



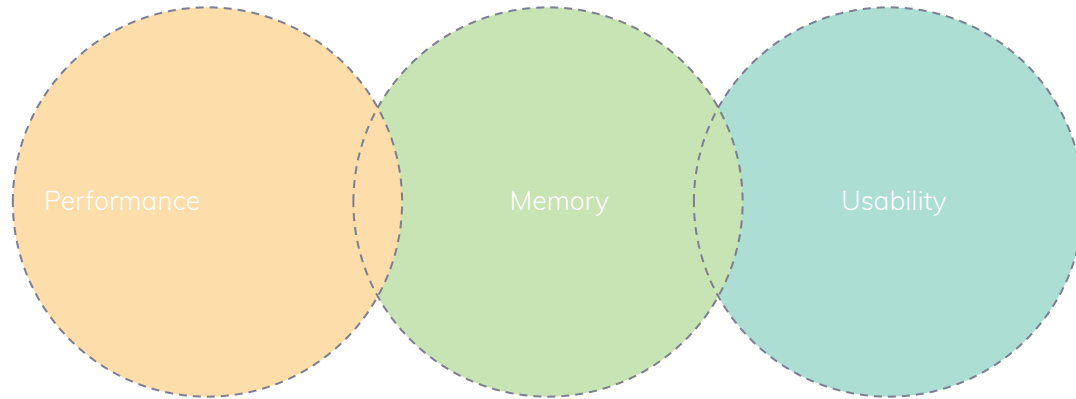
DATUM 2020-09-09



WHAT MAKES AN EFFICIENT CODE?



## PYTHON PRACTICES FOR EFFICIENT CODES



## USE LIST COMPREHENSIONS

- use loop comprehensions

Ex) You want to find the cubes of all numbers in a given range

```
cube_numbers = []  
for n in range(0,10):  
    if n % 2 == 1:  
        cube_numbers.append(n**3)
```

## USE LIST COMPREHENSIONS

list comprehension

- Faster
- Short & more Concise

```
cube_numbers = [n**3 for n in range(1,10) if n%2 == 1]
```

## SETS AND UNIONS

- using too much loops puts unnecessary strain

ex) print

Overlapping

Values in set

A&B

```
a = [1,2,3,4,5]
b = [2,3,4,5,6]

overlaps = []
for x in a:
    for y in b:
        if x==y:
            overlaps.append(x)

print(overlaps)
```

## SETS AND UNIONS

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

```
b = [2, 3, 4, 5, 6]
```

```
overlaps = set(a) & set(b)
```

```
print(overlaps)
```

## MULTIPLE ASSIGNMENT

- remember to use multiple assignment
- Swap values of variables

```
first_name, last_name, city = "Kevin", "Cunningham", "B  
righton"
```

```
x, y = y, x
```



## MULTIPLE ASSIGNMENT

- use swap method instead of this approach:

```
temp = x
```

```
x = y
```

```
y = temp
```

```
x, y = y, x
```

## USE JOIN() TO CONCATENATE STRINGS

- use array module to modify character instead of join()

```
new = "This" + "is" + "going" + "to" + "require" + "a"  
+ "new" + "string" + "for" + "every" + "word"  
print(new)
```

```
Thisisgoingtorequireanewstringforeveryword
```

## USE JOIN() TO CONCATENATE STRINGS

- this does not involve creating new string and copying old content at each step

```
new = " ".join(["This", "will", "only", "create", "one",  
               , "string", "and", "we", "can", "add", "spaces."])  
print(new)
```

This will only create one string and we can add spaces.

## ROCK PAPER SCISSORS LIZARD SPOCK

- As you can see, there are a total of 5 options ( $X \rightarrow Y$  means  $X$  wins over  $Y$ ):
  - Rock  $\rightarrow$  Lizard & Scissors
  - Paper  $\rightarrow$  Rock & Spock
  - Scissors  $\rightarrow$  Paper & Lizard
  - Lizard  $\rightarrow$  Spock & Paper
  - Spock  $\rightarrow$  Scissors & Rock

```
def give_winner(first_selection, second_selection):  
    if(first_selection is 1):  
        if(second_selection is 3 or second_selection is 4):  
            return True  
    elif(first_selection is 2):  
        if(second_selection is 1 or second_selection is 5):  
            return True  
    elif(first_selection is 3):  
        if(second_selection is 2 or second_selection is 4):  
            return True  
    elif(first_selection is 4):  
        if(second_selection is 2 or second_selection is 5):  
            return True  
    elif(first_selection is 5):  
        if(second_selection is 3 or second_selection is 1):  
            return True  
    return False
```

# LAMBDA

```
# Program to show the use of lambda functions
double = lambda x: x * 2

print(double(5))
```

## Output

10

```
double = lambda x: x * 2
```

is nearly the same as:

```
def double(x):
    return x * 2
```

```
def give_winner(first_selection, second_selection):  
    rules = {  
        1: lambda x: x in (3, 4),  
        2: lambda x: x in (1, 5),  
        3: lambda x: x in (2, 4),  
        4: lambda x: x in (2, 5),  
        5: lambda x: x in (3, 1)  
    }  
    return rules[first_selection](second_selection)
```

```
dictionary = {  
    1: [3, 4],  
    2: [1, 5],  
    3: [2, 4],  
    4: [2, 5],  
    5: [3, 1]  
}  
  
def give_winner(first_selection, second_selection):  
    if dictionary.has_key(first_selection):  
        if second_selection in dictionary[first_selection]:  
            return True  
    return False
```





Trade your code with  
members of your group and  
talk about places where it can  
be more efficient!





“

Brainstorm projects you want  
to do with your teammates!

— • —

# 동준's BOOTCAMP

If you feel like your Python is rusty or having difficulties with the activities in the Academia, please fill out the form to attend extra sessions to help with Python!

<https://forms.gle/hptJiRmaMteniNAKA>

## 동준's Bootcamp

If you feel like you need additional help with Python, please sign up with the sheet below :)  
The sessions will be held by voting in everyone's available times!

Name

Your answer

Group

- ☐ programming
- ☐ data science

What do you feel like you have difficulties with?

Your answer