Template Class MetaParameter

Table of contents

- Defined in File parameter.hpp

Class Documentation

template<typename ValueT>
class MetaParameter

Class to define a parameter.

Public Functions

MetaParameter() = default

inline explicit MetaParameter(const ValueT &value)

Construct a new MetaParameter object.

Parameters

value – The value of the parameter.

inline explicit MetaParameter(ValueT &&value)

Construct a new MetaParameter object.

Parameters

value – The value of the parameter.

inline MetaParameter(const ValueT &value, const char *key, const char *headline, const char *description, ParameterFlag flag)

Construct a new MetaParameter object.

Parameters

- **value** – The value of the parameter.
- **key** – The key (name) of the parameter.
- **headline** – The headline of the parameter.

- **description** – The description of the parameter.
- **flag** – The flag of the parameter (default: ParameterFlag::kNone).

inline MetaParameter &operator=(const ValueT &value)

Define the assignment operator.

Parameters

value – The value of the parameter.

Returns

The reference to the parameter.

inline MetaParameter &&operator=(ValueT &&value)

Define the assignment operator.

Parameters

value – The value of the parameter.

Returns

The reference to the parameter.

inline const std::string &key() const

Get the key (name) of the parameter.

Returns

The key (name) of the parameter.

inline const std::string &headline() const

Get the headline of the parameter.

Returns

The headline of the parameter.

inline const std::string &description() const

Get the description of the parameter.

Returns

The description of the parameter.

inline const ParameterFlag &flag() const

Get the flag of the parameter.

Returns

The flag of the parameter.

inline bool has_value() const

Check whether the parameter contains a value.

Returns

true if the parameter contains a value.

inline ValueT &get()

Get the value of the parameter.

Returns

The reference to the value of the parameter.

inline std::optional<ValueT> &try_get()

Try to get the value of the parameter.

Return the reference to the std::optional value of the parameter.

Returns

The reference to the optional value of the parameter.

template<typename PointerT = ValueT, typename = std::enable_if_t<holoscan::is_shared_ptr_v<PointerT> || std::is_pointer_v<PointerT>>>
inline holoscan::remove_pointer_t<PointerT> *operator->()

Provides a pointer to the object managed by the shared pointer pointed to by the parameter value or dereferences the pointer.

Template Parameters

PointerT – The type of the pointer.

Returns

The pointer to the object managed by the shared pointer pointed to by the parameter value or the dereferenced pointer.

template<typename PointerT = ValueT, typename = std::enable_if_t<holoscan::is_shared_ptr_v<PointerT> || std::is_pointer_v<PointerT>>>
inline holoscan::remove_pointer_t<PointerT> operator*()

Provides a reference to the object managed by the shared pointer pointed to by the parameter value or dereferences the pointer.

Template Parameters

PointerT – The type of the pointer.

Returns

The reference to the object managed by the shared pointer pointed to by the parameter value or the dereferenced pointer.

inline void set_default_value()

Set the default value object if the parameter does not contain a value.

inline ValueT &default_value()

Return the default value object.

Returns

The default value object.

inline bool has_default_value() const

Check whether the parameter contains a default value.

Returns

true if the parameter contains a default value.

inline operator ValueT&()

Get the value of the argument.

Returns

The reference to the value of the parameter.