

Arithmetic Operation: summing up all numbers from 1 to 100000000 (100 million).

- JavaScript:

```
sum_i=0^100 = 5000000050000000  
Doing arithmetic adding for 100000000 times took 666.985 milliseconds.  
Example of Javascript arithmetic.
```

- Java:

```
LiuZhimengs-MacBook-Pro:HW5 YvaineLiu$ time java SpeedTestInt  
a = 987459712  
100000000 iterations, 41 milliseconds ==> 0 ns/iteration  
  
real    0m0.114s  
user    0m0.095s  
sys     0m0.018s
```

- Python:

```
LiuZhimengs-MacBook-Pro:HW5 YvaineLiu$ time python3 arithTest.py  
5000000050000000  
  
100000000 iterations, 11356.333 milliseconds 113.56333 ns/iteration  
  
real    0m11.416s  
user    0m11.364s  
sys     0m0.020s  
LiuZhimengs-MacBook-Pro:HW5 YvaineLiu$ time python3 arithTest.py  
5000000050000000  
  
100000000 iterations, 12243.227 milliseconds 122.43227 ns/iteration  
  
real    0m12.276s  
user    0m12.260s  
sys     0m0.009s
```

- Conclusion:

By running 100 million iterations of pure arithmetic operations, it turns out that JavaScript is slower than Java but still much faster than Python. I know that JS is also an interpreted language like Python, but I guess it might have retained some of Java's features when running large numbers of logic/iterations.

More specifically, when the number of iterations is large enough, it took Java a little more than 0.1s to run, but it took JavaScript about 0.67s, which is about 60 times slower than Java. For Python, 100 million iterations took more than 12s, which is 120 times slower than Java, and about 20 times slower than JavaScript.