
Amortizer

Release 0.0.3

vlntsolo

May 22, 2021

CONTENTS:

1	amortizer	1
1.1	Amortizer package	1
2	Indices and tables	5
	Python Module Index	7
	Index	9

AMORTIZER

1.1 Amortizer package

1.1.1 Submodules

1.1.2 amortizer.generator module

class amortizer.generator.**Amortizer**(*amount: float, period: int, interest_rate: float*)

Bases: object

Amortizer Class instantiates an new object for a loan ammortization calculations. Attributes are used as initial values for the two implemented methods of payment schedule: **annuity** and classic **straight** amortization.

Attributes:

amount (float): Loan body (i.e. 100000 or 900.50)

period (int): Number of months (i.e. 5 years = 60)

interest_rate (float): Interest rate in percents (i.e. if 9% then 9 without ‘%’ sign. Decimal types: 9.5 are allowed.)

Methods:

straight_amortization(): Calculates amortization table with straight amortization and returns a dataframe.

annuity_amortization(): Calculates amortization table with annuity payments and returns a dataframe.

get_summary(method=”annuity”): Calculates amortization dataframe (methods: ‘straight’ or ‘annuity’) and returns a dictionary with summary statistics.

to_html(method=”annuity”): Calculates amortization dataframe (methods: ‘straight’ or ‘annuity’) and exports results to a string with html markup.

to_json(method=”annuity”): Calculates amortization dataframe (methods: ‘straight’ or ‘annuity’) and exports results to a string with JSON object.

to_csv(path: str, method=”annuity”): Calculates amortization dataframe (methods: ‘straight’ or ‘annuity’) and exports results to the .csv file.

annuity_amortization()

Calculates amortization table with annuity payments

A loan amortized in the annuity method comprises a series of payments made between equal time intervals.

This amortization type is used to pay an equal amount each payment period. In the beginning of the loan period,

the payments will consist of a bigger part interest and a smaller part amortization.

The further towards the end of the loan period you will come, the bigger part of the payment will be amortization and the smaller part interest.

Returns Returns pandas DataFrame with calculated values

Return type DataFrame

get_summary(*method='annuity'*)

Calculates amortization dataframe and returns a dictionary with summary statistics.

Parameters **method** (*str*, *optional*) – Specify one of the following methods to calculate the amortization table: 'straight' or 'annuity' (default)

Returns Returns a dictionary with the following keys: 'total_cost' (Total cost of the loan), 'average_interest_exp' (Average interest expense), 'average_monthly_pmt' (Average monthly payment), 'total_interest_exp' (Sum of interest payments)

Return type dict

straight_amortization()

Calculates amortization table with straight amortization. A loan amortized in the classic method comprises a series of payments made between equal time intervals.

Classic amortization means that the borrower will amortize an equal amount each payment period, but the interest will differ in size.

Due to that the interest rate is based on the outstanding debt, the interest paid each payment period is going to decrease the further in to the loan period you get.

Which means that in the beginning of the loan period, the payments will be the biggest and in the end of the loan period the payments will get smaller and smaller.

Returns Returns pandas DataFrame with calculated values

Return type DataFrame

to_csv(*path: str, method='annuity'*)

Calculates amortization dataframe and exports results to .csv file.

Parameters

- **path** (*str*) – Absolute path to a folder where the script should export [method]_amortization.csv file. For unix '/tmp/' or Windows 'C:/User/Andy/Desktop/' (Only forward slash accepted)
- **method** (*str*, *optional*) – Specify one of the following methods to calculate the amortization table: 'straight' or 'annuity' (default)

Returns prints success message after the exported file was recorded on your server/local machine.

Return type print

to_html(*method='annuity'*)

Calculates amortization dataframe and exports results to html markup.

Parameters **method** (*str*, *optional*) – Specify one of the following methods to calculate the amortization table: 'straight' or 'annuity' (default)

Returns Returns calculated amortization table within a string in html/css markup.

Return type str

to_json(*method='annuity'*)

Calculates amortization dataframe and exports results to JSON.

Parameters **method** (*str*, *optional*) – Specify one of the following methods to calculate the amortization table: 'straight' or 'annuity' (default)

Returns Returns calculated amortization table within a string in JSON format.

Return type str

1.1.3 Module contents

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

a

`amortizer, ??`

`amortizer.generator, 1`

INDEX

A

`amortizer`
 module, 1, 3
`Amortizer` (*class in amortizer.generator*), 1
`amortizer.generator`
 module, 1
`annuity_amortization()` (*amortizer.generator.Amortizer method*), 1

G

`get_summary()` (*amortizer.generator.Amortizer method*), 2

M

module
 amortizer, 1, 3
 amortizer.generator, 1

S

`straight_amortization()` (*amortizer.generator.Amortizer method*), 2

T

`to_csv()` (*amortizer.generator.Amortizer method*), 2
`to_html()` (*amortizer.generator.Amortizer method*), 2
`to_json()` (*amortizer.generator.Amortizer method*), 2