time_str

Release 1.0.0

BobDotCom

Sep 21, 2022

CONTENTS

A package to convert user input into datetime.timedelta objects.

PyPI: https://pypi.org/project/time_Str/

Docs: https://time-str.readthedocs.io/en/latest/

CHAPTER ONE

INSTALLATION

You can install released versions of time_str from the Python Package Index via pip or a similar tool:

Stable Release: `pip install time_str`

Working Version: `pip install git+https://github.com/BobDotCom/time_str.git`

CHAPTER

TWO

USAGE

```
import time_str
converter = Converter('11 months 9days 1m 3 sec')
print(converter.timedelta_precise())
print(converter.datetime_precise())
```

2.1 Manuals

2.1.1 API Reference

The following section outlines the API of time_str.

Core Utilities

These provide the main functionality of time_str.

Shortcut Functions

`time_str.`**`parse_interval`**(*interval: str*, *max_unit: typing_extensions.Literal[seconds, minutes, hours, days, weeks, months, years, decades, centuries] = 'centuries'*) → *IntervalConverter*

A shortcut function for *`IntervalConverter`*.

Parameters

- **`interval`** (`str`) – The string to parse.
- **`max_unit`** (*`Literal["seconds", "minutes", "hours", "days", "weeks", "months", "years", "decades", "centuries"]`*) – The maximum unit to parse to. Defaults to `"centuries"`.

Returns

A converter object.

Return type

`IntervalConverter`

Converter Classes

class `time_str.`**`IntervalConverter`**(*input_string: str, max_unit: typing_extensions.Literal[seconds, minutes, hours, days, weeks, months, years, decades, centuries] = 'centuries'*)

A converter to parse user input representing an amount of time into `datetime.datetime` and `datetime.timedelta` objects.

Parameters

- **`input_string`** (`str`) – A string (usually user input) to be converted.
- **`max_unit`** (*`Literal["seconds", "minutes", "hours", "days", "weeks", "months", "years", "decades", "centuries"]`*) – The maximum unit to convert to. Defaults to "centuries".

property input_string: str

A string (usually user input) to be converted.

`datetime_precise`() → datetime

A precise converter that uses the current system time, and accounts for conditional changes such as leap years, and months with varying days.

Note: The return value of this method is cached, so it will always return the same value when called on the same instance. However, it may return a different result when called at different times across multiple objects. This is because the current system time when the parent object was created is used to calculate the result.

Returns
A datetime object representing the parsed time.

Return type
`datetime.datetime`

`datetime_relative`() → datetime

A relative converter that doesn't take leap years into account and uses rounded values for months.

Note: It is almost always recommended to use *`datetime_precise()`* instead.

Note: The return value of this method is cached, so it will always return the same value when called on the same instance. However, it may return a different result when called at different times across multiple objects. This is because the current system time when the parent object was created is used to calculate the result.

Returns
A datetime object representing the parsed time.

Return type
`datetime.datetime`

timedelta_precise() → timedelta

A precise converter that uses the current system time, and accounts for conditional changes such as leap years, and months with varying days.

Note: The return value of this method is cached, so it will always return the same value when called on the same instance. However, it may return a different result when called at different times across multiple objects. This is because the current system time when the parent object was created is used to calculate the result.

Returns
A timedelta object representing the parsed amount of time.

Return type
`datetime.timedelta`

timedelta_relative() → timedelta

A relative converter that doesn't take leap years into account and uses rounded values for months.

Note: Unless you cannot rely on system time or need a static return value, you should use *timedelta_precise()* instead.

Returns
A timedelta object representing the parsed amount of time.

Return type
`datetime.timedelta`

CHAPTER THREE

INDICES AND TABLES

- genindex
- modindex
- search

INDEX