Fibonacci+Golden-Ratio

August 13, 2020

1 Python Progarm to Calcualte 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.

1.2 '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….

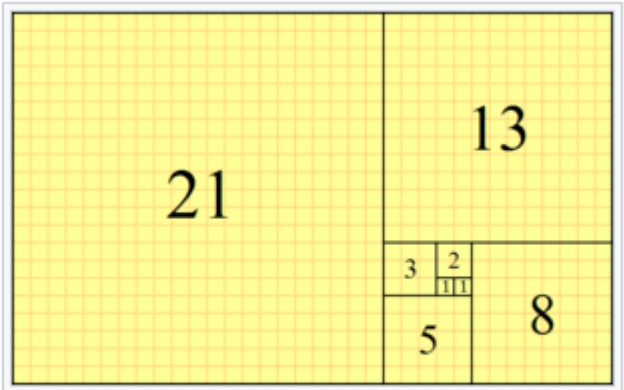
[1]:
```
print ("*********************************************************************")
print ("***'Fibonacci Numbers' are shown as squares whose side lengths *****")
print ("*** are successive Fibonacci numbers: 1, 1, 2, 3, 5, 8, 13, 21.. ***")
print ("*********************************************************************")

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.

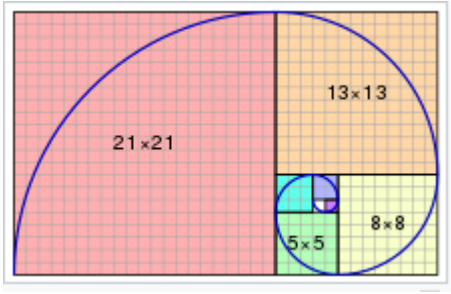
[2]:
```
print ("**********************************************************************")
print ("**'Fibonacci Spiral'= drawing a Spiral into the above squares:**")
print ("**********************************************************************")

from IPython.display import Image

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

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

[2]:



[3]:
```
# 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]:
```
# 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]:
```
# 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))

# End calculation and print of Fibonacci numbers

print ("****end of list of Fibo Numbers****")
```

```
Hello Hermann. How many terms you want? 30
The Fibonacci numbers until 30 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
121393
196418
317811
514229
****end of list of Fibo Numbers****
```

2 End of First Part

3 Second Part: 'Golden Ration'

[6]:
```
# 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
1.618033988738303
1.6180339887543225
1.6180339887482036
1.6180339887505408
```

[7]:
```
# print current date and time
import time
print("date",time.strftime("%d.%m.%Y %H:%M:%S"))
print ("end")
```

```
date 13.08.2020 22:53:32
end
```