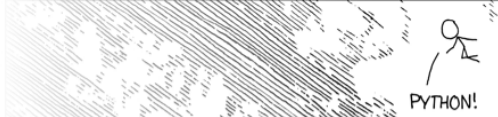


PYTHON MÁS RÁPIDO QUE C

SIN HACER TRAMPA!

PRESENTACIÓN



LES HABLA: ROBERTO ALSINA.

EMAIL: ROBERTO.ALSINA@GMAIL.COM

TWITTER: @RALSINA

PÁGINA: [HTTPS://RALSINA.ME](https://ralsina.me)

ARRANCAMOS CON PYTHON MALO



COMO ESTE QUE ME DIO JUAN PEDRO FISANOTTI:

```
def fibonacci(n: int) -> int:
    if n in (1, 2):
        return 1
    else:
        return fibonacci(n - 1) + fibonacci(n - 2)
```

Y TENEMOS EL EQUIVALENTE EN C



```
int fibonacci(int n) {  
    if(n == 2 || n == 1){  
        return 1;  
    } else {  
        return (fibonacci(n-1) + fibonacci(n-2));  
    }  
}
```

¿ES MÁS RÁPIDO PYTHON O C?



```
gcc fibonacci.c -o fibonacci_en_c  
time ./fibonacci_en_c 35
```

- EN C TARDA 0.07 SEGUNDOS PARA CALCULAR fibonacci(35)

¿ES MÁS RÁPIDO PYTHON O C?



```
time python -c "from fibonacci import fibonacci; print(fibonacci(35))"
```

- EN PYTHON TARDA 3.33 SEGUNDOS,

¿ES MÁS RÁPIDO PYTHON O C?



- EN C TARDA 0.07 SEGUNDOS PARA CALCULAR `fibonacci(35)`
- EN PYTHON TARDA 3.33 SEGUNDOS,
- C ES ~50 VECES MÁS RÁPIDO

¡ PODRÍAMOS ARREGLAR EL PYTHON MALO!



- NO TIENE GRACIA

¡ PODRÍAMOS ARREGLAR EL PYTHON MALO!



- NO TIENE GRACIA
- TAMBIÉN PODRÍAMOS ARREGLAR EL C MALO

¡ PODRÍAMOS ARREGLAR EL PYTHON MALO!



- NO TIENE GRACIA
- TAMBIÉN PODRÍAMOS ARREGLAR EL C MALO
- ESTA CHARLA ES SOBRE HACER QUE TU CÓDIGO SEA RÁPIDO, NO BUENO

¡ YA SÉ, MYPYC!



```
mypyc fibonacci.py
```

- EN C TARDA 0.07 SEGUNDOS PARA CALCULAR fibonacci(35)

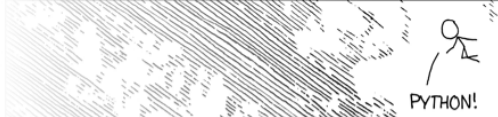
¡ YA SÉ, MYPYC!



```
mypyc fibonacci.py
```

- EN C TARDA 0.07 SEGUNDOS PARA CALCULAR `fibonacci(35)`
- CÓDIGO COMPILADO CON MYPYC TARDA 1.57 SEGUNDOS

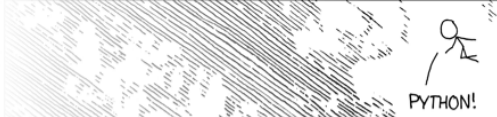
¡ YA SÉ, MYPYC!



```
mypyc fibonacci.py
```

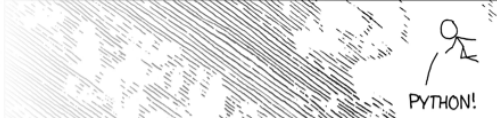
- EN C TARDA 0.07 SEGUNDOS PARA CALCULAR `fibonacci(35)`
- CÓDIGO COMPILADO CON MYPYC TARDA 1.57 SEGUNDOS
- C ES 22 VECES MÁS RÁPIDO

¡ YA SÉ, PYPY!



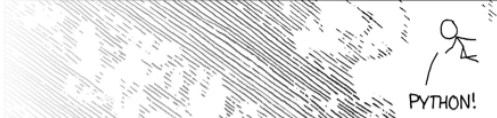
- EN C TARDA 0.07 SEGUNDOS PARA CALCULAR `fibonacci(35)`

¡ YA SÉ, PYPY!



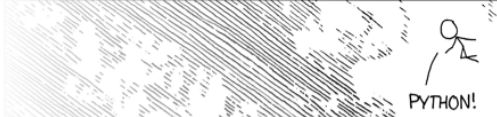
- EN C TARDA 0.07 SEGUNDOS PARA CALCULAR `fibonacci(35)`
- PYPY TARDA 0.36 SEGUNDOS

¡ YA SÉ, PYPY!



- EN C TARDA 0.07 SEGUNDOS PARA CALCULAR `fibonacci(35)`
- PYPY TARDA 0.36 SEGUNDOS
- C ES ~5 VECES MÁS RÁPIDO

¡ YA SÉ, CYTHON!



```
cythonize -3 -i fibonaccy.py
```

- EN C TARDA 0.07 SEGUNDOS PARA CALCULAR
fibonacci(35)

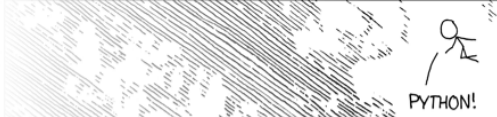
¡ YA SÉ, CYTHON!



```
cythonize -3 -i fibonaccy.py
```

- EN C TARDA 0.07 SEGUNDOS PARA CALCULAR `fibonacci(35)`
- CÓDIGO COMPILADO CON CYTHON TARDA 0.05 SEGUNDOS

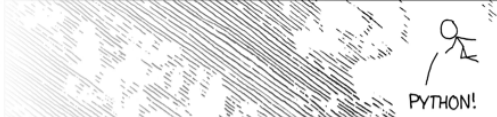
¡ YA SÉ, CYTHON!



```
cythonize -3 -i fibonaccy.py
```

- EN C TARDA 0.07 SEGUNDOS PARA CALCULAR `fibonacci(35)`
- CÓDIGO COMPILADO CON CYTHON TARDA 0.05 SEGUNDOS
- C ES 40% MÁS LENTO

¡ YA SÉ, CYTHON!



```
cythonize -3 -i fibonaccy.py
```

- EN C TARDA 0.07 SEGUNDOS PARA CALCULAR `fibonacci(35)`
- CÓDIGO COMPILADO CON CYTHON TARDA 0.05 SEGUNDOS
- C ES 40% MÁS LENTO
- SIEMPRE QUE TE OLVIDES DE USAR LA OPCIÓN `-o`