tiling; (see preceding image) Fibonacci numbers are strongly related to the golden ratio: Binet's
formula expresses the nth Fibonacci number in terms of n and the golden ratio, and implies that the ratio of two
consecutive Fibonacci numbers tends to the golden ratio as n increases.

A tiling with squares whose side lengths are successive Fibonacci numbers: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144....
```

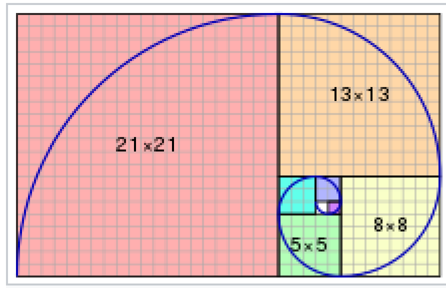
[2]:
```
print ("****************************************************************")
print ("**'Fibonacci Spiral'= drawing a spiral into the above squares:**")
print ("****************************************************************")

from IPython.display import Image

Image('fibonacci-sp.png')
```

```
****************************************************************
**'Fibonacci Spiral'= drawing a spiral into the above squares:**
****************************************************************
```

[2]:



[3]:
```python
# First Part: 'Fibonacci Numbers'
print ("*********************************************************************")
print ("****The program calculates the Fibonacci Numbers & Golden Ratio******")
print ("*********************************************************************")
print (" ********** First Part: 'Fibonacci Numbers'**************************")
print ("*********************************************************************")
```

```
*********************************************************************
****The program calculates the Fibonacci Numbers & Golden Ratio******
*********************************************************************
 ********** First Part: 'Fibonacci Numbers'**************************
*********************************************************************
```

[4]:
```python
# Start calculation of Fibo

def recur_fibo(n):
   if n <= 1:
       return n
   else:
       return(recur_fibo(n-1) + recur_fibo(n-2))

# End calulation of Fibo
```

[5]:
```python
# Take the input from the user + print the Fibo numbers

nterms = int(input("Hello Hermann. How many terms you want? "))

# Check if the number of terms is valid

if nterms <= 0:
   print("Plese enter a positive integer")
else:
   print("The Fibonacci numbers until",nterms, "are:")
   for i in range(nterms):
       print(recur_fibo(i))
```

```python
# End calculation and print of Fibonacci numbers
# print '**** End of First Part ********'

print ("**** End of List of Fibo Numbers****")
print ("*******************************************")
print ("************* End of First Part ***********")
print ("********************************************")
```

```
Hello Hermann. How many terms you want? 26
The Fibonacci numbers until 26 are:
0
1
1
2
3
5
8
13
21
34
55
89
144
233
377
610
987
1597
2584
4181
6765
10946
17711
28657
46368
75025
**** End of List of Fibo Numbers****
*******************************************
************* End of First Part ***********
********************************************
```

1.2 Second Part: 'Golden Ration'

In mathematics, two quantities are in the golden ratio if their ratio is the same as the ratio of their sum to the larger of the two quantities. The figure iunder this description illustrates the geometric relationship.

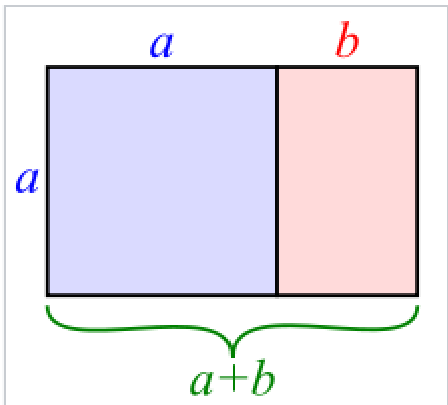
[6]:
```python
print ("********************************************************************")
print ("**************** Second Part: 'Golden Ratio ************************")
print ("*******************************************************************")
print ("********************************************************************")

from IPython.display import Image

Image('golden-ratio.png')
```

```
********************************************************************
**************** Second Part: 'Golden Ratio ************************
*******************************************************************
********************************************************************
```

[6]:



A golden rectangle with longer side a and shorter side b, when placed adjacent to a square with sides of length a, will produce a similar golden rectangle with longer side $a + b$ and shorter side a. This illustrates the relationship $\frac{a+b}{a} = \frac{a}{b} \equiv \varphi$.

It is an irrational number that is a solution to the quadratic equation with a value of= 1.6180339887 The golden ratio is also called the golden mean or golden section (Latin: sectio aurea).[4][5] Other names include extreme and mean ratio,[6] medial section, divine proportion (Latin: proportio divina),[7] divine section (Latin: sectio divina), golden proportion, golden cut,[8] and golden number.[9][10][11]

Mathematicians since Euclid have studied the properties of the golden ratio, including its appearance in the dimensions of a regular pentagon and in a golden rectangle, which may be cut into a square and a smaller rectangle with the same aspect ratio. The golden ratio has also been used to analyze the proportions of natural objects as well as man-made systems such as financial markets, in some cases based on dubious fits to data.[12] The golden ratio appears in some patterns in nature, including the spiral arrangement of leaves and other plant parts.

Some twentieth-century artists and architects, including Le Corbusier and Salvador Dalí, have proportioned their works to approximate the golden ratio, believing this to be aesthetically pleasing. These often appear in the form of the golden rectangle, in which the ratio of the longer side to the shorter is the golden ratio.

[7]:
```python
# second part: 'Golden Ratio'
print("Hello Hermann. Do you want to see the the 'Golden Ratio' numbers until␣
 ↪this term? ")
go = int(input("Then type '1'"))
if go == 1:
   print ("***************************************************************")
   print ("*** The program calculates 'Golden Ratio'= (Fibo(i+1)/Fibo(i)) ***")
   # Start printing 'Golden Ratio'
   print ("***************************************************************")
   print ("Please check the values for: 'Golden-Ratio'~1.61803398875")
# Start printing 'Golden Ratio'
   for i in range(nterms):
       print(recur_fibo(i+2)/recur_fibo(i+1))
else:
   print ("no 'Golden Ratio' numbers are calculated")

## End of Second Part
```

```
Hello Hermann. Do you want to see the the 'Golden Ratio' numbers until this
term?
Then type '1'1
*****************************************************************
*** The program calculates 'Golden Ratio'= (Fibo(i+1)/Fibo(i)) ***
*****************************************************************
Please check the values for: 'Golden-Ratio'~1.61803398875
1.0
2.0
1.5
1.6666666666666667
1.6
1.625
1.6153846153846154
1.619047619047619
1.6176470588235294
1.6181818181818182
1.6179775280898876
1.6180555555555556
1.6180257510729614
```

```
1.6180371352785146
1.618032786885246
1.618034447821682
1.6180338134001253
1.618034055727554
1.6180339631667064
1.6180339985218033
1.618033985017358
1.6180339901755971
1.618033988205325
1.618033988957902
1.6180339886704431
1.6180339887802426
```

[8]:

```python
# print current date and time
import time
print("date",time.strftime("%d.%m.%Y %H:%M:%S"))
print ("end")
```

```
date 16.08.2020 14:05:52
end
```