
Ujo Types Documentation

Release 0.1.2

The Titan Project

Mar 11, 2021

1	Installing the extension	3
2	Building the extension	5
2.1	Running Unit Tests	5
2.2	Code Quality	5
2.2.1	pylint	5
2.2.2	flake8	6
2.2.2.1	ujotypes	6
2.2.2.2	C API Wrapper	18
2.2.2.3	Index and Tables	27
	Python Module Index	29
	Index	31

This Python extension allows you to use the `ujotypes-c` library in Python. Use of `ujotypes` in Python is primarily through a class wrapper. Additionally you can access the native C-API functions if you have to.

- [Python 3.7.x](#)
- [Windows x64 and Linux x64 binary wheels available on PyPi](#)
- [Sourcecode](#)
- [Documentation](#)

CHAPTER 1

Installing the extension

The `ujotypes` extension is available on the Python Package Index (PyPi). The extension is available as a binary wheel for Linux x64 systems that are `manylinux` compatible and on Windows x64 systems.

To install it without having to build it yourself open a command line and run:

```
pip install ujotypes
```

Building the extension

Building the extensions requires a local compiler. When building on Windows you need to match the compiler of the Python version you want to build the extension for. A [comprehensive listing](#) is available in the pythondev documentation.

Also the [ujotypes-c](#) project needs to be cloned and built locally.

The standard build process for the ujotypes-py Python extension expects to find the directory of the ujotypes-c project to be next to the ujotypes-py directory. Should your clone of the ujotypes-c project be in a different location you will have to adjust the paths in the *setup.py* file.

To build and install open a command line and run:

```
python setup.py build
pip install .
```

2.1 Running Unit Tests

The ujotype-py project comes with a set of unittests. To run tests from the package root dir call:

```
python -m unittest discover test -v
```

2.2 Code Quality

We value code quality and ask you to check your code with pylint and flake8 before making a merge request.

2.2.1 pylint

Running pylint for the python code in this project the following commands are used:

```
pylint --rcfile=.pylintrc ./ujotypes/  
pylint --rcfile=.pylintrc --disable=duplicate-code ./test/
```

2.2.2 flake8

You can run flake 8 with:

```
flake8
```

2.2.2.1 ujotypes

ujotypes.variants.base

This module provides the base object all ujotypes inherit from

class ujotypes.variants.base.UjoBase (variant_handle, owner=False)

Bases: object

Base class wrapping an ujotypes-c object into a Python object

Parameters

- **variant_handle** (*int*) – A handle returned by the Ujo C-API representing a variant
- **owner** (*bool*) – Indicates if the class takes the ownership of the variant handle

handle

C API handle of the variant

Type (int)

refcount

Current reference count of Ujo variant

Type (int)

subtype

Subtype constant of the variant

Type (int)

type

Type constant of the variant

Type (int)

value

Abstract method to be implemented in concrete classes.

variant_subtype = 0

variant_type = 0

ujotypes.variants.none

Ujo Types None wrapper class

```
class ujointypes.variants.none.UjoNone (variant_handle=None, owner=False)
```

Bases: `ujointypes.variants.base.UjoBase`

Wrap Ujo None type

Parameters

- **variant_handle** (*int*) – A handle returned by the Ujo C-API representing a variant. If no handle is passed a new handle will be created.
- **owner** (*bool*) – Indicates if the class owns the variant

value

Current Value

Type (None)

variant_type = 15

ujointypes.variants.boolean

Ujo Types Boolean wrapper class

```
class ujointypes.variants.boolean.UjoBool (value=None, variant_handle=None,
                                           owner=False)
```

Bases: `ujointypes.variants.base.UjoBase`

Wrap Ujo Boolean type

Parameters

- **value** (*bool*, None) – The value can be None if a valid variant handle is passed.
- **variant_handle** (*int*) – A handle returned by the Ujo C-API representing a variant. If a handle is passed the value is ignored.
- **owner** (*bool*) – Indicates if the class owns the variant

value

Current Value

Type (bool)

variant_type = 13

ujointypes.variants.integer

Ujo Types Integer wrapper class

```
class ujointypes.variants.integer.UjoInt16 (value=None, variant_handle=None,
                                           owner=False)
```

Bases: `ujointypes.variants.base.UjoBase`

Wrap Ujo int16 type

Parameters

- **value** (*int*, None) – The value can be None if a valid variant handle is passed.
- **variant_handle** (*int*) – A handle returned by the Ujo C-API representing a variant. If a handle is passed the value is ignored.
- **owner** (*bool*) – Indicates if the class owns the variant

value

Current Value

Type (int)

variant_type = 7

class `ujotypes.variants.integer.UjoInt32` (*value=None*, *variant_handle=None*,
owner=False)

Bases: `ujotypes.variants.base.UjoBase`

Wrap Ujo int32 type

Parameters

- **value** (*int*, *None*) – The value can be None if a valid variant handle is passed.
- **variant_handle** (*int*) – A handle returned by the Ujo C-API representing a variant. If a handle is passed the value is ignored.
- **owner** (*bool*) – Indicates if the class owns the variant

value

Current Value

Type (int)

variant_type = 6

class `ujotypes.variants.integer.UjoInt64` (*value=None*, *variant_handle=None*,
owner=False)

Bases: `ujotypes.variants.base.UjoBase`

Wrap Ujo int64 type

Parameters

- **value** (*int*, *None*) – The value can be None if a valid variant handle is passed.
- **variant_handle** (*int*) – A handle returned by the Ujo C-API representing a variant. If a handle is passed the value is ignored.
- **owner** (*bool*) – Indicates if the class owns the variant

value

Current Value

Type (int)

variant_type = 5

class `ujotypes.variants.integer.UjoInt8` (*value=None*, *variant_handle=None*,
owner=False)

Bases: `ujotypes.variants.base.UjoBase`

Wrap Ujo int8 type

Parameters

- **value** (*int*, *None*) – The value can be None if a valid variant handle is passed.
- **variant_handle** (*int*) – A handle returned by the Ujo C-API representing a variant. If a handle is passed the value is ignored.
- **owner** (*bool*) – Indicates if the class owns the variant

value

Current Value

Type (int)

variant_type = 8

```
class ujointypes.variants.integer.UjoUInt16 (value=None, variant_handle=None,
                                             owner=False)
```

Bases: *ujointypes.variants.base.UjoBase*

Wrap Ujo uint16 type

Parameters

- **value** (*int*, *None*) – The value can be None if a valid variant handle is passed.
- **variant_handle** (*int*) – A handle returned by the Ujo C-API representing a variant. If a handle is passed the value is ignored.
- **owner** (*bool*) – Indicates if the class owns the variant

value

Current Value

Type (int)

variant_type = 11

```
class ujointypes.variants.integer.UjoUInt32 (value=None, variant_handle=None,
                                             owner=False)
```

Bases: *ujointypes.variants.base.UjoBase*

Wrap Ujo uint32 type

Parameters

- **value** (*int*, *None*) – The value can be None if a valid variant handle is passed.
- **variant_handle** (*int*) – A handle returned by the Ujo C-API representing a variant. If a handle is passed the value is ignored.
- **owner** (*bool*) – Indicates if the class owns the variant

value

Current Value

Type (int)

variant_type = 10

```
class ujointypes.variants.integer.UjoUInt64 (value=None, variant_handle=None,
                                             owner=False)
```

Bases: *ujointypes.variants.base.UjoBase*

Wrap Ujo uint64 type

Parameters

- **value** (*int*, *None*) – The value can be None if a valid variant handle is passed.
- **variant_handle** (*int*) – A handle returned by the Ujo C-API representing a variant. If a handle is passed the value is ignored.
- **owner** (*bool*) – Indicates if the class owns the variant

value

Current Value

Type (int)

variant_type = 9

```
class ujointypes.variants.integer.UjoUInt8 (value=None, variant_handle=None,
                                             owner=False)
Bases: ujointypes.variants.base.UjoBase
```

Wrap Ujo uint8 type

Parameters

- **value** (*int*, *None*) – The value can be None if a valid variant handle is passed.
- **variant_handle** (*int*) – A handle returned by the Ujo C-API representing a variant. If a handle is passed the value is ignored.
- **owner** (*bool*) – Indicates if the class owns the variant

value

Current Value

Type (*int*)

variant_type = 12

ujointypes.variants.float

Ujo Types Floats wrapper class

```
class ujointypes.variants.float.UjoFloat16 (value=None, variant_handle=None,
                                             owner=False)
Bases: ujointypes.variants.base.UjoBase
```

Wrap Ujo float16 type

Parameters

- **value** (*float*, *None*) – The value can be None if a valid variant handle is passed.
- **variant_handle** (*int*) – A handle returned by the Ujo C-API representing a variant. If a handle is passed the value is ignored.
- **owner** (*bool*) – Indicates if the class owns the variant

value

Current Value

Type (*float*)

variant_type = 3

```
class ujointypes.variants.float.UjoFloat32 (value=None, variant_handle=None,
                                             owner=False)
Bases: ujointypes.variants.base.UjoBase
```

Wrap Ujo float32 type

Parameters

- **value** (*float*, *None*) – The value can be None if a valid variant handle is passed.
- **variant_handle** (*int*) – A handle returned by the Ujo C-API representing a variant. If a handle is passed the value is ignored.
- **owner** (*bool*) – Indicates if the class owns the variant

value

Current Value

Type (float)

variant_type = 2

class `ujotypes.variants.float.UjoFloat64` (*value=None*, *variant_handle=None*,
owner=False)

Bases: `ujotypes.variants.base.UjoBase`

Wrap Ujo float64 type

Parameters

- **value** (*float*, *None*) – The value can be None if a valid variant handle is passed.
- **variant_handle** (*int*) – A handle returned by the Ujo C-API representing a variant. If a handle is passed the value is ignored.
- **owner** (*bool*) – Indicates if the class owns the variant

value

Current Value

Type (float)

variant_type = 1

ujotypes.variants.string

Ujo Types String wrapper class

class `ujotypes.variants.string.UjoStringC` (*value=None*, *variant_handle=None*,
owner=False)

Bases: `ujotypes.variants.base.UjoBase`

Wrap Ujo C string type

Parameters

- **value** (*str*, *None*) – The value can be None if a valid variant handle is passed.
- **variant_handle** (*int*) – A handle returned by the Ujo C-API representing a variant. If a handle is passed the value is ignored.
- **owner** (*bool*) – Indicates if the class owns the variant

value

Current Value

Type (str)

variant_subtype = 0

variant_type = 4

class `ujotypes.variants.string.UjoStringUTF8` (*value=None*, *variant_handle=None*,
owner=False)

Bases: `ujotypes.variants.base.UjoBase`

Wrap Ujo UTF-8 string type

Parameters

- **value** (*str*, *None*) – The value can be None if a valid variant handle is passed.
- **variant_handle** (*int*) – A handle returned by the Ujo C-API representing a variant. If a handle is passed the value is ignored.

- **owner** (*bool*) – Indicates if the class owns the variant

value

Current Value

Type (*str*)

variant_subtype = 1

variant_type = 4

ujotypes.variants.timestamp

Ujo Types Timestamp wrapper class

```
class ujotypes.variants.timestamp.UjoTimestamp (value=None,    variant_handle=None,
                                              owner=False)
```

Bases: *ujotypes.variants.base.UjoBase*

Wrap Ujo Timestamp type

Parameters

- **value** (*datetime*, *None*) – The value can be None if a valid variant handle is passed.
- **variant_handle** (*int*) – A handle returned by the Ujo C-API representing a variant. If a handle is passed the value is ignored.
- **owner** (*bool*) – Indicates if the class owns the variant

value

Current Value

Type (*datetime*)

variant_type = 19

ujotypes.variants.container (List and Map)

Ujo Types container wrapper classes

```
class ujotypes.variants.container.UjoList (variant_handle=None, owner=False)
Bases: ujotypes.variants.base.UjoBase, collections.abc.MutableSequence
```

An Ujo list of Ujo objects

The Ujo list is comparable to a python list and provides the same operators and functions being available by Python Lists. The main difference being that the Ujo list itself contains only typed Ujo objects.

Parameters

- **variant_handle** (*int*) – A handle returned by the Ujo C-API representing a variant. If None is passed a new handle is created.
- **owner** (*bool*) – Indicates if the class owns the variant

append (*value*)

Append an Ujo Types element to the end of the list

Parameters **value** (*UjoBase*) – The Ujo variant to be added to the list

Raises *TypeError* – If objects other than objects derived from *UjoBase* are appended

as_pyobject () → list

Converts the Ujo container to its corresponding python representation.

Returns Python representation of Ujo object

Return type list

copy (deepcopy=False)

Create a copy of the ujo list

Parameters **deepcopy** (bool) – create a copy of contained items (recursively), default: False

Returns a new instance of an UjoList holding the same (or a copy of) the content

Return type *UjoList*

count (value)

Return number of occurrences of Ujo elements in list having same type and value

Parameters **value** (*UjoBase*) – The Ujo variant to count occurrences for

Returns Number of occurrences

Return type (int)

get_value (index, default=None, constraint=None)

Read a value from an index position with default fallback and constraint check

Parameters

- **index** (int) – item index
- **default** (any) – default fallback value
- **constraint** (class) – class type as constraint for the list item

Returns item value from the list at index position or default

Return type (*UjoBase*)

get_variant (index, default=<class 'ujotypes.variants.none.UjoNone'>, constraint=None)

Read an item from an index position with default fallback and constraint check

Parameters

- **index** (int) – item index
- **default** (*UjoBase*) – default fallback value
- **constraint** (class) – class type as constraint for the list item

Returns item from the list at index position or default

Return type (*UjoBase*)

index (value, start=0, stop=None)

Return first index of Ujo element in list having same type and value

For searching the element just inside a slice of the list a start and stop index can be passed to the index function. Negative numbers to count from the end of the list are allowed.

Parameters

- **value** (*UjoBase*) – The Ujo variant to return index for
- **index** (int) – Index of element to insert before
- **start** (int) – Optional index to start searching at

- **stop** (*int*) – Optional index to stop searching at

Returns Index of first element in list

Return type (*int*)

Raises `ValueError` – If element of same type and value is not present

insert (*index*, *value*)

Insert an Ujo variant element before index

Parameters

- **index** (*int*) – Index of element to insert before
- **value** (`UjoBase`) – The Ujo variant to be added to the list

Raises `TypeError` – If objects other than objects derived from `UjoBase` are inserted

set_value (*index*, *value*, *value_type*)

Replace a value at an index position

The value has to fit the *value_type*. If the value is an instance of `UjoBase`, the *value_type* is ignored.

Parameters

- **index** (*int*) – item index
- **value** (*any*) – the value to set
- **value_type** (*class*) – class type for the list item

value

Python list

Type (*list*)

variant_type = 48

class `ujotypes.variants.container.UjoMap` (*variant_handle=None*, *owner=True*)

Bases: `ujotypes.variants.base.UjoBase`, `collections.abc.MutableMapping`

An Ujo map of Ujo objects

The Ujo map is comparable to a Python dictionary and provides the same operators and functions being available by Python dictionaries. The main difference being that the Ujo map itself contains only typed Ujo objects.

Parameters

- **variant_handle** (*int*) – A handle returned by the Ujo C-API representing a map variant. If `None` is passed a new handle is created.
- **owner** (*bool*) – Indicates if the class owns the variant

as_pyobject () → *dict*

Converts an Ujo container to its corresponding python representation.

Duplicate keys of different Ujo key types will only create one entry in the Python dictionary with the value of the Ujo key value pair evaluated last.

Returns Python representation of Ujo object

Return type *dict*

copy (*deepcopy=False*)

Create a copy of the ujo map

Parameters **deepcopy** (*bool*) – create a copy of contained items (recursively), default: `False`

Returns a new instance of an UjoMap holding the same (or a copy of) the content

Return type *UjoMap*

get_value (*key*, *default=None*, *constraint=None*, *key_type=<class 'ujotypes.variants.string.UjoStringC'>*)

Get value from map by its key with default fallback and constraint check

The key has to fit the *key_type*. If the key is an instance of UjoBase, the *key_type* is ignored.

Parameters

- **key** (*any*) – the key to find value with
- **default** (*any*) – default fallback value
- **constraint** (*class*) – class type as constraint for the map item
- **key_type** (*class*) – class type for the key in the map

Returns value of item from the map or default

Return type (*any*)

get_variant (*key*, *default=<class 'ujotypes.variants.none.UjoNone'>*, *constraint=None*, *key_type=<class 'ujotypes.variants.string.UjoStringC'>*)

Get item from map by its key with default fallback and constraint check

The key has to fit the *key_type*. If the key is an instance of UjoBase, the *key_type* is ignored.

Parameters

- **key** (*any*) – the key to find value with
- **default** (*UjoBase*) – default fallback value
- **constraint** (*class*) – class type as constraint for the map item
- **key_type** (*class*) – class type for the key in the map

Returns item from the map or default

Return type (*UjoBase*)

set_value (*key*, *value*, *value_type*, *key_type=<class 'ujotypes.variants.string.UjoStringC'>*)

Set or add a key value pair to the map

The key has to fit the *key_type*. The value has to fit the *value_type*. If the key is an instance of UjoBase, the *key_type* is ignored. If the value is an instance of UjoBase, the *value_type* is ignored.

Parameters

- **key** (*any*) – the key to store the value with
- **value** (*any*) – the value to store
- **value_type** (*class*) – class type for the map item
- **key_type** (*class*) – class type for the key in the map

value

Python dictionary

Type (*dict*)

variant_type = 49

`ujotypes.variants.container.read_buffer` (*buffer*)

Read Ujo container from in memory buffer to in memory Ujo Types variant

Parameters **buffer** (*bytes*) – Binary buffer containing an Ujo container

Returns Ujo Types variant instance

Return type (UjoTypesBase)

`ujotypes.variants.container.read_file(filename)`

Read Ujo file from disc to in memory Ujo Types variant

Parameters **filename** (*str*) – Full path of file to read

Returns Ujo Types variant instance

Return type (UjoTypesBase)

`ujotypes.variants.container.variant_factory(variant_handle, owner=False)`

Put an Ujo variant handle inside a class

Parameters

- **variant_handle** (*int*) – C-API variant handle
- **owner** (*bool*) – indicates if the class wrapper owns the handle

Returns A variant wrapper class containing the handle

Return type (*UjoBase*)

`ujotypes.variants.container.write_buffer(variant)`

Write in memory Ujo Types variant to in memory binary buffer

Parameters **variant** (*UjoTypesBase*) – List or map Ujo Types variant

Returns Binary buffer containing Ujo container

Return type (*bytes*)

`ujotypes.variants.container.write_file(variant, filename)`

Write in memory Ujo Types variant to Ujo file on disk

Parameters

- **variant** (*UjoTypesBase*) – List or map Ujo Types variant
- **filename** (*str*) – Full path of file to write

ujotypes.tools

Auxiliary functions for easy ujo type access and conversion

ujotypes.tools.ujo_to_python

Converting Ujo Container objects to Python Object

`ujotypes.tools.ujo_to_python.ujo_to_python` (*variant: Union[ujotypes.variants.none.UjoNone, ujotypes.variants.boolean.UjoBool, ujotypes.variants.integer.UjoUInt8, ujotypes.variants.integer.UjoInt8, ujotypes.variants.integer.UjoUInt16, ujotypes.variants.integer.UjoInt16, ujotypes.variants.integer.UjoUInt32, ujotypes.variants.integer.UjoInt32, ujotypes.variants.integer.UjoUInt64, ujotypes.variants.integer.UjoInt64, ujotypes.variants.float.UjoFloat64, ujotypes.variants.float.UjoFloat32, ujotypes.variants.float.UjoFloat16, ujotypes.variants.string.UjoStringC, ujotypes.variants.string.UjoStringUTF8, ujotypes.variants.timestamp.UjoTimestamp]*) → object

Recursively convert Ujo container object to native Python objects

Parameters **variant** (*UjoVariant*) – Instance of any Ujo object to convert

Returns

Python object representation of Ujo variant and all contained subobjects

Return type Union[list,dict]

ujotypes.tools.type_conversion

Type conversion helpers

`ujotypes.tools.type_conversion.ujot_ujo_cstr_by_key` (*dictionary, key*)
Get dictionary entry as UjoStringC

If the key isn't found in the dictionary, UjoNone is returned.

Parameters

- **dictionary** (*dict*) – The dictionary to search in
- **key** (*Any*) – The key to search in dictionary with

Returns The Ujo variant

Return type (*UjoStringC* or *UjoNone*)

`ujotypes.tools.type_conversion.ujot_ujo_uint16_by_key` (*dictionary, key*)
Get dictionary entry as UjoUInt16

If the key isn't found in the dictionary, UjoNone is returned.

Parameters

- **dictionary** (*dict*) – The dictionary to search in
- **key** (*Any*) – The key to search in dictionary with

Returns The Ujo variant

Return type (*UjoUInt16* or *UjoNone*)

`ujotypes.tools.type_conversion.ujot_ujo_uint64_by_key` (*dictionary*, *key*)
Get dictionary entry as UjoUInt64

If the key isn't found in the dictionary, UjoNone is returned.

Parameters

- **dictionary** (*dict*) – The dictionary to search in
- **key** (*Any*) – The key to search in dictionary with

Returns The Ujo variant

Return type (*UjoUInt64* or *UjoNone*)

Ujotypes-py provides a pythonic wrapper for the ujotypes-c library

`ujotypes.ujotypesError`
alias of `ujotypes.error`

`ujotypes.ujotypesError_CIRCULAR_REFERENCE`
alias of `ujotypes.error_CIRCULAR_REFERENCE`

`ujotypes.ujotypesError_ILLEGAL_FILE_CONTENT`
alias of `ujotypes.error_ILLEGAL_FILE_CONTENT`

`ujotypes.ujotypesError_INDEX_OUT_OF_RANGE`
alias of `ujotypes.error_INDEX_OUT_OF_RANGE`

`ujotypes.ujotypesError_INVALID_TYPE`
alias of `ujotypes.error_INVALID_TYPE`

`ujotypes.ujotypesError_KEY_NOT_EXISTS`
alias of `ujotypes.error_KEY_NOT_EXISTS`

`ujotypes.ujotypesError_NULL_PTR`
alias of `ujotypes.error_NULL_PTR`

`ujotypes.ujotypesError_NYI`
alias of `ujotypes.error_NYI`

`ujotypes.ujotypesError_OUT_OF_MEMORY`
alias of `ujotypes.error_OUT_OF_MEMORY`

`ujotypes.ujotypesError_REFCNT_EXCD`
alias of `ujotypes.error_REFCNT_EXCD`

`ujotypes.ujotypesError_UNDEFINED`
alias of `ujotypes.error_UNDEFINED`

`ujotypes.ujotypesError_UNKNOWN`
alias of `ujotypes.error_UNKNOWN`

`ujotypes.ujotypesError_WRONG_STRING_SIZE`
alias of `ujotypes.error_WRONG_STRING_SIZE`

2.2.2.2 C API Wrapper

The C API wrapper provides direct access to the ujotypes-c library functions.

`ujotypes._ujotypes.ujot_float_to_half` (*value*)
Convert single precision float to half precision float value

Parameters **value** (*float*) – the float32 value to be converted

Returns binary float16 value representation

Return type (int)

`ujotypes._ujotypes.ujot_get_version()`

Get the current API and library version

Returns (library version, api version)

Return type (tuple of (int, int))

`ujotypes._ujotypes.ujot_half_to_float(value)`

Convert half precision float to single precision float value

Parameters **value** (*int*) – binary float16 value representation to be converted

Returns float32 value

Return type (float)

`ujotypes._ujotypes.ujot_read_from_buffer(buffer)`

Reads a container variant from a buffer

Parameters **buffer** (*bytes*) – the Ujo representation of the container variant

Returns container variant handle

Return type (int)

`ujotypes._ujotypes.ujot_read_from_file(filename)`

Reads a container variant from a file

Parameters **filename** (*str*) – full path to Ujo file to read

Returns container variant handle

Return type (int)

`ujotypes._ujotypes.ujot_variant_as_binary(variant)`

Read the value from an binary variant

Parameters **variant** (*int*) – variant handle

Returns (binary data, binary type)

Return type (tuple of (bytes, int))

`ujotypes._ujotypes.ujot_variant_as_bool(variant)`

Read the value from a bool variant

Parameters **variant** (*int*) – variant handle

Returns the value of the variant

Return type (int)

`ujotypes._ujotypes.ujot_variant_as_float16(variant)`

Read the value from an float16 variant

Parameters **variant** (*int*) – variant handle

Returns binary float16 value representation

Return type (int)

`ujotypes._ujotypes.ujot_variant_as_float32(variant)`

Read the value from an float32 variant

Parameters **variant** (*int*) – variant handle

Returns the value of the variant

Return type (float)

`ujotypes._ujotypes.ujot_variant_as_float64 (variant)`

Read the value from an float64 variant

Parameters `variant` (*int*) – variant handle

Returns the value of the variant

Return type (float)

`ujotypes._ujotypes.ujot_variant_as_int16 (variant)`

Read the value from an int16 variant

Parameters `variant` (*int*) – variant handle

Returns the value of the variant

Return type (int)

`ujotypes._ujotypes.ujot_variant_as_int32 (variant)`

Read the value from an int32 variant

Parameters `variant` (*int*) – variant handle

Returns the value of the variant

Return type (int)

`ujotypes._ujotypes.ujot_variant_as_int64 (variant)`

Read the value from an int64 variant

Parameters `variant` (*int*) – variant handle

Returns the value of the variant

Return type (int)

`ujotypes._ujotypes.ujot_variant_as_int8 (variant)`

Read the value from an int8 variant

Parameters `variant` (*int*) – variant handle

Returns the value of the variant

Return type (int)

`ujotypes._ujotypes.ujot_variant_as_string_c (variant)`

Read the value from a c-string variant

Parameters `variant` (*int*) – variant handle

Returns the value of the variant

Return type (str)

`ujotypes._ujotypes.ujot_variant_as_string_utf8 (variant)`

Read the value from a UTF8-string variant

Parameters `variant` (*int*) – variant handle

Returns the value of the variant

Return type (str)

`ujotypes._ujotypes.ujot_variant_as_timestamp (variant)`

Read the value from an timestamp variant

Parameters **variant** (*int*) – variant handle

Returns (year, month, day, hour, minute, second, microsecond)

Return type (tuple of (int, int, int, int, int, int, int))

`ujotypes._ujotypes.ujot_variant_as_uint16 (variant)`

Read the value from an uint16 variant

Parameters **variant** (*int*) – variant handle

Returns the value of the variant

Return type (int)

`ujotypes._ujotypes.ujot_variant_as_uint32 (variant)`

Read the value from an uint32 variant

Parameters **variant** (*int*) – variant handle

Returns the value of the variant

Return type (int)

`ujotypes._ujotypes.ujot_variant_as_uint64 (variant)`

Read the value from an uint64 variant

Parameters **variant** (*int*) – variant handle

Returns the value of the variant

Return type (int)

`ujotypes._ujotypes.ujot_variant_as_uint8 (variant)`

Read the value from an uint8 variant

Parameters **variant** (*int*) – variant handle

Returns the value of the variant

Return type (int)

`ujotypes._ujotypes.ujot_variant_decref (variant)`

Decrement variant reference counter

Parameters **variant** (*int*) – variant handle

`ujotypes._ujotypes.ujot_variant_get_refcount (variant)`

Get current reference count

Parameters **variant** (*int*) – variant handle

Returns reference counter value

Return type (int)

`ujotypes._ujotypes.ujot_variant_get_type (variant)`

Determines the type of the given variant

Parameters **variant** (*int*) – variant handle

Returns (type, subtype)

Return type (tuple of (int, int))

`ujotypes._ujotypes.ujot_variant_incref (variant)`

Increment variant reference counter

Parameters **variant** (*int*) – variant handle

`ujotypes._ujotypes.ujot_variant_list_append(list, value)`

Appends a value to a list

Parameters

- **list** (*int*) – variant list handle
- **value** (*int*) – handle of variant to append to the list

Returns index of the appended value

Return type (*int*)

`ujotypes._ujotypes.ujot_variant_list_clear(list)`

Clears the list

Parameters **list** (*int*) – variant list handle

`ujotypes._ujotypes.ujot_variant_list_contains(list, value)`

Checks if an item is in the list

Parameters

- **list** (*int*) – variant list handle
- **value** (*int*) – handle to variant to check, if in the list

Returns index of the value or `UJOT_LIST_INDEX_UNKNOWN` if not found

Return type (*int*)

`ujotypes._ujotypes.ujot_variant_list_contains_in_subset(list, value, start, stop)`

Checks if an item is in the subset of a list

If you specify a negative stop index, contains is checked for all elements from start index to the end of the list.

Parameters

- **list** (*int*) – variant list handle
- **value** (*int*) – handle to variant to check, if in the list
- **start** (*int*) – index where to start the subset (inclusive)
- **stop** (*int*) – index where to stop the subset (inclusive)

Returns index of the value or `UJOT_LIST_INDEX_UNKNOWN` if not found

Return type (*int*)

`ujotypes._ujotypes.ujot_variant_list_delete_item(list, index)`

Deletes an item in a list

Parameters

- **list** (*int*) – variant list handle
- **index** (*int*) – index of the item to be deleted

`ujotypes._ujotypes.ujot_variant_list_get_item(list, index)`

Gets a value from the list

Parameters

- **list** (*int*) – variant list handle
- **index** (*int*) – index to get the value from

Returns (borrowed) variant handle

Return type (int)

`ujotypes._ujotypes.ujot_variant_list_get_size(list)`

Gets the number of items in the list

Parameters `list` (*int*) – variant list handle

Returns item count

Return type (int)

`ujotypes._ujotypes.ujot_variant_list_insert(list, value, index)`

Inserts a value to a list

Parameters

- `list` (*int*) – variant list handle
- `value` (*int*) – handle of variant to insert into the list
- `index` (*int*) – index to insert the value at

`ujotypes._ujotypes.ujot_variant_list_set_item(list, value, index)`

Overwrites a value in a list

Parameters

- `list` (*int*) – variant list handle
- `value` (*int*) – handle of variant to set into the list
- `index` (*int*) – index to set the value at

`ujotypes._ujotypes.ujot_variant_map_clear(map)`

Remove all items from a map

Parameters `map` (*int*) – variant map handle

`ujotypes._ujotypes.ujot_variant_map_delete(map, key)`

Delete from map by key

Parameters

- `map` (*int*) – variant map handle
- `key` (*int*) – variant handle of the key to be deleted

`ujotypes._ujotypes.ujot_variant_map_first(map)`

Set the iterator to the beginning of a map

Parameters `map` (*int*) – variant map handle

`ujotypes._ujotypes.ujot_variant_map_get(map, key)`

Get a value from the map by key

Parameters

- `map` (*int*) – variant map handle
- `key` (*int*) – variant handle of the key

Returns variant handle of the value or none, if key doesn't exist

Return type (int)

`ujotypes._ujotypes.ujot_variant_map_get_size(map)`

Gets the number of items in the map

Parameters `map` (*int*) – variant map handle

Returns item count

Return type (int)

`ujotypes._ujotypes.ujot_variant_map_haskey (map, key)`

Check if a key exists in a map

Parameters

- **map** (*int*) – variant map handle
- **key** (*int*) – variant handle of the key to search for

Returns

- UJOT_TRUE if key was found
- UJOT_FALSE if key wasn't found

Return type (int)

`ujotypes._ujotypes.ujot_variant_map_next (map)`

Get next item of a map

Parameters **map** (*int*) – variant map handle

Returns

- (key variant handle, value variant handle)
- None, if there are no more items in the map

Return type (tuple of (int, int), None)

`ujotypes._ujotypes.ujot_variant_map_set (map, key, value)`

Set a key value pair in a map

Parameters

- **map** (*int*) – variant map handle
- **key** (*int*) – variant handle of the key
- **value** (*int*) – variant handle of the value

`ujotypes._ujotypes.ujot_variant_new_binary (buffer, bin_type)`

Creates a variant with an binary value

Parameters

- **buffer** (*bytes*) – binary data
- **bin_type** (*int*) – binary type

Returns variant handle

Return type (int)

`ujotypes._ujotypes.ujot_variant_new_bool (value)`

Creates a variant with an boolean value

Parameters **value** (*int*) – the value to assign to the new variant

Returns variant handle

Return type (int)

`ujotypes._ujotypes.ujot_variant_new_float16 (value)`

Creates a variant with an float16 value

Parameters `value` (*int*) – binary float16 value representation

Returns variant handle

Return type (*int*)

`ujotypes._ujotypes.ujot_variant_new_float32` (*value*)

Creates a variant with an float32 value

Parameters `value` (*float*) – the value to assign to the new variant

Returns variant handle

Return type (*int*)

`ujotypes._ujotypes.ujot_variant_new_float64` (*value*)

Creates a variant with an float64 value

Parameters `value` (*float*) – the value to assign to the new variant

Returns variant handle

Return type (*int*)

`ujotypes._ujotypes.ujot_variant_new_int16` (*value*)

Creates a variant with an int16 value

Parameters `value` (*int*) – the value to assign to the new variant

Returns variant handle

Return type (*int*)

`ujotypes._ujotypes.ujot_variant_new_int32` (*value*)

Creates a variant with an int32 value

Parameters `value` (*int*) – the value to assign to the new variant

Returns variant handle

Return type (*int*)

`ujotypes._ujotypes.ujot_variant_new_int64` (*value*)

Creates a variant with an int64 value

Parameters `value` (*int*) – the value to assign to the new variant

Returns variant handle

Return type (*int*)

`ujotypes._ujotypes.ujot_variant_new_int8` (*value*)

Creates a variant with an int8 value

Parameters `value` (*int*) – the value to assign to the new variant

Returns variant handle

Return type (*int*)

`ujotypes._ujotypes.ujot_variant_new_list` ()

Create a new variant list

Returns variant handle

Return type (*int*)

`ujotypes._ujotypes.ujot_variant_new_map` ()

Create a new variant map

Returns variant handle

Return type (int)

`ujotypes._ujotypes.ujot_variant_new_none()`

Create new variant of type none

Returns variant handle

Return type (int)

`ujotypes._ujotypes.ujot_variant_new_string_c(string)`

Create a c-string variant

Parameters **string** (*str*) – the value to assign to the new variant

Returns variant handle

Return type (int)

`ujotypes._ujotypes.ujot_variant_new_string_utf8(string)`

Create an UTF8-string variant

Parameters **string** (*str*) – the value to assign to the new variant

Returns variant handle

Return type (int)

`ujotypes._ujotypes.ujot_variant_new_timestamp(year, month, day, hour, minute, second, microsecond)`

Creates a variant with an timestamp value

Parameters

- **year** (*int*) – the full year (e.g. 2018)
- **month** (*int*) – the month (1-12)
- **day** (*int*) – the day (1-31)
- **hour** (*int*) – the hour (0-23)
- **minute** (*int*) – the minute (0-59)
- **second** (*int*) – the second (0-59)
- **microsecond** (*int*) – the microsecond (0-999999)

Returns variant handle

Return type (int)

`ujotypes._ujotypes.ujot_variant_new_uint16(value)`

Creates a variant with an uint16 value

Parameters **value** (*int*) – the value to assign to the new variant

Returns variant handle

Return type (int)

`ujotypes._ujotypes.ujot_variant_new_uint32(value)`

Creates a variant with an uint32 value

Parameters **value** (*int*) – the value to assign to the new variant

Returns variant handle

Return type (int)

`ujotypes._ujotypes.ujot_variant_new_uint64(value)`

Creates a variant with an uint64 value

Parameters `value` (*int*) – the value to assign to the new variant

Returns variant handle

Return type (*int*)

`ujotypes._ujotypes.ujot_variant_new_uint8(value)`

Creates a variant with an uint8 value

Parameters `value` (*int*) – the value to assign to the new variant

Returns variant handle

Return type (*int*)

`ujotypes._ujotypes.ujot_variant_type_check(variant, type)`

Check type of an Ujo variant

Parameters

- **variant** (*int*) – variant handle
- **type** (*int*) – type id to be checked

Returns True if variant type matches

Return type (*bool*)

`ujotypes._ujotypes.ujot_write_to_buffer(variant)`

Writes a container variant into a buffer

Parameters `variant` (*int*) – container variant handle

Returns the Ujo representation of the container variant

Return type (*bytes*)

`ujotypes._ujotypes.ujot_write_to_file(variant, filename)`

Writes a container variant into a file

Parameters

- **variant** (*int*) – container variant handle
- **filename** (*str*) – full path to Ujo file to write

2.2.2.3 Index and Tables

- `genindex`
- `modindex`

u

- `ujotypes`, [18](#)
- `ujotypes._ujotypes`, [18](#)
- `ujotypes.tools`, [16](#)
- `ujotypes.tools.type_conversion`, [17](#)
- `ujotypes.tools.ujo_to_python`, [16](#)
- `ujotypes.variants.base`, [6](#)
- `ujotypes.variants.boolean`, [7](#)
- `ujotypes.variants.container`, [12](#)
- `ujotypes.variants.float`, [10](#)
- `ujotypes.variants.integer`, [7](#)
- `ujotypes.variants.none`, [6](#)
- `ujotypes.variants.string`, [11](#)
- `ujotypes.variants.timestamp`, [12](#)

A

`append()` (*ujotypes.variants.container.UjoList method*), 12
`as_pyobject()` (*ujotypes.variants.container.UjoList method*), 12
`as_pyobject()` (*ujotypes.variants.container.UjoMap method*), 14

C

`copy()` (*ujotypes.variants.container.UjoList method*), 13
`copy()` (*ujotypes.variants.container.UjoMap method*), 14
`count()` (*ujotypes.variants.container.UjoList method*), 13

G

`get_value()` (*ujotypes.variants.container.UjoList method*), 13
`get_value()` (*ujotypes.variants.container.UjoMap method*), 15
`get_variant()` (*ujotypes.variants.container.UjoList method*), 13
`get_variant()` (*ujotypes.variants.container.UjoMap method*), 15

H

`handle` (*ujotypes.variants.base.UjoBase attribute*), 6

I

`index()` (*ujotypes.variants.container.UjoList method*), 13
`insert()` (*ujotypes.variants.container.UjoList method*), 14

R

`read_buffer()` (*in module ujo-*
types.variants.container), 15

`read_file()` (*in module ujo-*
types.variants.container), 16

`refcount` (*ujotypes.variants.base.UjoBase attribute*), 6

S

`set_value()` (*ujotypes.variants.container.UjoList method*), 14
`set_value()` (*ujotypes.variants.container.UjoMap method*), 15
`subtype` (*ujotypes.variants.base.UjoBase attribute*), 6

T

`type` (*ujotypes.variants.base.UjoBase attribute*), 6

U

`ujo_to_python()` (*in module ujo-*
types.tools.ujo_to_python), 16
`UjoBase` (*class in ujo-*
types.variants.base), 6
`UjoBool` (*class in ujo-*
types.variants.boolean), 7
`UjoFloat16` (*class in ujo-*
types.variants.float), 10
`UjoFloat32` (*class in ujo-*
types.variants.float), 10
`UjoFloat64` (*class in ujo-*
types.variants.float), 11
`UjoInt16` (*class in ujo-*
types.variants.integer), 7
`UjoInt32` (*class in ujo-*
types.variants.integer), 8
`UjoInt64` (*class in ujo-*
types.variants.integer), 8
`UjoInt8` (*class in ujo-*
types.variants.integer), 8
`UjoList` (*class in ujo-*
types.variants.container), 12
`UjoMap` (*class in ujo-*
types.variants.container), 14
`UjoNone` (*class in ujo-*
types.variants.none), 6
`UjoStringC` (*class in ujo-*
types.variants.string), 11
`UjoStringUTF8` (*class in ujo-*
types.variants.string), 11
`ujot_float_to_half()` (*in module ujo-*
types._ujotypes), 18
`ujot_get_version()` (*in module ujo-*
types._ujotypes), 19
`ujot_half_to_float()` (*in module ujo-*
types._ujotypes), 19
`ujot_read_from_buffer()` (*in module ujo-*
types._ujotypes), 19

<code>ujot_read_from_file()</code> (in module <code>ujotypes._ujotypes</code>), 19	<code>ujot_variant_list_contains_in_subset()</code> (in module <code>ujotypes._ujotypes</code>), 22
<code>ujot_ujo_cstr_by_key()</code> (in module <code>ujotypes.tools.type_conversion</code>), 17	<code>ujot_variant_list_delete_item()</code> (in module <code>ujotypes._ujotypes</code>), 22
<code>ujot_ujo_uint16_by_key()</code> (in module <code>ujotypes.tools.type_conversion</code>), 17	<code>ujot_variant_list_get_item()</code> (in module <code>ujotypes._ujotypes</code>), 22
<code>ujot_ujo_uint64_by_key()</code> (in module <code>ujotypes.tools.type_conversion</code>), 17	<code>ujot_variant_list_get_size()</code> (in module <code>ujotypes._ujotypes</code>), 23
<code>ujot_variant_as_binary()</code> (in module <code>ujotypes._ujotypes</code>), 19	<code>ujot_variant_list_insert()</code> (in module <code>ujotypes._ujotypes</code>), 23
<code>ujot_variant_as_bool()</code> (in module <code>ujotypes._ujotypes</code>), 19	<code>ujot_variant_list_set_item()</code> (in module <code>ujotypes._ujotypes</code>), 23
<code>ujot_variant_as_float16()</code> (in module <code>ujotypes._ujotypes</code>), 19	<code>ujot_variant_map_clear()</code> (in module <code>ujotypes._ujotypes</code>), 23
<code>ujot_variant_as_float32()</code> (in module <code>ujotypes._ujotypes</code>), 19	<code>ujot_variant_map_delete()</code> (in module <code>ujotypes._ujotypes</code>), 23
<code>ujot_variant_as_float64()</code> (in module <code>ujotypes._ujotypes</code>), 20	<code>ujot_variant_map_first()</code> (in module <code>ujotypes._ujotypes</code>), 23
<code>ujot_variant_as_int16()</code> (in module <code>ujotypes._ujotypes</code>), 20	<code>ujot_variant_map_get()</code> (in module <code>ujotypes._ujotypes</code>), 23
<code>ujot_variant_as_int32()</code> (in module <code>ujotypes._ujotypes</code>), 20	<code>ujot_variant_map_get_size()</code> (in module <code>ujotypes._ujotypes</code>), 23
<code>ujot_variant_as_int64()</code> (in module <code>ujotypes._ujotypes</code>), 20	<code>ujot_variant_map_haskey()</code> (in module <code>ujotypes._ujotypes</code>), 24
<code>ujot_variant_as_int8()</code> (in module <code>ujotypes._ujotypes</code>), 20	<code>ujot_variant_map_next()</code> (in module <code>ujotypes._ujotypes</code>), 24
<code>ujot_variant_as_string_c()</code> (in module <code>ujotypes._ujotypes</code>), 20	<code>ujot_variant_map_set()</code> (in module <code>ujotypes._ujotypes</code>), 24
<code>ujot_variant_as_string_utf8()</code> (in module <code>ujotypes._ujotypes</code>), 20	<code>ujot_variant_new_binary()</code> (in module <code>ujotypes._ujotypes</code>), 24
<code>ujot_variant_as_timestamp()</code> (in module <code>ujotypes._ujotypes</code>), 20	<code>ujot_variant_new_bool()</code> (in module <code>ujotypes._ujotypes</code>), 24
<code>ujot_variant_as_uint16()</code> (in module <code>ujotypes._ujotypes</code>), 21	<code>ujot_variant_new_float16()</code> (in module <code>ujotypes._ujotypes</code>), 24
<code>ujot_variant_as_uint32()</code> (in module <code>ujotypes._ujotypes</code>), 21	<code>ujot_variant_new_float32()</code> (in module <code>ujotypes._ujotypes</code>), 25
<code>ujot_variant_as_uint64()</code> (in module <code>ujotypes._ujotypes</code>), 21	<code>ujot_variant_new_float64()</code> (in module <code>ujotypes._ujotypes</code>), 25
<code>ujot_variant_as_uint8()</code> (in module <code>ujotypes._ujotypes</code>), 21	<code>ujot_variant_new_int16()</code> (in module <code>ujotypes._ujotypes</code>), 25
<code>ujot_variant_decref()</code> (in module <code>ujotypes._ujotypes</code>), 21	<code>ujot_variant_new_int32()</code> (in module <code>ujotypes._ujotypes</code>), 25
<code>ujot_variant_get_refcount()</code> (in module <code>ujotypes._ujotypes</code>), 21	<code>ujot_variant_new_int64()</code> (in module <code>ujotypes._ujotypes</code>), 25
<code>ujot_variant_get_type()</code> (in module <code>ujotypes._ujotypes</code>), 21	<code>ujot_variant_new_int8()</code> (in module <code>ujotypes._ujotypes</code>), 25
<code>ujot_variant_incref()</code> (in module <code>ujotypes._ujotypes</code>), 21	<code>ujot_variant_new_list()</code> (in module <code>ujotypes._ujotypes</code>), 25
<code>ujot_variant_list_append()</code> (in module <code>ujotypes._ujotypes</code>), 21	<code>ujot_variant_new_map()</code> (in module <code>ujotypes._ujotypes</code>), 25
<code>ujot_variant_list_clear()</code> (in module <code>ujotypes._ujotypes</code>), 22	<code>ujot_variant_new_none()</code> (in module <code>ujotypes._ujotypes</code>), 26
<code>ujot_variant_list_contains()</code> (in module <code>ujotypes._ujotypes</code>), 22	<code>ujot_variant_new_string_c()</code> (in module <code>ujotypes._ujotypes</code>), 26

ujoinvariant_new_string_utf8() (in module *ujotypes._ujotypes*), 26
 ujoinvariant_new_timestamp() (in module *ujotypes._ujotypes*), 26
 ujoinvariant_new_uint16() (in module *ujotypes._ujotypes*), 26
 ujoinvariant_new_uint32() (in module *ujotypes._ujotypes*), 26
 ujoinvariant_new_uint64() (in module *ujotypes._ujotypes*), 27
 ujoinvariant_new_uint8() (in module *ujotypes._ujotypes*), 27
 ujoinvariant_type_check() (in module *ujotypes._ujotypes*), 27
 ujoinwrite_to_buffer() (in module *ujotypes._ujotypes*), 27
 ujoinwrite_to_file() (in module *ujotypes._ujotypes*), 27
 UjoinTimestamp (class in *ujotypes.variants.timestamp*), 12
ujotypes (module), 18
ujotypes._ujotypes (module), 18
ujotypes.tools (module), 16
ujotypes.tools.type_conversion (module), 17
ujotypes.tools.ujoto_python (module), 16
ujotypes.variants.base (module), 6
ujotypes.variants.boolean (module), 7
ujotypes.variants.container (module), 12
ujotypes.variants.float (module), 10
ujotypes.variants.integer (module), 7
ujotypes.variants.none (module), 6
ujotypes.variants.string (module), 11
ujotypes.variants.timestamp (module), 12
ujotypesError (in module *ujotypes*), 18
ujotypesError_CIRCULAR_REFERENCE (in module *ujotypes*), 18
ujotypesError_ILLEGAL_FILE_CONTENT (in module *ujotypes*), 18
ujotypesError_INDEX_OUT_OF_RANGE (in module *ujotypes*), 18
ujotypesError_INVALID_TYPE (in module *ujotypes*), 18
ujotypesError_KEY_NOT_EXISTS (in module *ujotypes*), 18
ujotypesError_NULL_PTR (in module *ujotypes*), 18
ujotypesError_NYI (in module *ujotypes*), 18
ujotypesError_OUT_OF_MEMORY (in module *ujotypes*), 18
ujotypesError_REFCNT_EXCD (in module *ujotypes*), 18
ujotypesError_UNDEFINED (in module *ujotypes*), 18
ujotypesError_UNKNOWN (in module *ujotypes*), 18
ujotypesError_WRONG_STRING_SIZE (in module *ujotypes*), 18
 UjoinUInt16 (class in *ujotypes.variants.integer*), 9
 UjoinUInt32 (class in *ujotypes.variants.integer*), 9
 UjoinUInt64 (class in *ujotypes.variants.integer*), 9
 UjoinUInt8 (class in *ujotypes.variants.integer*), 9

V

value (*ujotypes.variants.base.UjoinBase* attribute), 6
 value (*ujotypes.variants.boolean.UjoinBool* attribute), 7
 value (*ujotypes.variants.container.UjoinList* attribute), 14
 value (*ujotypes.variants.container.UjoinMap* attribute), 15
 value (*ujotypes.variants.float.UjoinFloat16* attribute), 10
 value (*ujotypes.variants.float.UjoinFloat32* attribute), 10
 value (*ujotypes.variants.float.UjoinFloat64* attribute), 11
 value (*ujotypes.variants.integer.UjoinInt16* attribute), 7
 value (*ujotypes.variants.integer.UjoinInt32* attribute), 8
 value (*ujotypes.variants.integer.UjoinInt64* attribute), 8
 value (*ujotypes.variants.integer.UjoinInt8* attribute), 8
 value (*ujotypes.variants.integer.UjoinUInt16* attribute), 9
 value (*ujotypes.variants.integer.UjoinUInt32* attribute), 9
 value (*ujotypes.variants.integer.UjoinUInt64* attribute), 9
 value (*ujotypes.variants.integer.UjoinUInt8* attribute), 10
 value (*ujotypes.variants.none.UjoinNone* attribute), 7
 value (*ujotypes.variants.string.UjoinStringC* attribute), 11
 value (*ujotypes.variants.string.UjoinStringUTF8* attribute), 12
 value (*ujotypes.variants.timestamp.UjoinTimestamp* attribute), 12
 variant_factory() (in module *ujotypes.variants.container*), 16
 variant_subtype (*ujotypes.variants.base.UjoinBase* attribute), 6
 variant_subtype (*ujotypes.variants.string.UjoinStringC* attribute), 11
 variant_subtype (*ujotypes.variants.string.UjoinStringUTF8* attribute), 12
 variant_type (*ujotypes.variants.base.UjoinBase* attribute), 6
 variant_type (*ujotypes.variants.boolean.UjoinBool* attribute), 7
 variant_type (*ujotypes.variants.container.UjoinList* attribute), 14
 variant_type (*ujotypes.variants.container.UjoinMap* attribute), 15
 variant_type (*ujotypes.variants.float.UjoinFloat16* attribute), 10
 variant_type (*ujotypes.variants.float.UjoinFloat32* attribute), 11

```
variant_type (ujotypes.variants.float.UjoFloat64 at-  
             attribute), 11  
variant_type (ujotypes.variants.integer.UjoInt16 at-  
             attribute), 8  
variant_type (ujotypes.variants.integer.UjoInt32 at-  
             attribute), 8  
variant_type (ujotypes.variants.integer.UjoInt64 at-  
             attribute), 8  
variant_type (ujotypes.variants.integer.UjoInt8 at-  
             attribute), 9  
variant_type (ujotypes.variants.integer.UjoUInt16  
             attribute), 9  
variant_type (ujotypes.variants.integer.UjoUInt32  
             attribute), 9  
variant_type (ujotypes.variants.integer.UjoUInt64  
             attribute), 9  
variant_type (ujotypes.variants.integer.UjoUInt8 at-  
             attribute), 10  
variant_type (ujotypes.variants.none.UjoNone at-  
             attribute), 7  
variant_type (ujotypes.variants.string.UjoStringC  
             attribute), 11  
variant_type                                     (ujo-  
             types.variants.string.UjoStringUTF8 attribute),  
12  
variant_type                                     (ujo-  
             types.variants.timestamp.UjoTimestamp  
             attribute), 12
```

W

```
write_buffer()      (in      module      ujo-  
                     types.variants.container), 16  
write_file()        (in      module      ujo-  
                     types.variants.container), 16
```