

CPU Deterministic Profilers- Program level

Name: %time
Description: Time the execution of a single statement
Installation: Not needed

```
%%time
lst = list(range(3000000))
res = sum(lst)

CPU times: user 75.9 ms, sys: 4.55 ms, total: 80.4 ms
Wall time: 79.7 ms
```

CPU Deterministic Profilers-Function level

Name: %prun
Description: Run code with the profiler
Installation: Not needed

```
%prun
lst = list(range(3000000))
res = sum(lst)

function calls in 0.083 seconds
by: internal time
tottime  percall  ctime  percall  filename:lineno(function)
0.067  0.067  0.083  0.083  <string>:1(<module>)
0.016  0.016  0.016  0.016  (built-in method builtins.sum)
0.000  0.000  0.083  0.083  (built-in method builtins.exec)
0.000  0.000  0.000  0.000  {method 'disable' of '_lsprof.Profiler' objects}
```

CPU Deterministic Profilers- Line level

Name: %lprun
Description: Run code with the line byline profiler
Installation: pip install line_profiler

```
%load_ext line_profiler

%lprun
lst = list(range(3000000))
res = sum(lst)
```

Memory Deterministic Profiler- Program Level

Name: %memit
Description: Measure the memory use of a single statement
Installation: pip install memory_profiler

```
%memit
lst = list(range(3000000))
res = sum(lst)

peak memory: 1311.73 MiB, increment: 0.00 MiB
```

Memory Deterministic Profiler-Function level

```
In [15]: %load_ext memory_profiler

In [16]: %memit
a = sum(list(range(3000000)))

peak memory: 796.88 MiB, increment: 0.33 MiB
```

Memory Profiler- Deterministic Line Level

Name: %mprun
Description: Run code with the line by line memory profiler
Installation: pip install memory_profiler

```
%mprun
lst = list(range(3000000))
res = sum(lst)
```

Profilers Visualizers

Name: %snakeviz
Description: Run code with the line by line memory profiler
Installation: pip install snakeviz

```
%load_ext snakeviz

%snakeviz
lst = list(range(3000000))
res = sum(lst)
```

CPU Statistical Profilers- Line level

Name: vmpfrof
Description: Run code with the line by line memory profiler in statistical manner
Installation: pip install vmpfrof

```
python -m vmpfrof --lines -o <output-file> <your program> <your program args>
```

Tips

Tips

All profilers add overhead

Statistical Profilers add less overhead

Method level profiling is not always enough



By Eyal Trabelsi (Eyaltra)
cheatography.com/eyaltra/

Published 10th October, 2020.
 Last updated 10th October, 2020.
 Page 1 of 1.

Sponsored by [ApolloPad.com](https://apollopod.com)
 Everyone has a novel in them. Finish Yours!
<https://apollopod.com>