

Timing

At the command line:

```
In [1]: !time python slow_functions.py
```

```
10.7445980559
```

```
real    0m2.545s
user    0m2.476s
sys     0m0.030s
```

```
In [2]: import slow_functions
```

With the time module:

```
In [4]: import time
```

```
In [5]: t1 = time.time()
result = slow_functions.main()
t2 = time.time()
print 'slow_functions.main took {} seconds'.format(t2 - t1)
```

```
slow_functions.main took 2.52013397217 seconds
```

IPython magic: %run

```
In [6]: %run -t slow_functions.py
```

```
10.7445980559
```

```
IPython CPU timings (estimated):
  User    :      2.43 s.
  System  :      0.01 s.
  Wall time:      2.44 s.
```

lpython magic: %timeit

```
In [7]: %timeit slow_functions.main()
```

```
1 loops, best of 3: 2.67 s per loop
```

```
In [8]: import numpy as np
from numpy import interp
from scipy.interpolate import interp1d
```

```
In [9]: x = np.linspace(0, 2*np.pi, 10)
y = np.sin(x)
xvals = np.linspace(0, 2*np.pi, 50)
```

```
In [10]: #scipy
f = interp1d(x, y)
%timeit f(xvals)
```

10000 loops, best of 3: 92.8 us per loop

```
In [11]: # numpy
%timeit interp(xvals, x, y)
```

100000 loops, best of 3: 3.93 us per loop

```
In [12]: scipy_vals = f(xvals)
numpy_vals = interp(xvals, x, y)
print np.allclose(scipy_vals, numpy_vals)
```

True

Profiling

At the command line:

```
In [13]: !python -m cProfile slow_functions.py
```

10.7445980559

2019 function calls in 2.426 seconds

Ordered by: standard name

ncalls	tottime	percall	cumtime	percall	filename:lineno(function)
1	0.003	0.003	2.426	2.426	
slow_functions.py:1(<module>)					
1001	0.780	0.001	0.780	0.001	slow_functions.py:11(func2)
1001	0.779	0.001	0.779	0.001	slow_functions.py:19(func3)
1	0.000	0.000	0.781	0.781	slow_functions.py:27(func4)
11	0.862	0.078	0.862	0.078	slow_functions.py:3(func1)
1	0.002	0.002	1.560	1.560	slow_functions.py:34(func5)
1	0.000	0.000	2.423	2.423	slow_functions.py:41(main)
1	0.000	0.000	0.000	0.000	{math.log10}
1	0.000	0.000	0.000	0.000	{method 'disable' of
'_lsprof.Profiler' objects}					

Viewing with [pstats](#)

```
In [14]: !python -m cProfile -o slow_functions.prof slow_functions.py
```

```
10.7445980559
```

```
In [15]: import pstats
```

```
In [16]: stats = pstats.Stats('slow_functions.prof')
```

```
In [17]: stats.print_stats()
```

```
Thu Jul 5 12:21:54 2012    slow_functions.prof
```

```
2019 function calls in 2.360 seconds
```

```
Random listing order was used
```

ncalls	tottime	percall	cumtime	percall	filename:lineno(function)
11	0.837	0.076	0.837	0.076	slow_functions.py:3(func1)
1	0.003	0.003	2.360	2.360	
slow_functions.py:1(<module>)					
1	0.002	0.002	1.518	1.518	slow_functions.py:34(func5)
1	0.000	0.000	0.761	0.761	slow_functions.py:27(func4)
1001	0.757	0.001	0.757	0.001	slow_functions.py:11(func2)
1	0.000	0.000	0.000	0.000	{math.log10}
1	0.000	0.000	2.357	2.357	slow_functions.py:41(main)
1	0.000	0.000	0.000	0.000	{method 'disable' of
'_lsprof.Profiler' objects}					
1001	0.762	0.001	0.762	0.001	slow_functions.py:19(func3)

```
Out[17]: <pstats.Stats instance at 0x104b4b710>
```

```
In [18]: stats.sort_stats('cum').print_stats()
```

```
Thu Jul 5 12:21:54 2012    slow_functions.prof
```

```
2019 function calls in 2.360 seconds
```

```
Ordered by: cumulative time
```

ncalls	tottime	percall	cumtime	percall	filename:lineno(function)
1	0.003	0.003	2.360	2.360	
slow_functions.py:1(<module>)					
1	0.000	0.000	2.357	2.357	slow_functions.py:41(main)
1	0.002	0.002	1.518	1.518	slow_functions.py:34(func5)
11	0.837	0.076	0.837	0.076	slow_functions.py:3(func1)

```

1001    0.762    0.001    0.762    0.001 slow_functions.py:19(func3)
   1    0.000    0.000    0.761    0.761 slow_functions.py:27(func4)
1001    0.757    0.001    0.757    0.001 slow_functions.py:11(func2)
   1    0.000    0.000    0.000    0.000 {math.log10}
   1    0.000    0.000    0.000    0.000 {method 'disable' of
'_lsprof.Profiler' objects}

```

Out[18]: <pstats.Stats instance at 0x104b4b710>

In [19]: stats.sort_stats('time').print_stats()

```

Thu Jul  5 12:21:54 2012    slow_functions.prof

2019 function calls in 2.360 seconds

Ordered by: internal time

ncalls  tottime  percall  cumtime  percall filename:lineno(function)
   11    0.837    0.076    0.837    0.076 slow_functions.py:3(func1)
  1001    0.762    0.001    0.762    0.001 slow_functions.py:19(func3)
  1001    0.757    0.001    0.757    0.001 slow_functions.py:11(func2)
     1    0.003    0.003    2.360    2.360
slow_functions.py:1(<module>)
     1    0.002    0.002    1.518    1.518 slow_functions.py:34(func5)
     1    0.000    0.000    2.357    2.357 slow_functions.py:41(main)
     1    0.000    0.000    0.761    0.761 slow_functions.py:27(func4)
     1    0.000    0.000    0.000    0.000 {math.log10}
     1    0.000    0.000    0.000    0.000 {method 'disable' of
'_lsprof.Profiler' objects}

```

Out[19]: <pstats.Stats instance at 0x104b4b710>

In [20]: stats.sort_stats('time').print_stats(3)

```

Thu Jul  5 12:21:54 2012    slow_functions.prof

2019 function calls in 2.360 seconds

Ordered by: internal time
List reduced from 9 to 3 due to restriction <3>

ncalls  tottime  percall  cumtime  percall filename:lineno(function)
   11    0.837    0.076    0.837    0.076 slow_functions.py:3(func1)
  1001    0.762    0.001    0.762    0.001 slow_functions.py:19(func3)
  1001    0.757    0.001    0.757    0.001 slow_functions.py:11(func2)

```

Out[20]: <pstats.Stats instance at 0x104b4b710>

In [21]: stats.sort_stats('cum').print_stats(r'func\d')

```
Thu Jul 5 12:21:54 2012    slow_functions.prof
```

```
2019 function calls in 2.360 seconds
```

```
Ordered by: cumulative time
```

```
List reduced from 9 to 5 due to restriction <'func\\d'>
```

ncalls	tottime	percall	cumtime	percall	filename:lineno(function)
1	0.002	0.002	1.518	1.518	slow_functions.py:34(func5)
11	0.837	0.076	0.837	0.076	slow_functions.py:3(func1)
1001	0.762	0.001	0.762	0.001	slow_functions.py:19(func3)
1	0.000	0.000	0.761	0.761	slow_functions.py:27(func4)
1001	0.757	0.001	0.757	0.001	slow_functions.py:11(func2)

```
Out[21]: <pstats.Stats instance at 0x104b4b710>
```

```
In [22]: stats.sort_stats('time').print_stats(r'func\d', 3)
```

```
Thu Jul 5 12:21:54 2012    slow_functions.prof
```

```
2019 function calls in 2.360 seconds
```

```
Ordered by: internal time
```

```
List reduced from 9 to 5 due to restriction <'func\\d'>
```

```
List reduced from 5 to 3 due to restriction <3>
```

ncalls	tottime	percall	cumtime	percall	filename:lineno(function)
11	0.837	0.076	0.837	0.076	slow_functions.py:3(func1)
1001	0.762	0.001	0.762	0.001	slow_functions.py:19(func3)
1001	0.757	0.001	0.757	0.001	slow_functions.py:11(func2)

```
Out[22]: <pstats.Stats instance at 0x104b4b710>
```

```
More IPython magic: %prun
```

```
In [23]: %prun slow_functions.main()
```

```
In [24]: %prun -D slow_functions_main.prof slow_functions.main()
```

```
*** Profile stats marshalled to file u'slow_functions_main.prof'.
```

```
In [25]: stats = %prun -q -r f(xvals)
```

```
In [26]: # work around a bug in IPython
import sys
stats.stream = sys.stdout
```

```
In [27]: stats.sort_stats('time').print_stats(10)
```

17 function calls in 0.001 seconds

Ordered by: internal time

List reduced from 15 to 10 due to restriction <10>

ncalls	tottime	percall	cumtime	percall	filename:lineno(function)
1	0.000	0.000	0.000	0.000	
1	0.000	0.000	0.001	0.001	interpolate.py:286(_call_linear)
1	0.000	0.000	0.000	0.000	interpolate.py:338(__call__)
1	0.000	0.000	0.000	0.000	interpolate.py:391(_check_bounds)
1	0.000	0.000	0.000	0.000	fromnumeric.py:756(searchsorted)
2	0.000	0.000	0.000	0.000	{method 'any' of 'numpy.ndarray' objects}
1	0.000	0.000	0.001	0.001	<string>:1(<module>)
1	0.000	0.000	0.000	0.000	{method 'clip' of 'numpy.ndarray' objects}
1	0.000	0.000	0.000	0.000	{method 'searchsorted' of 'numpy.ndarray' objects}
1	0.000	0.000	0.000	0.000	numeric.py:216(asarray)
1	0.000	0.000	0.000	0.000	{method 'astype' of 'numpy.ndarray' objects}

```
Out[27]: <pstats.Stats instance at 0x104b50f80>
```

```
In [ ]: %%prun
f(xvals)
interp(xvals, x, y)
```

[line_profiler](#) with IPython magic

```
In [28]: %load_ext line_profiler
```

```
In [29]: %lprun -f slow_functions.main slow_functions.main()
```

```
In [30]: %lprun -f slow_functions.func5 slow_functions.main()
```

```
In [31]: %lprun -f f.__call__ f(xvals)
```

RunSnakeRun

```
In [32]: %prun -q -D scipy_interp.prof f(xvals)
```

```
*** Profile stats marshalled to file u'scipy_interp.prof'.
```

```
In [33]: %prun -q -D numpy_interp.prof interp(xvals, x, y)
```

```
*** Profile stats marshalled to file u'numpy_interp.prof'.
```

```
In [ ]:
```