Eflatun UAV

Release 2023

Muhammed Sezer, Şevval Dikkaya

May 05, 2023

CONTENTS:

CHAPTER

ONE

ALL MODULES

eflatun_uav	

1.1 eflatun_uav

Modules

eflatun_uav.helpers	
eflatun_uav.number_generators	This module creates numbers for given variable type of inputs

1.1.1 eflatun_uav.helpers

Modules

eflatun_uav.helpers.number_generators	This module creates numbers for given variable type of inputs

eflatun_uav.helpers.number_generators

This module creates numbers for given variable type of inputs

Functions

convert_string_to_int(string, *[, base])	Converts a string to an deterministicly random integer representation using the specified base.

Classes

TesterClass()	This is a summary for the TesterClass.

class eflatun_uav.helpers.number_generators.**TesterClass**

Bases: `object`

This is a summary for the TesterClass.

__init__() → None

Initialize the TesterClass instance.

tester__(*adam: int*, *kadin: float*) → str

This is a summary for the **tester__** method.

Parameters

- **adam** (*int*) – A description for the 'adam' parameter.
- **kadin** (*float*) – A description for the 'kadin' parameter.

Returns

A description for the return value.

Return type

str

eflatun_uav.helpers.number_generators.**convert_string_to_int**(*string: str*, *, *base: int | None = 256*) → int

Converts a string to an deterministicly random integer representation using the specified base.

This function calculates two totals, one for the forward direction of the string and another for the reverse direction. The final result is the sum of both totals modulo the given base.

Works better for texts longer than 5 letters.

Parameters

- **string** (*str*) – The input string to be converted to an integer.
- **base** (*Optional[int], optional*) – The base to be used for the conversion. Defaults to 256.

Raises

ValueError – If the base is not an integer or if it is 0, -1, or 1.

Returns

The integer representation of the input string.

Return type

int

Example

```
>>> convert_string_to_int("Hello, World!")
157
>>> convert_string_to_int("Hello, World")
84
>>> convert_string_to_int("Hello, World!", base = 36)
13
```

1.1.2 eflatun_uav.number_generators

This module creates numbers for given variable type of inputs

Functions

convert_string_to_int(string, *[, base])	Converts a string to an deterministicly random integer representation using the specified base.

Classes

TesterClass()	_summary_

class `eflatun_uav.number_generators.`**`TesterClass`**

Bases: `object`

summary

`__init__`() → None

summary

`tester__`(*adam: int*, *kadin: float*) → str

summary

Parameters

- **adam** (*int*) – _description_
- **kadin** (*float*) – _description_

Returns

description

Return type

string

`eflatun_uav.number_generators.`**`convert_string_to_int`**(*string: str*, *, *base: int | None = 256*) → int

Converts a string to an deterministicly random integer representation using the specified base.

This function calculates two totals, one for the forward direction of the string and another for the reverse direction. The final result is the sum of both totals modulo the given base.

Works better for texts longer than 5 letters.

Parameters

- **string** (*str*) – The input string to be converted to an integer.
- **base** (*Optional[int], optional*) – The base to be used for the conversion. Defaults to 256.

Raises

ValueError – If the base is not an integer or if it is 0, -1, or 1.

Returns

The integer representation of the input string.

Return type

int

Example

```
>>> convert_string_to_int("Hello, World!")
157
>>> convert_string_to_int("Hello, World")
84
>>> convert_string_to_int("Hello, World!", base = 36)
13
```

CHAPTER

TWO

INDICES AND TABLES

- modindex
- search

PYTHON MODULE INDEX

INDEX