Day 39 10/04/2025

→ key : value (pairs)

JSON → JavaScript Object Notation.

→ (Newtonsoft Json) → Framework

↓

lightweight data interchange formate

↓

easy to read and write ~~bo~~ both Human & machines

used for (widely).

transmitting data between a server and Client in web appliatios

Json → supported in multiple langages
eg : C# , Java , Python , JavaScript .

Big-O notation → program speed

* Big-O natation tell how fast an operation is Done . .

1) O(1) -> constant Time.

2) O(n) -> Linear Time.

3) O(log n) -> Logarithmic Time.

-> no growth in operations
eg:- (get value using key) -> get stright answer

number of operations is proportinal to the growth of items. (eg: check all the item to get the answer)

-> growth of operations is smaller than the number of items.
(eg: it check item by diviedng the item by half to get the answer)

=> -> Lambda expression

```
List <int> nums = new List <int>() {1,2,3,4};
var sum = nums.where(item => item%2==0)
(parameter => expression)      variable      Lambda expression
List <int> nums = sum.ToList();
```

var data = sum . Fisrist() ;

↓ return the First data.

{1, 2, 3, 4)

→ (Return 1)

→ if no data meanes error.

var data = sum . First or default();

↓ return the First data.

↓ if no data meanes return default value.

orderby

var emp = Listname . orderby (item => item)

← axending

↓ orderby dexending

replace.ber => orderby dexending