

# Calculate Elapsed Time

## C

```
#include <time.h>

#define CLOCK_REALTIME 0

struct timespec start, end, elapsed;

clock_gettime(CLOCK_REALTIME, &start);

clock_gettime(CLOCK_REALTIME, &end);

elapsed.tv_sec = end.tv_sec - start.tv_sec;

printf("Elapsed time: %ld seconds\n", elapsed.tv_sec);
```

## C++

```
#include <chrono>

auto start = chrono::high_resolution_clock::now();

auto end = chrono::high_resolution_clock::now();

chrono::duration<double> elapsed = end - start;

cout << "Elapsed time: " << round(elapsed.count()) << " seconds" << endl;
```

## C#

```
using System.Diagnostics;

Stopwatch stopwatch = Stopwatch.StartNew();

stopwatch.Stop();

Console.WriteLine($"Elapsed time: {stopwatch.ElapsedMilliseconds / 1000} seconds");
```

## Go

```
import "time"

start := time.Now()

elapsed := time.Since(start)

fmt.Printf("Elapsed time: %d seconds\n", int(math.Round(elapsed.Seconds())))
```

## Java

```
long startTime = System.currentTimeMillis();

long endTime = System.currentTimeMillis();
long elapsedTime = (endTime - startTime) / 1000;

System.out.println(String.format("Elapsed time: %d seconds", elapsedTime));
```

## JavaScript

```
var startTime = new Date().getTime();

var endTime = new Date().getTime();
var elapsedTimeInSeconds = (endTime - startTime) / 1000;

document.getElementById('elapsedTime').innerHTML =
` ${Math.round(elapsedTimeInSeconds)} seconds `;
```

## Python

```
import time

start_time = time.time()

end_time = time.time()
elapsed_time = end_time - start_time

print(f"Elapsed time: {elapsed_time:.0f} seconds")
```

## Ruby

```
start_time = Time.now

end_time = Time.now
elapsed_time = end_time - start_time

puts "Elapsed time: #{elapsed_time.round} seconds"
```

## Rust

```
use std::time::{Duration, Instant};

let start_time: Instant = Instant::now();

let elapsed: Duration = start_time.elapsed();
let elapsed_seconds: f64 = elapsed.as_secs_f64();

println!("Elapsed time: {:.0} seconds", elapsed_seconds);
```

## Swift

```
let start_time: Date = Date()

let end_time: Date = Date()
let elapsed_time: TimeInterval = end_time.timeIntervalSince(start_time)

print("Elapsed time: \(Int(round(elapsed_time))) seconds")
```

[c](#) and [cpp](#), [dotnet](#), [go](#), [java](#), [javascript](#), [python](#), [ruby](#), [rust](#), [swift](#)

From:

<https://kbase.devtoprd.com/> - **Knowledge Base**

Permanent link:

[https://kbase.devtoprd.com/doku.php?id=calculate\\_elapsed\\_time](https://kbase.devtoprd.com/doku.php?id=calculate_elapsed_time)

Last update: **2024/08/21 11:21**

