DATA SCIENCE BATCH-2023

Büşra Osmanoğlu

OGTIPDSBO037

The following commands were written and run in the Python Console in PyCharm.

1. Adding Two Numbers

```
>>> 3+5
8
```

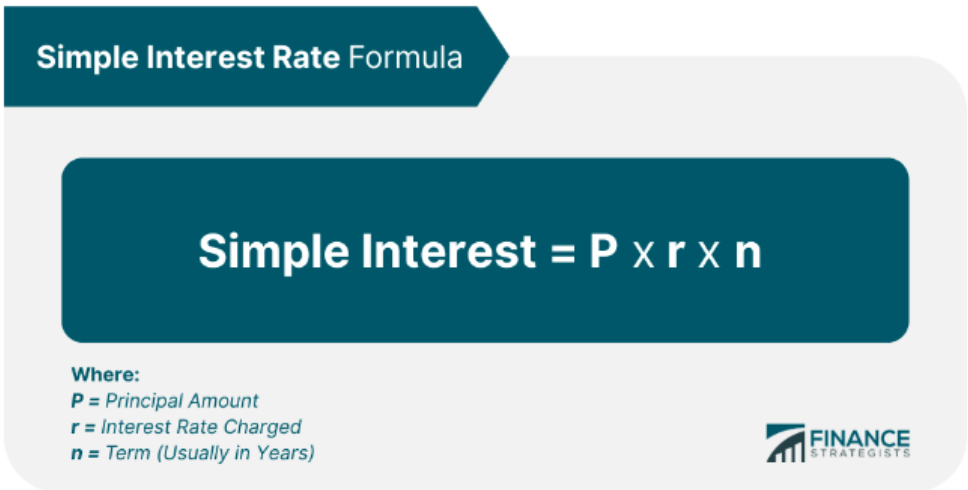
2. Maximum of Two Numbers

```
>>> max(3,5)
5
```

3. Factorial of a Number

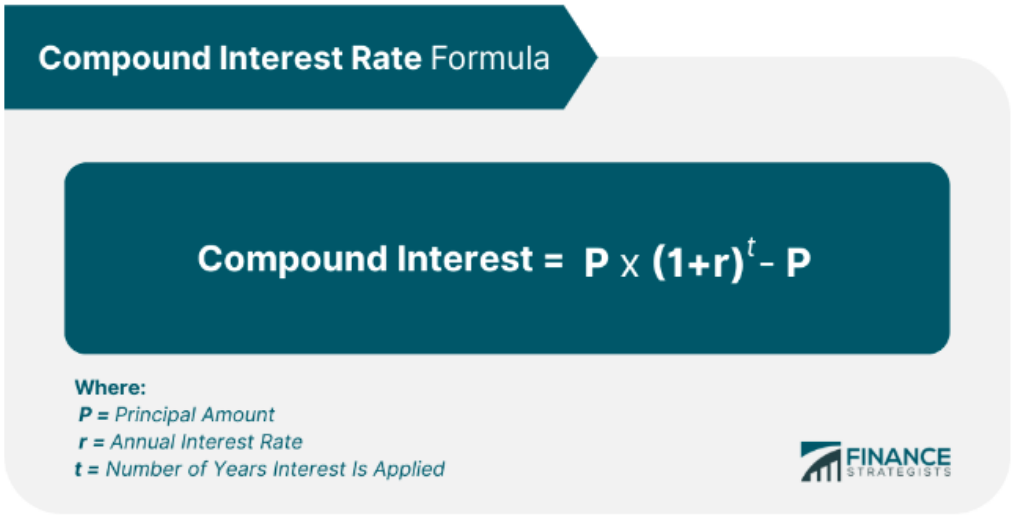
```
>>> import math
>>> math.factorial(5)
120
```

4. Function for Simple Interest



```
>>> def SimpleInterest(P,r,n):
...     return P*r*n
...
>>> SimpleInterest(1000, 0.05, 2)
100.0
```

5. Function for Compound Interest




```
>>> def CompoundInterest(P,r,t):
...     return P*(1+r)**t-P
...
>>> CompoundInterest(1000,0.05,2)
102.5
```

6. Check Armstrong Number

Examples: $153 = 1^3 + 5^3 + 3^3$

$93084 = 9^5 + 3^5 + 0^5 + 8^5 + 4^5$

```
>>> def isArmstrongNumber(n):
...     digits = list(str(n))
...     return True if sum([int(i)**len(digits) for i in digits]) == n else False
...
>>> isArmstrongNumber(153)
True
```

7. Function for Area of a Circle

```
>>> def AreaofCircle(r):
...     return round(math.pi*r**2,2)
...
>>> AreaofCircle(1)
3.14
```

- Positive Divisors Except 1 and n

```
>>> def PositiveDivisorsE1n(n):
...     return [i for i in range(2,n) if n%i == 0] if n > 1 else None
...
>>> PositiveDivisorsE1n(12)
[2, 3, 4, 6]
```

8. All Prime Numbers in an Interval by using *PositiveDivisorsE1n* function above

```
>>> def PrimesBetween(b,e):
...     return [i for i in range(b,e+1) if PositiveDivisorsE1n(i) == []]
...
>>> PrimesBetween(-20,20)
[2, 3, 5, 7, 11, 13, 17, 19]
```

9. Check If a Number Prime or Not by using *PositiveDivisorsE1n* function

```
>>> def isPrime(n):
...     return True if PositiveDivisorsE1n(n) == [] else False
...
>>> isPrime(1)
False
```

10. n-th Fibonacci Number

```
>>> def fibonacci(n):
...     return 1 if n==1 or n==2 else fibonacci(n-1)+fibonacci(n-2)
...
>>> fibonacci(7)
13
```

11. Function Checking If a Number is a Fibonacci Number by using *fibonacci* function (10)

```
>>> def isFibonacci(x):
...     n=1
...     while fibonacci(n) < x:
...         n += 1
...
...     if fibonacci(n) == x:
...         return True
...
...     return False
...
>>> isFibonacci(7)
False
>>> isFibonacci(13)
True
```

12. mth Multiple of a number in Fibonacci Series by using *fibonacci* function (10)

```
>>> def FibonacciMultipleby(n,m):
...     return fibonacci(n)*m
...
>>> FibonacciMultipleby(3,5)
10
```

13. ASCII Value of a Character

```
>>> ord("b")
98
```

14. Sum Of Squares of First n Natural Numbers

```
>>> def SumOfSquares(n):
...     return sum([i**2 for i in range(1,n+1)])
...
>>> SumOfSquares(3)
14
```

15. Cube Some of First n Natural Numbers

```
>>> def SumOfCubes(n):
...     return sum([i**3 for i in range(1,n+1)])
...
>>> SumOfCubes(3)
36
```

SOURCES

- https://www.financestrategists.com/banking/interest-rate/?gclid=Cj0KCQjw6cKiBhD5ARIsAKXUdyZKeqX5hhJtI9egJK2w90nP0wkS7v_PEpbmknFuhqAqX5-bkiuTjBoaAgucEALw_wcB