

Algorithms and Data Structure with Dart

الخوارزميات و هياكل البيانات بلغة دارت

مفهوم الـ **Big O** الجزء الثانى



مهندس/ عاصم سحفان

مفهوم الـ **Big O** الجزء الثاني

اكتب Function بحسب المجموع من أول 1 إلى
الرقم الذي انتادخله #

$$5 \\ 1 + 2 + \underline{3 + 4 + 5}$$

مفهوم الـ Big O الجزء الثاني

الطريقة الأولى

```
double addUpTo(n) {  
    double total = 0;  
    for (int i = 1; i <= n; i++) {  
        total += i;  
    }  
    return total;  
}
```



الطريقة الثانية

```
double addUpToo(n) {  
    return n * (n + 1) / 2;  
}
```

$$\frac{50 \times (51)}{2} = \underline{\underline{1275}}$$

مفهوم الـ Big O الجزء الثاني

```
double addUpToo(n) {  
    return n * (n + 1) / 2;  
}
```

$$\begin{aligned} \text{addUpToo}(n) &= 1 + 2 + 3 + \dots + (n-1) + n \\ \text{addUpToo}(n) &= n + (n-1) + (n-2) + \dots + 3 + 2 + 1 \end{aligned}$$

$$\begin{aligned} 2 \text{addUpToo}(n) &= (n+1) + (n+1) + \dots + (n+1) + (n+1) \\ \text{addUpToo}(n) &= \frac{n * (n+1)}{2} \end{aligned}$$

مفهوم الـ **Big O** الجزء الثاني

شلال حسب الوقت

١) على كل جهاز حسابات مختلفة
٢) على الجهاز الواحد يتم تنفيذ

مفهوم الـ **Big O** الجزء الثانى

الوقت من العمل ليس مهم

Penche

مفهوم الـ Big O الجزء الثاني

```
double addUpTo(n) {  
    double total = 0;  
    for (int i = 1; i <= n; i++) {  
        total += i;  
    }  
    return total;  
}
```

assignment
head
n comparisons

الـ 5n
as —

$5n + 2$

n

```
double addUpToo(n) {  
    return n * (n + 1) / 2;  
}
```

٣ عمليات بسيطة
(عدد ثابت)

مفهوم الـ Big O الجزء الثانى

```
double addUpToo(n) {  
    return n * (n + 1) / 2;  
}
```

عدد العمليات
عابته

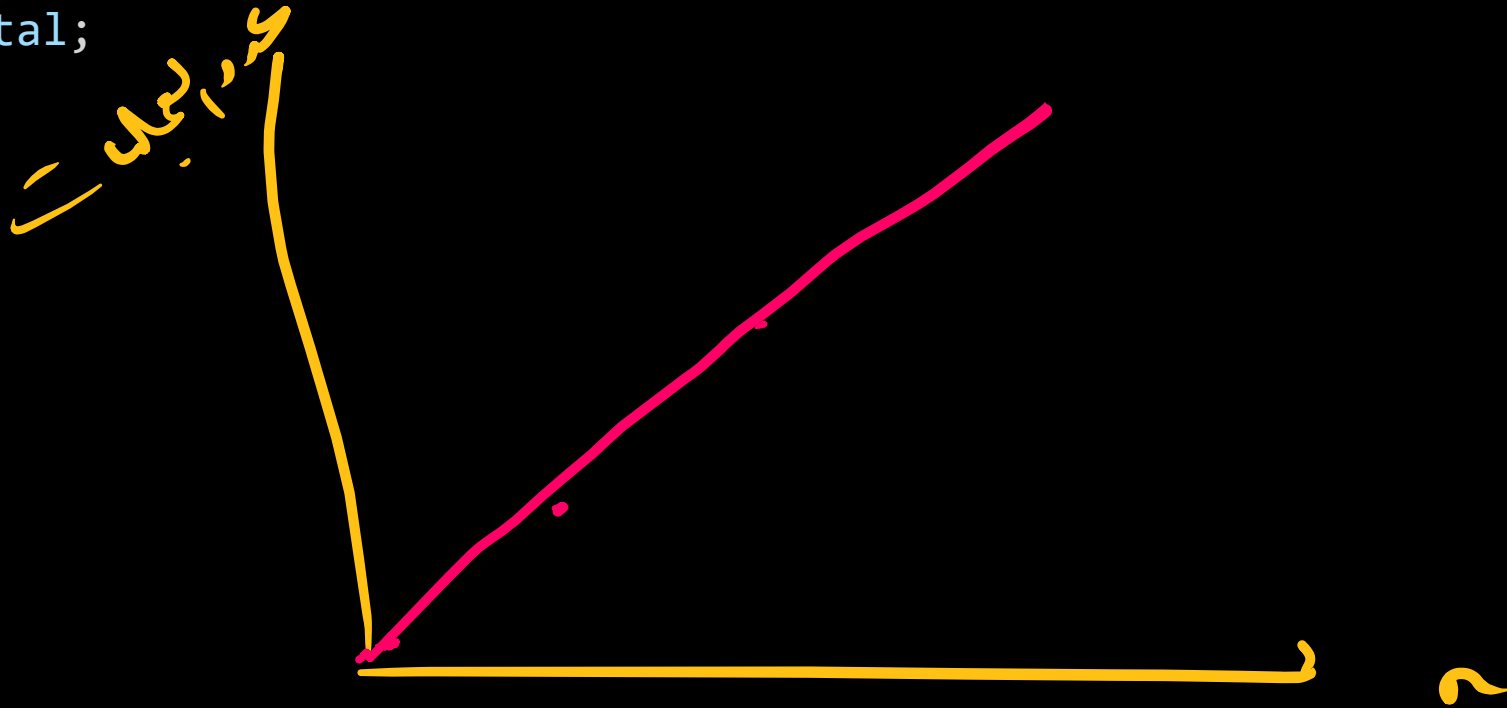
عدد العمليات

②

مفهوم الـ Big O الجزء الثاني

```
double addUpTo(n) {  
    double total = 0;  
    for (int i = 1; i <= n; i++) {  
        total += i;  
    }  
    return total;  
}
```

$5n+2 \rightarrow \Theta$



مفهوم الـ Big O الجزء الثاني

~~Big O~~

كيف نحدد بربط زيادة الـ runtime

input

زيادة الـ input

$O(f(n))$

$f(n) = 1 \rightarrow f(n) = \text{Constant}$
 $f(n) = n \rightarrow f(n) = \text{Linear} \checkmark$
 $f(n) = n^2 \rightarrow f(n) = \text{Quadratically} \checkmark$

4

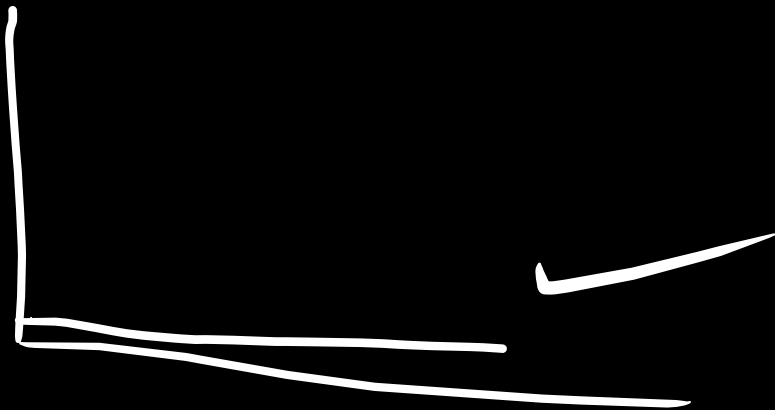
مفهوم الـ **Big O** الجزء الثانى

```
double addUpToo(n) {  
    return n * (n + 1) / 2;  
}
```

3

$f(n) = 1$

$O(1)$



مفهوم الـ Big O الجزء الثاني

```
double addUpTo(n) {  
    double total = 0;  
    for (int i = 1; i <= n; i++) {  
        total += i;  
    }  
    return total;  
}
```

$\rightarrow 5n + 2$ $f(n) = n$

$O(n)$



مفهوم الـ Big O الجزء الثاني

```
void countUpAndDown(n) {  
    print("Going up!");  
    for (int i = 0; i < n; i++) {  
        print(i);  
    }  
    print("At the top!");  
    for (int j = n - 1; j >= 0; j--  
) {  
        print(j);  
    }  
    print("Back down.");  
}
```

$O(n)$

$O(n)$

$O(n)$

مفهوم الـ Big O الجزء الثاني

```
void printAllPairs(n) {  
    for (var i = 0; i < n; i++) {  
        for (var j = 0; j < n; j++) {  
            print("$i, $j");  
        }  
    }  
}
```

عملية الطباعة

$O(n)$

$O(n)$ $O(n)$ $O(n)$

$O(n^2)$

0, 0
0, 1
0, 2
0, 3
0, 4
0, 5
1, 0
1, 1
1, 2
1, 3
1, 4
1, 5
2, 0
2, 1
2, 2
2, 3
2, 4
2, 5
3, 0
3, 1
3, 2
3, 3
3, 4
3, 5
4, 0
4, 1
4, 2
4, 3
4, 4
4, 5
5, 0
5, 1
5, 2
5, 3
5, 4
5, 5

مفهوم الـ Big O الجزء الثاني

