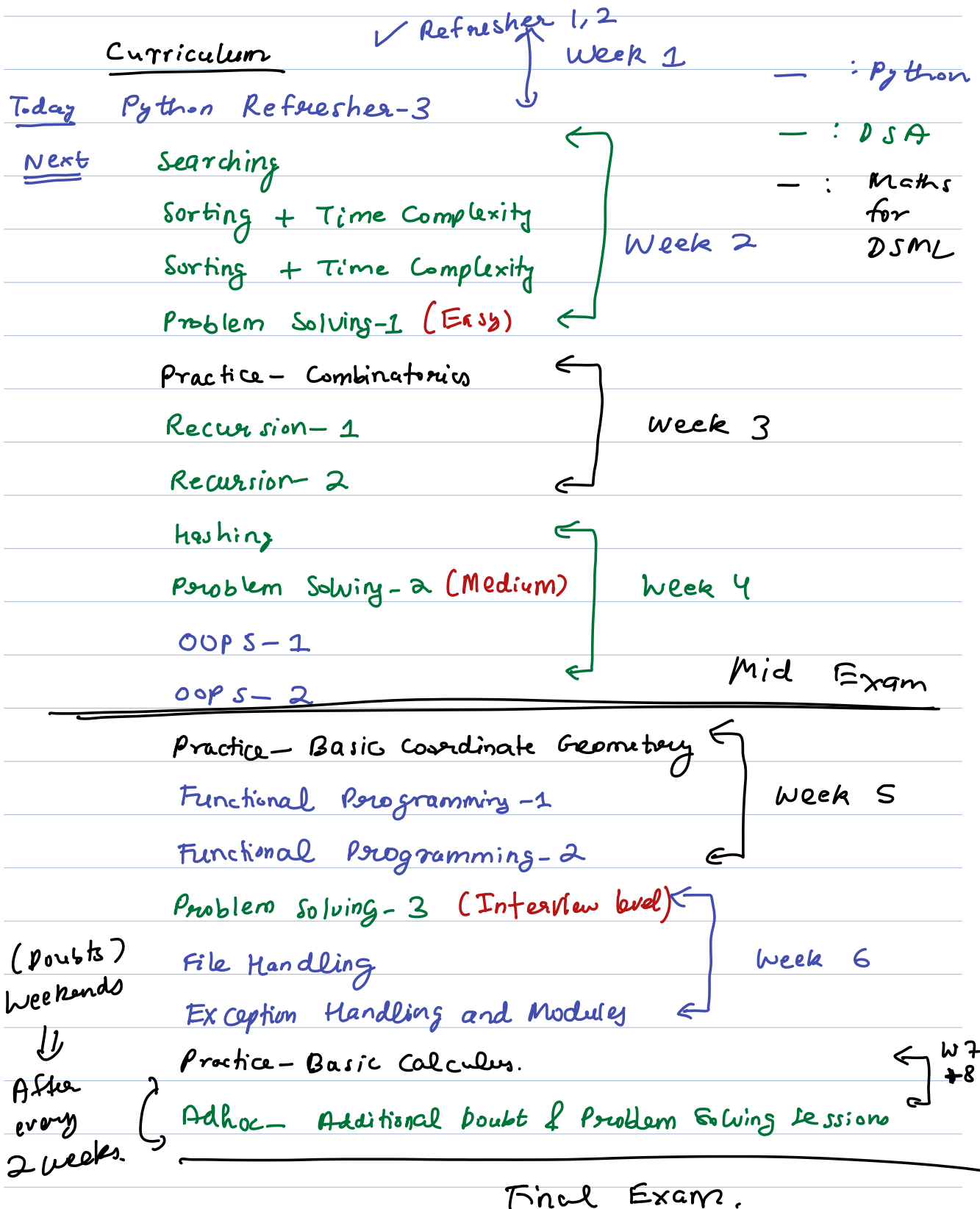


DSML Intermediate-3

(8 weeks)

Curriculum



① Iteration Protocol & For Loop

process

```
l = [1, 2, 3, 4, 5]
for i in l:
    print(i)
```

Iterable [List, sets, tuples, dict, range??, strings]

Block Body of the loop.

Iterator (points to `i`)

When to use while and when for?

↑
exit condition

↑
Iterable

② While Loop

x = 0

while (x < 10) :

x += 2

print(x)

exit. Condition / Loop condition.

Type conversions

bool(0) \Rightarrow False All other int \Rightarrow True

bool(0.0) \Rightarrow False All other float \Rightarrow True

bool('') \Rightarrow False All other str \Rightarrow True

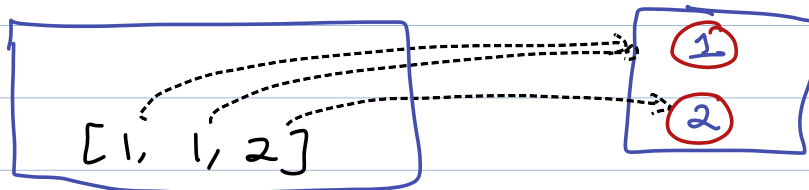
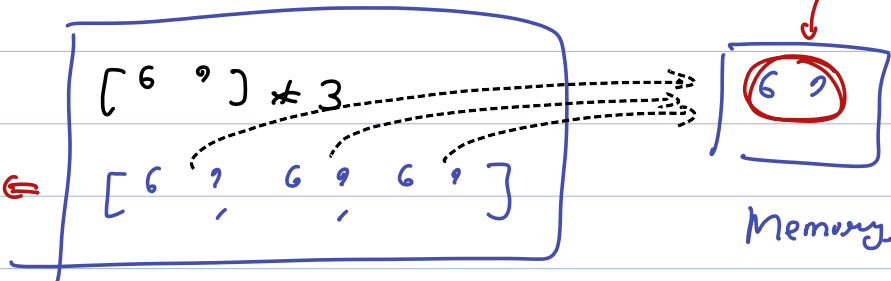
bool(False) \Rightarrow False True bool \Rightarrow True

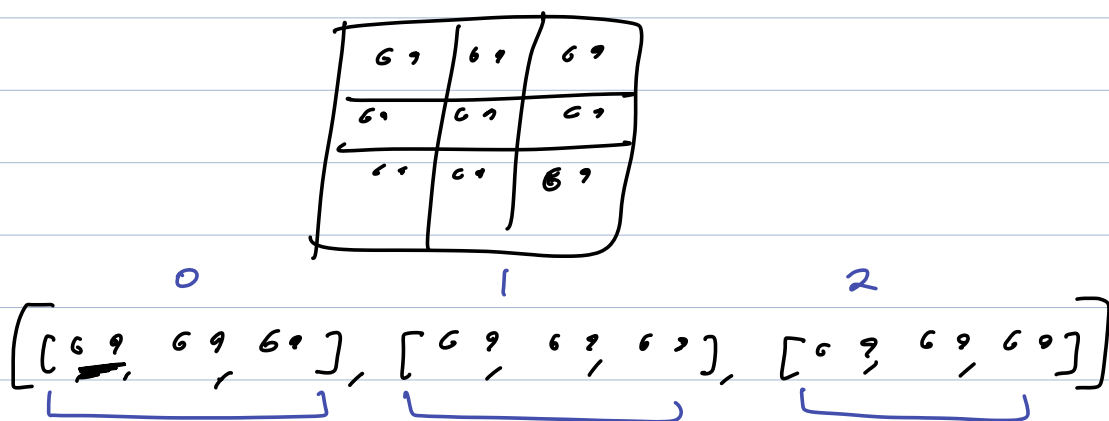
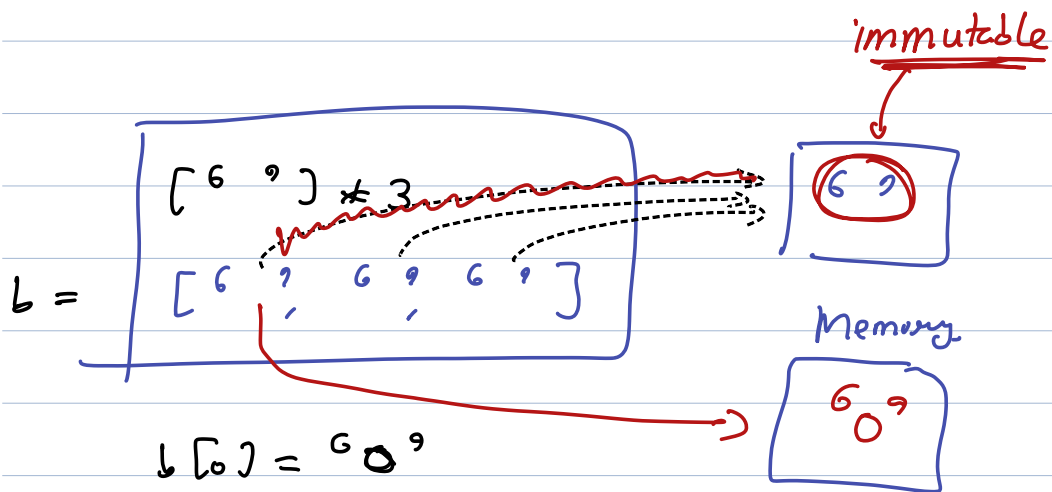
Board HW

immutable

Optimization

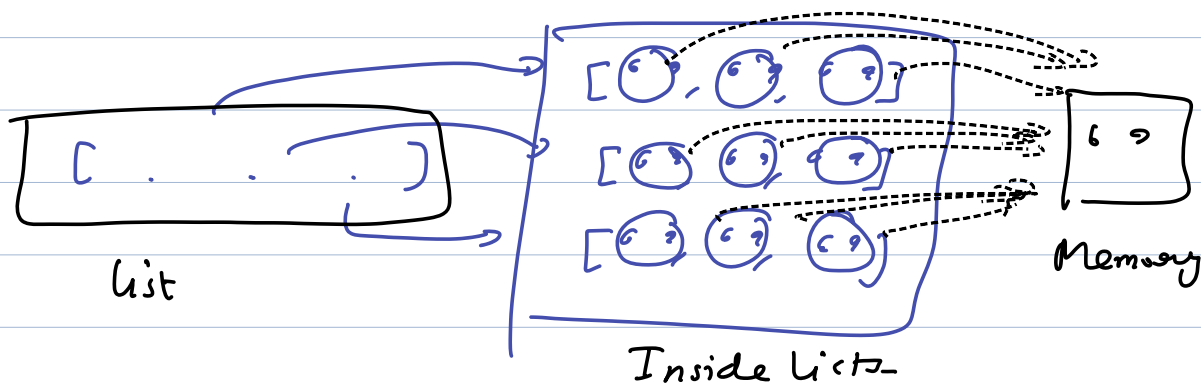
Same object.



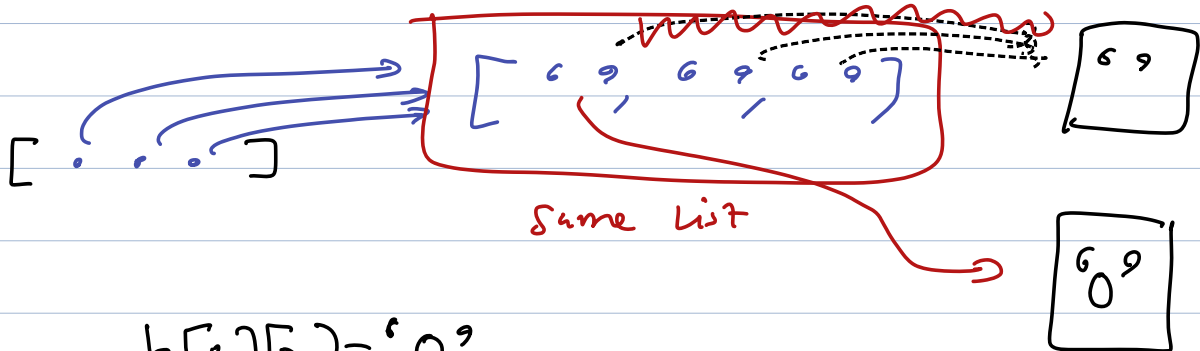


lists => mutable

3 diff lists.



$b = [['?', '?', '?']] * 3$



$b[0][0] = '0'$

Diff Brackets.

$l = list()$

$l = []$



List

$t = tuple()$

$t = ()$



Tuple

$d = dict()$

$d = \{ \}$



Dict

$s = set()$

$s = \{ 1, 3 \}$



Set

elements
with
a comma

$t = (1,)$

$t = 1,$

$t = 1, 2, 3$

Tuples

$t = \underline{\underline{(123)}}$



Int

Tuples unpacking

$a, b = [1, 2]$

$a, b, c = (1, 2, 3)$

$x = 1, 2, 3$



pack into a tuple

High performance.

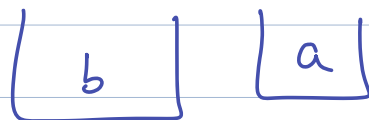
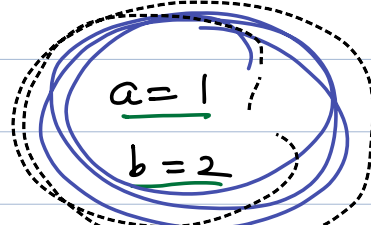
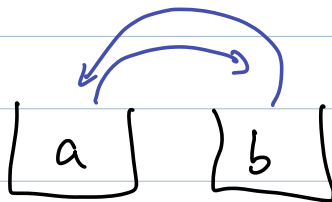
def func():

return 10, 20, 30



tuple.

Swap



$a = 2$

$b = 1$

Code

1

$a, b = b, a$

↳

1. $a, b = \underline{(2, 1)}$

Tuple packing

$\boxed{a = 1} \quad \boxed{b = 2}$

2. $a, b = (2, 1)$

Tuple unpacking

$\boxed{a = 2} \quad \boxed{b = 1}$