kel

January 23, 2023

1 kel

Use the "Run" button to execute the code.

[5]:
```
print('Hello World')
```

```
Hello World
```

[8]:
```
def loan_emi(amount):
    emi=amount/12
    print("loan amount per month is $ {}".format(emi))
```

[15]:
```
loan_emi(1200000,5)
```

```
loan amount per month is $ 240000.0
```

[10]:
```
def loan_emi(amount,duration):# define a function which has local variables␣
 ↪others are global
    emi=amount/duration
    print("loan amount per month is $ {}".format(emi))
```

[12]:
```
loan_emi(1200000,12)
```

```
loan amount per month is $ 100000.0
```

[98]:
```
def loan_emi(amount,duration):
    emi=amount/duration
    return(emi)
```

[99]:
```
emi1=loan_emi(1260000*1.1,12)
emi2=loan_emi(1260000*1.08,12)
print(emi1)
print(emi2)
if emi1>emi2:
    print("emi2 is better than emi1 and amount is{}".format(emi2))
if emi1<emi2:
     print("emi1 is better than emi2 and amount is{}".format(emi1))
```

```
115500.0
113400.0
emi2 is better than emi1 and amount is113400.0
```

[95]:
```
def loan_emis(amount,duration,downpayment=0):
    emis=(amount-downpayment)/duration
    print("loan amount per month is ${}".format(emis))
    math.ceil(emis)
    return(emis)
```

[96]:
```
emia=loan_emis(1200000*1.1,8,300000)
emib=loan_emis(1200000*1.08,12)
print(emia)
print(emib)
```

```
loan amount per month is $127500.0
loan amount per month is $108000.0
127500.0
108000.0
```

[60]:
```
def loan_paid(loan_amount,rate,time,downpayment=0):
    loan_amount=amount-downpayment
    emi =loan_amount*rate*((1+rate)**time)/(((1+rate)**time)-1)
    return(emi)
```

[61]:
```
b=loan_paid(1260000,0.1/12,10*12,300000)
print(b)
a=loan_paid(1260000,0.1/12,10*12)
print(a)
```

```
11893.566319358573
15858.088425811431
```

[54]:
```
a=loan_emi(1260000,0.1/12,10*12)
print(a)
```

```
16650.992847102003
```

[88]:
```
def loan_paid(amount,loan_amount,rate,time,downpayment=0):
    loan_amount=amount-downpayment
    loan_paid =loan_amount*rate*((1+rate)**time)/(((1+rate)**time)-1)
io=loan_paid(
       amount=1260000,
       loan_amount=amount-0,
       time=10*12,
       rate=0.1/12,
       downpayment=0
)
```

[89]:
```
print(io)
```

```
None
```

[94]:
```
import math
help(math.ceil)
math.ceil(1.2)
```

```
Help on built-in function ceil in module math:

ceil(x, /)
    Return the ceiling of x as an Integral.

    This is the smallest integer >= x.
```

[94]:
```
2
```

[142]:
```
cost=800000
loan_amount=cost-downpayment
duration=6*12
rate=0.07/12
downpayment=0.25*800000
def loan_emi(loan_amount,rate,duration,downpayment):
    loan_amount=cost-downpayment
    emi =loan_amount*rate*((1+rate)**duration)/(((1+rate)**duration)-1)
    math.ceil(n)
    return(emi)
```

[143]:
```
n=loan_emi(loan_amount,rate,duration,downpayment)
print(math.ceil(n))
```

```
10230
```

[152]:
```
price=60000
rate=0.12/12
duration=1*12
def car_emi(price,duration,rate):
    emi =price*rate*((1+rate)**duration)/(((1+rate)**duration)-1)
    return(emi)
```

[153]:
```
y=car_emi(price,duration,rate)
print(math.ceil(y))
```

```
5331
```

[135]:
```
math.ceil(1.2)
```

[135]: 2

[173]:
```
total=100000
rate=0.09/12
duration=10*12
def loan_emi(amount,rate,duration):
    emi =total*rate*((1+rate)**duration)/(((1+rate)**duration)-1)
    return(emi)
```

[174]:
```
with_interest=loan_emi(amount,rate,duration)
print(math.ceil(with_interest))
```

```
1267
```

[164]:
```
g=100000/(10*12)
```

[172]:
```
print(math.ceil(g))
```

```
834
```

[176]:
```
print(math.ceil((with_interest-g)*10))
```

```
4335
```

[184]:
```
try:
    print("computing the result")#used when error is involved
    h=6/0
    print(h)
    print("computation was successful")
except ZeroDivisionError:
    print("failed to compute")
    result=None
    print(result)
```

```
computing the result
failed to compute
None
```

[202]:
```
price=100000
rate=0.0/12
duration=10*12
def car_emi(price,duration,rate):
    try:
        emi=price*rate*((1+rate)**duration)/(((1+rate)**duration)-1)
    except ZeroDivisionError:
          emi=price/duration
    return(emi)
```

[204]:
```
d=car_emi(price,duration,rate)
print(math.ceil(d))
```

```
834
```

[205]:
```
print(math.ceil((with_interest-d)*10))
```

```
4335
```

[ ]:
```

```