Class Timer

Table of contents

- Defined in File timer.hpp

Class Documentation

class Timer

Utility class to measure time.

This class is used to measure time. It can be used to measure the time between two points in the code, or to measure the time of a function.

The class can be used in two ways:

- Using the start() and stop() methods to measure the time between two points in the code.
- Using the constructor with the auto_start parameter set to true to measure the time of a function.

The class can also be used to print the time elapsed between two points in the code, or the time elapsed to execute a function.

The class can be used in two ways:

- Using the print() method to print the time elapsed between two points in the code.
- Using the constructor with the auto_output parameter set to true to print the time elapsed to execute a function.

Examples:

```
#include "holoscan/utils/timer.hpp" void foo() { holoscan::Timer timer("foo() took {:.8f} seconds\n"); // Do something } ...
```

```
void foo() { holoscan::Timer timer("bar() took {:.8f} seconds\n", true, false); bar(); timer.stop(); timer.print(); return 0; }
void foo() { holoscan::Timer timer("", true, false); bar(); double elapsed_time = timer.stop(); fmt::print(stderr, "bar() took {:.8f} seconds", elapsed_time);
```

Public Functions

inline explicit Timer(const char *message, bool auto_start = true, bool auto_output = true)

Construct a new Timer object.

The `message` parameter is used to print the message when the print() method is called or when the auto_output parameter is set to true and the destructor is called. The first parameter `{}` in the message will be replaced by the time elapsed.

`auto_start` is used to start the timer when the constructor is called. `auto_output` is used to print the time elapsed when the destructor is called. By default, `auto_start` and `auto_output` are set to `true`.

```
Timer timer("Time elapsed for foo() method: {:.8f} seconds\n", true, false); foo(); timer.stop(); timer.print();
```

Parameters

- **message** – The message to print when the timer is stopped.
- **auto_start** – If true, the timer is started when the constructor is called.
- **auto_output** – If true, the time elapsed is printed when the destructor is called.

inline ~Timer()

Destroy the Timer object.

If the auto_output parameter is set to true, the time elapsed is printed when the destructor is called.

inline void start()

Start the timer.

inline double stop()

Stop the timer.

Returns

The time elapsed in seconds.

inline double elapsed_time()

Return the time elapsed in seconds.

Returns

The time elapsed in seconds.

inline void print(const char *message = nullptr)

Print the time elapsed.

The message passed to the constructor is printed with the time elapsed if no `message` parameter is passed to the print() method.

The first parameter `{}` in the message will be replaced by the time elapsed.

Parameters

message – The message to print.