

std::chrono::steady_clock::now()

```
auto now = std::chrono::steady_clock::now(); // e.g. t=1'052'348'281 ns
```

Returns the current time as a `time_point` from a monotonically increasing clock — it never goes backwards, even if the system clock is adjusted. Used to measure elapsed time and set deadlines.

std::chrono::milliseconds

```
std::chrono::milliseconds ms(100); // 100 milliseconds
std::chrono::seconds s(5); // 5 seconds
// used as function arguments
sleep_for(std::chrono::milliseconds(10)); // sleep 10ms
sleep_for(std::chrono::seconds(5)); // sleep 5s
// conversion
std::chrono::milliseconds from_seconds = std::chrono::seconds(1); // 1000ms
```

Represents a duration of time in milliseconds.

Part of the `std::chrono` library which provides type-safe time durations.

Can be converted to and from other duration types like `std::chrono::seconds`.

std::chrono::time_point

```
std::chrono::time_point<std::chrono::steady_clock> tp = std::chrono::steady_clock::now();
```

Represents a specific point in time relative to a clock's epoch. You rarely construct one directly — you typically get one back from `now()` and subtract two of them to get a duration.

std::chrono::seconds

```
std::chrono::seconds timeout = std::chrono::seconds(30); // timeout.count() == 30
// Extracts the raw numeric value from a duration. Necessary because chrono deliberately hides the number
// to prevent accidental unit mismatches — you have to opt in to get it.
.timeout.count()
```

A duration type representing a number of seconds. It is a convenient alias for `std::chrono::duration<long long>`. Call `.count()` on it to get the raw integer value out.



By **blakecromar**

cheatography.com/blakecromar/

Not published yet.

Last updated 29th June, 2026.

Page 1 of 1.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>