5 Ways to Make Your Python Code Faster

Use Built-in Functions

```
In [5]:  import time
```

```
In [6]:  # Slow:

         start = time.time()

         new_list = []
         for word in word_list:
             new_list.append(word.capitalize())

         print(time.time() - start, "seconds")
```
4.935264587402344e-05 seconds

```
In [7]:  #fast

         start = time.time()

         word_list = ['Ways', 'to', 'Make', 'Your', 'Python', 'Code', 'Faster']
         new_list = list(map(str.capitalize, word_list))

         print(time.time() - start, "seconds")
```
0.00029468536376953125 seconds

String Concatenation vs join()

```
In [9]:  # Slow:

         start = time.time()

         new_list = []
         for word in word_list:
             new_list += word

         print(time.time() - start, "seconds")
```
5.7220458984375e-05 seconds

```
In [10]:  #fast

          start = time.time()

          word_list = ['Ways', 'to', 'Make', 'Your', 'Python', 'Code', 'Faster']
          new_list = "".join(word_list)

          print(time.time() - start, "seconds")
```
0.0001430511474609375 seconds

Create Lists and Dictionaries Faster

```python
In [11]: #slow

         list()
         dict()
```

Out[11]: {}

```python
In [12]: # fast

         ()
         {}
```

Out[12]: {}

```python
In [19]: import timeit

         slower_list = timeit.timeit("list()", number=10**6)
         slower_dict = timeit.timeit("dict()", number=10**6)

         faster_list = timeit.timeit("[]", number=10**6)
         faster_dict = timeit.timeit("{}", number=10**6)

         print("slower_list:",slower_list, "sec")   #Should have used f string here..
         print("slower_dict:",slower_dict, "sec")
         print("faster_list:",faster_list, "sec")
         print("faster_dict:",faster_dict, "sec")
```

```
slower_list: 0.03651356499995018 sec
slower_dict: 0.047055111999952715 sec
faster_list: 0.010666104999927484 sec
faster_dict: 0.010838708999926894 sec
```

f-Strings

```python
In [22]: #slow
         start = time.time()

         me = "Python"
         string = "Make " + me + " faster"

         print(time.time() - start, "seconds")
```

```
4.506111145019531e-05 seconds
```

```python
In [23]: #fast
         start = time.time()

         me = "Python"
         string = f"Make {me} faster"

         print(time.time() - start, "seconds")
```

```
0.00016546249389648438 seconds
```

List Comprehensions

```python
In [24]: #slow
         start = time.time()

         new_list = []
```

```python
existing_list = range(1000000)
for i in existing_list:
    if i % 2 == 1:
        new_list.append(i)

print(time.time() - start, "seconds")
```

0.06872344017028809 seconds

In [25]:
```python
#fast
start = time.time()

existing_list = range(1000000)
new_list = [i for i in existing_list if i % 2 == 1]

print(time.time() - start, "seconds")
```

0.04211759567260742 seconds

Bonus: Check out more Python Built-in Functions here

https://docs.python.org/3/library/functions.html

In [ ]: