# Command Line Interface

```python
"-".join(str(n) for n in range(100))
```

```
'"-".join([str(n) for n in range(100)])'
```

```
'"-".join(map(str, range(100)))'
```

# Python Interface

```
'"-".join(str(n) for n in range(100))'    10000
```

```
# => 0.30186118488820001
```

```
'"-".join([str(n) for n in range(100)])'          10000
```

```
# => 0.2727368790656328
```

```
'"-".join(map(str, range(100)))'                    10000
```

```
# => 0.23702679807320237
```

# timeit - time short snippets

```
# Command Line Interface
(cs41-env)$ python -m timeit '"-".join(str(n) for n in range(100))'
10000 loops, best of 3: 30.2 usec per loop
(cs41-env)$ python -m timeit '"-".join([str(n) for n in range(100)])'
10000 loops, best of 3: 27.5 usec per loop
(cs41-env)$ python -m timeit '"-".join(map(str, range(100)))'
10000 loops, best of 3: 23.2 usec per loop

# Python Interface
import timeit
timeit.timeit('"-".join(str(n) for n in range(100))', number=10000)
# => 0.3018611848820001
timeit.timeit('"-".join([str(n) for n in range(100)])', number=10000)
# => 0.2727368790656328
timeit.timeit('"-".join(map(str, range(100)))', number=10000)
# => 0.23702679807320237
```

# "Cute" Modules

I couldn't resist!