

WeHelp

Assignment - Week 2

運用任何你學過的 Python、JavaScript 程式語言基本知識，不依賴任何第三方的模組或套件的情況下，完成以下程式要求。

注意：請使用 Python 3 以上的版本進行開發。

要求一：函式與流程控制

完成以下函式，在函式中使用迴圈計算最小值到最大值之間，固定間隔的整數總和。其中你可以假設 max 一定大於 min 且為整數，step 為正整數。

提醒：請勿更動題目中任何已經寫好的程式。

Python

```
def calculate(min, max, step):  
    # 請用你的程式補完這個函式的區塊  
calculate(1, 3, 1) # 你的程式要能夠計算 1+2+3，最後印出 6  
calculate(4, 8, 2) # 你的程式要能夠計算 4+6+8，最後印出 18  
calculate(-1, 2, 2) # 你的程式要能夠計算 -1+1，最後印出 0
```

JavaScript

```
function calculate(min, max, step){  
    // 請用你的程式補完這個函式的區塊  
}  
calculate(1, 3, 1); // 你的程式要能夠計算 1+2+3，最後印出 6  
calculate(4, 8, 2); // 你的程式要能夠計算 4+6+8，最後印出 18  
calculate(-1, 2, 2); // 你的程式要能夠計算 -1+1，最後印出 0
```

WeHelp
Assignment - Week 2

要求二: Python 字典與列表、JavaScript 物件與陣列

完成以下函式, 正確計算出非 manager 的員工平均薪資, 所謂非 manager 就是在資料中 manager 欄位標註為 False (Python) 或 false (JavaScript) 的員工, 程式需考慮員工資料數量不定的情況。

提醒: 請勿更動題目任何已經寫好的程式。

Python

```
def avg(data):
    # 請用你的程式補完這個函式的區塊
    avg({
        "employees":[
            {
                "name":"John",
                "salary":30000,
                "manager":False
            },
            {
                "name":"Bob",
                "salary":60000,
                "manager":True
            },
            {
                "name":"Jenny",
                "salary":50000,
                "manager":False
            },
            {
                "name":"Tony",
                "salary":40000,
                "manager":False
            }
        ]
    }) # 呼叫 avg 函式
```

WeHelp
Assignment - Week 2

JavaScript

```
function avg(data){  
    // 請用你的程式補完這個函式的區塊  
}  
avg({  
    "employees":[  
        {  
            "name":"John",  
            "salary":30000,  
            "manager":false  
        },  
        {  
            "name":"Bob",  
            "salary":60000,  
            "manager":true  
        },  
        {  
            "name":"Jenny",  
            "salary":50000,  
            "manager":false  
        },  
        {  
            "name":"Tony",  
            "salary":40000,  
            "manager":false  
        }  
    ]  
}); // 呼叫 avg 函式
```

WeHelp
Assignment - Week 2

要求三：

完成以下函式，最後能印出程式中註解所描述的結果。

提醒：請勿更動題目中任何已經寫好的程式。

Python

```
def func(a):  
    # 請用你的程式補完這個函式的區塊  
func(2)(3, 4) # 你補完的函式能印出 2+(3*4) 的結果 14  
func(5)(1, -5) # 你補完的函式能印出 5+(1*-5) 的結果 0  
func(-3)(2, 9) # 你補完的函式能印出 -3+(2*9) 的結果 15  
# 一般形式為 func(a)(b, c) 要印出 a+(b*c) 的結果
```

JavaScript

```
function func(a){  
    // 請用你的程式補完這個函式的區塊  
}  
func(2)(3, 4); // 你補完的函式能印出 2+(3*4) 的結果 14  
func(5)(1, -5); // 你補完的函式能印出 5+(1*-5) 的結果 0  
func(-3)(2, 9); // 你補完的函式能印出 -3+(2*9) 的結果 15  
// 一般形式為 func(a)(b, c) 要印出 a+(b*c) 的結果
```

WeHelp
Assignment - Week 2

要求四：

找出至少包含兩筆整數的列表 (Python) 或陣列 (JavaScript) 中，兩兩數字相乘後的最大值。

提醒：請勿更動題目中任何已經寫好的程式，不可以使用排序相關的內建函式。

Python

```
def maxProduct(nums):  
    # 請用你的程式補完這個函式的區塊  
maxProduct([5, 20, 2, 6]) # 得到 120  
maxProduct([10, -20, 0, 3]) # 得到 30  
maxProduct([10, -20, 0, -3]) # 得到 60  
maxProduct([-1, 2]) # 得到 -2  
maxProduct([-1, 0, 2]) # 得到 0  
maxProduct([5,-1, -2, 0]) # 得到 2  
maxProduct([-5, -2]) # 得到 10
```

JavaScript

```
function maxProduct(nums){  
    // 請用你的程式補完這個函式的區塊  
}  
maxProduct([5, 20, 2, 6]) // 得到 120  
maxProduct([10, -20, 0, 3]) // 得到 30  
maxProduct([10, -20, 0, -3]) // 得到 60  
maxProduct([-1, 2]) // 得到 -2  
maxProduct([-1, 0, 2]) // 得到 0 或 -0  
maxProduct([5, -1, -2, 0]) // 得到 2  
maxProduct([-5, -2]) // 得到 10
```

WeHelp

Assignment - Week 2

要求五：

Given an array of integers, show indices of the two numbers such that they add up to a specific target. You can assume that each input would have exactly one solution, and you can not use the same element twice.

Python

```
def twoSum(nums, target):  
    # your code here  
result=twoSum([2, 11, 7, 15], 9)  
print(result) # show [0, 2] because nums[0]+nums[2] is 9
```

JavaScript

```
function twoSum(nums, target){  
    // your code here  
}  
let result=twoSum([2, 11, 7, 15], 9);  
console.log(result); // show [0, 2] because nums[0]+nums[2] is 9
```

WeHelp
Assignment - Week 2

要求六 (Optional):

給定只會包含 0 或 1 兩種數字的列表 (Python) 或陣列 (JavaScript), 計算連續出現 0 的最大長度。

提醒: 請勿更動題目有任何已經寫好的程式。

Python

```
def maxZeros(nums):  
    # 請用你的程式補完這個函式的區塊  
maxZeros([0, 1, 0, 0]) # 得到 2  
maxZeros([1, 0, 0, 0, 0, 1, 0, 1, 0, 0]) # 得到 4  
maxZeros([1, 1, 1, 1, 1]) # 得到 0  
maxZeros([0, 0, 0, 1, 1]) # 得到 3
```

JavaScript

```
function maxZeros(nums){  
    // 請用你的程式補完這個函式的區塊  
}  
maxZeros([0, 1, 0, 0]); // 得到 2  
maxZeros([1, 0, 0, 0, 0, 1, 0, 1, 0, 0]); // 得到 4  
maxZeros([1, 1, 1, 1, 1]); // 得到 0  
maxZeros([0, 0, 0, 1, 1]) // 得到 3
```