Loan Amortization Schedule

In [37]:
```
#Load the necessary libraries
import pandas as pd
import numpy as np
import numpy_financial as npf
import matplotlib.pyplot as plt
import seaborn as sns
```

In [38]:
```
def amortization_schedule(InterestRate,LoanAmount,years):
    """
    Document String

    This function takes in the following inputs as paraments and return a table:
    InterestRate = As a float or Integer
    Loan Amount = The Total Loan Amount
    Years = The Loan term

    """
    # Parameter Definition
    #the loan amount is converted to negative because money is going out
    LoanAmount = -(LoanAmount)

    # The annual interest rate is converted to monthly by dividing by 12
    InterestRate=(InterestRate / 100) / 12

    #We multiply the years by 12 to have the total number of period
    periods = years * 12

    #Create an Array
    nper = np.arange(1,periods+1)

    #Build the Amortization Schedule
    #Interest Payment
    interest = np.round(npf.ipmt(InterestRate,nper,periods,LoanAmount),2)

    #Principal payment
    principal = np.round(npf.ppmt(InterestRate,nper,periods,LoanAmount),2)

    #Join Data
    df= list(zip(nper,interest,principal))
    df=pd.DataFrame(df,columns=["Period","Interest","Principal"])

    #Monthly Loan payment
    df["Payment"] = df["Interest"] + df["Principal"]

    #df["test"] = pv

    #Cummulative payment
    df["Total Principal Paid"] = df["Principal"].cumsum()
    df["Total Interest Paid"] = df["Interest"].cumsum()
    df["Total Payment Paid"] = df["Payment"].cumsum()
    #Reverse values since we are paying down the balance
    #df["Outstanding_Bal"] = df["Outstanding_Bal"].values[::-1]

    return df
```

30 years Amortization

A 30-year fixed rate at 4%. The loan will amortize over 30 years.

In [39]:
```
Loan_Amount = 1_000_000
Interest_Rate = 4
Loan_Term = 30
```

In [40]:
```
df = amortization_schedule(Interest_Rate,Loan_Amount,Loan_Term)
df
```

Out[40]:

	Period	Interest	Principal	Payment	Total Principal Paid	Total Interest Paid	Total Payment Paid
0	1	3333.33	1440.82	4774.15	1440.82	3333.33	4774.15
1	2	3328.53	1445.62	4774.15	2886.44	6661.86	9548.30
2	3	3323.71	1450.44	4774.15	4336.88	9985.57	14322.45
3	4	3318.88	1455.28	4774.16	5792.16	13304.45	19096.61
4	5	3314.03	1460.13	4774.16	7252.29	16618.48	23870.77
...	...	...	...	...	...	...	...
355	356	78.78	4695.37	4774.15	981061.48	718537.01	1699598.49
356	357	63.13	4711.02	4774.15	985772.50	718600.14	1704372.64
357	358	47.43	4726.73	4774.16	990499.23	718647.57	1709146.80
358	359	31.67	4742.48	4774.15	995241.71	718679.24	1713920.95
359	360	15.86	4758.29	4774.15	1000000.00	718695.10	1718695.10

360 rows × 7 columns

Total Interest Paid = 718,695

Total Payment Made = 1,718,695

20 years Amortization

A 20-year fixed rate at 2.5%. The loan will amortize over 20 years.

In [41]:
```
Loan_Amount = 1_000_000
Interest_Rate = 2.5
Loan_Term = 30
```

In [42]:
```
df_20 = amortization_schedule(Interest_Rate,Loan_Amount,Loan_Term)
df_20
```

Out[42]:

	Period	Interest	Principal	Payment	Total Principal Paid	Total Interest Paid	Total Payment Paid
0	1	2083.33	1867.88	3951.21	1867.88	2083.33	3951.21
1	2	2079.44	1871.77	3951.21	3739.65	4162.77	7902.42
2	3	2075.54	1875.67	3951.21	5615.32	6238.31	11853.63
3	4	2071.63	1879.57	3951.20	7494.89	8309.94	15804.83
4	5	2067.72	1883.49	3951.21	9378.38	10377.66	19756.04
...	...	...	...	...	...	...	...
355	356	40.90	3910.31	3951.21	984277.12	422353.28	1406630.40
356	357	32.76	3918.45	3951.21	988195.57	422386.04	1410581.61
357	358	24.59	3926.62	3951.21	992122.19	422410.63	1414532.82
358	359	16.41	3934.80	3951.21	996056.99	422427.04	1418484.03
359	360	8.21	3942.99	3951.20	999999.98	422435.25	1422435.23

360 rows × 7 columns

Total Interest Paid = 422,435

Total Payment Made = 1,422,435

In [ ]: