

One-Liners

1) Multiple Variable Assignment

```
# Traditional way
a = 1
b = "ok"
c = False

# Pythonic way
a, b, c = 1, "ok", False

# Result
print(a, b, c)

# Show: 1 ok False
```

2) Variable Swap

```
# Traditional way
a = 1
b = "ok"

c = a
a = b
b = c

# Pythonic way
a, b = 1, "ok"
a, b = b, a

# Result
print(a, b)

# Shows: ok 1
```

```
# Pythonic way
a, b, c, d = 1, "ok", True, ["i", "j"]
```

```
a, b, c, d = c, a, d, b
```

```
# Result
```

```
print(a, b, c, d)
```

```
# Shows: True 1 ["i", "j"] ok
```

3) Variable Conditional Assignment

```
x = 3
```

```
# Traditional way
```

```
if x % 2 == 1:
```

```
    result = f"{x} is odd"
```

```
else:
```

```
    result = f"{x} is even"
```

```
# Pythonic way
```

```
result = f"{x} " + ("is odd" if x % 2 == 1 else "is even")
```

```
# Result
```

```
print(result)
```

```
# Shows: 3 is odd
```

4) Presence of a Value in a List

```
pet_list = ["cat", "dog", "parrot"]
```

```
# Traditional way
```

```
found = False
```

```
for item in my_list:
```

```
    if item == "cat":
```

```
        found = True
```

```
        break
```

```
# Pythonic way
```

```
found = "cat" in pet_list
```

```
# Result
```

```
print(found)
```

```
# Shows: True
```

```
pet_dict = {"cat": "Mitchi", "dog": "Max", "parrot": "Pepe"}  
found = "cat" in pet_dict  
print(found)  
# Shows: True
```

5) Operations on Lists

```
my_list = [1, 2, 3, 4, 5]  
  
# Traditional way  
max_value = 0  
for value in my_list:  
    if value > max_value:  
        max_value = value  
  
# Pythonic way  
max_value = max(my_list)  
  
# Result  
print(max_value)  
# Shows: 5
```

6) List Creation with Duplicate Values

```
size = 10  
  
# Traditional way  
my_list = []  
for i in range(size):  
    my_list.append(0)  
  
# Pythonic way  
my_list = [0] * size  
  
# Result  
print(my_list)  
# Shows: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

```
my_list = [1, 2] * 5
```

```
# Result: [1, 2, 1, 2, 1, 2, 1, 2, 1, 2]
```

```
my_tuple = (1, 2) * 5
```

```
print(my_tuple)
```

```
# Shows: (1, 2, 1, 2, 1, 2, 1, 2, 1, 2)
```

7) List Creation with Sequential Values

```
count = 10
```

```
# Traditional way
```

```
my_list = []
```

```
for i in range(count):
```

```
    my_list.append(i)
```

```
# Pythonic way
```

```
my_list = list(range(count))
```

```
# Result
```

```
print(my_list)
```

```
# Shows: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
# List with odd values
```

```
my_list = list(range(1, 10, 2))
```

```
print(my_list)
```

```
# Shows: [1, 3, 5, 7, 9]
```

```
# List with descending values and negative values
```

```
my_list = list(range(5, -5, -1))
```

```
print(my_list)
```

```
# Shows: [5, 4, 3, 2, 1, 0, -1, -2, -3, -4]
```

```
my_set = set(range(count))
```

```
my_tuple = tuple(range(count))
```

```
# Result
```

```
print(my_set)
```

```
# Shows: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
print(my_tuple)
# Shows: (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
```

8) List Creation with a Loop

```
count = 4

# Traditional way
my_list = []
for i in range(count):
    my_list.append(count**i)

# Pythonic way
my_list = [count**x for x in range(count)]

# Result
print(my_list)
# Shows: [1, 4, 16, 64]
```

```
my_set = set(count**x for x in range(count))
print(my_set)
# Shows: {1, 4, 16, 64}
```

9) List Creation with Conditions

```
users = [("Megan", 56),
("Karen", 32),
("Chad", 28),
("Brent", 44)]

# Traditional way
young_users = []
for user in users:
    if (user[1] < 35):
        young_users.append(user[0])

# Pythonic way
young_users = [x for x, y in users if y < 35]
```

```
# Result
```

```
print(young_users)
```

```
# ["Karen", "Chad"]
```

10) Reading a File Line by Line

```
# Traditional way
```

```
lines = []
```

```
with open(filename) as file:
```

```
for count, line in enumerate(file):
```

```
lines.append(f"Line {count + 1}: " + line.strip())
```

```
# Pythonic way
```

```
with open(filename) as file:
```

```
lines = [f"Line {count + 1}: " + line.strip() for count, line in enumerate(file)]
```

Revision #5

Created 19 June 2023 17:41:20 by Admin

Updated 20 June 2023 10:39:12 by Admin