

# Python Powerful One Liner Guide

One of the most useful things in Python is the ability to write elegant short code. Most are readable, some may not be so readable, but all can save a lot of time! Here are our favourites:

## 1. One liner IF statement

Normal IF statement:

```
x = 10
if x > 8:
    print("x is big")
```

One liner:

```
x = 10
if x > 8: print("x is big")
```

## 2. One liner IF-Else statement

Normal IF statement:

```
year = 2021
if year > 2000:
    print("21st century")
else:
    print("previous century")
```

One liner:

```
year = 2021
print("21st century") if year > 2000 else print("previous century")
```

### 3. One liner FOR statement

Normal FOR statement:

```
items = ["apple", "banana", "carrot"]  
  
for item in items:  
    print(item)
```

One liner:

```
items = ["apple", "banana", "carrot"]  
  
[ print(item) for item in items]
```

### 4. One liner FOR statement with IF condition

Normal FOR statement with an IF condition on the iteration:

```
items = ["apple", "banana", "carrot", "pineapple"]  
  
for item in items:  
    if "apple" in item:  
        print(item)
```

One liner version:

```
items = ["apple", "banana", "carrot", "pineapple"]  
  
[ print(item) for item in items if "apple" in item]
```

## 5. One liner nested FOR statement

Normal FOR statement with an IF condition on the iteration:

```
items = {"fruits": ["apple", "banana", "pineapple"],
         "vegetables": ["potato", "cabbage", "spinach"],
         "deserts": ["ice cream", "pudding", "pie"] }

for key in items.keys():
    for item in items[key]:
        print( item)
```

One liner version:

```
items = {"fruits": ["apple", "banana", "pineapple"],
         "vegetables": ["potato", "cabbage", "spinach"],
         "deserts": ["ice cream", "pudding", "pie"] }

[ print(item) for key in items.keys() for item in items[key] ]
```

## 6. Read File with One Line

Normal:

```
f= open( "data.txt", "r")
lines = f.readlines()

print(lines)
```

One liner version:

```
lines = [line.strip() for line in open( "data.txt" )]

print(lines)
```

## 7. Aggregate list with a single line

Normal:

```
items = [ 1, 2, 3, 4, 5]

total = 0
for item in items:
    total = total + item

print(total)
```

One liner version:

```
from functools import reduce

total = reduce((lambda x, y: x + y), items )

print(total)
```

## 8. Print repeated pattern in one line

Normal:

```
for count in range(80):
    print("x", end="")
```

One liner version:

```
print("x" * 80 )
```

## 9. Swap Variables in One Line

Normal:

```
a = 10
b = 20
temp = a

a = b
b = temp

print(a, b)
```

One liner version:

```
a = 10
b = 20

a, b = b, a

print(a, b)
```

## 10. Test Multiple conditions

Normal:

```
var = 'a'

if var == 'a' or var == 'b' or var == 'c':
    print("in list")
```

One liner version:

```
var = 'a'

if var in ['a', 'b', 'c']: print("in list")
```