

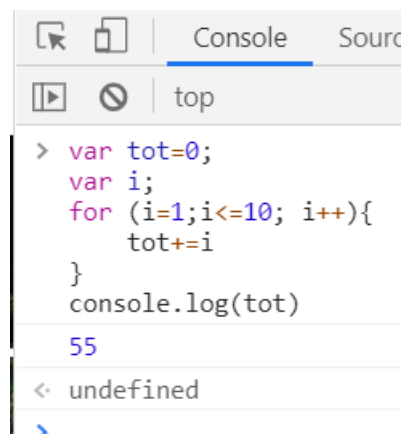
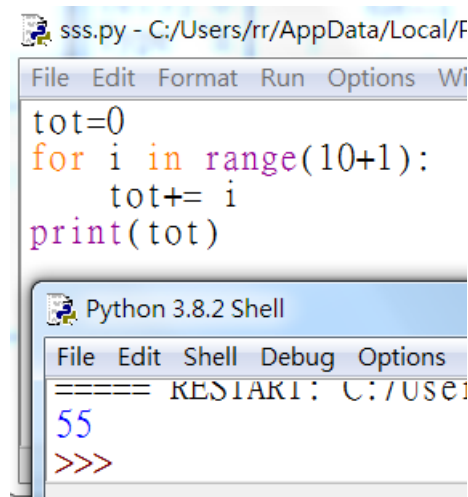
- Ex2-5Add1:一個簡單問題同時寫出 py 與 js 的程式：重覆結構的實驗，計算 1+2+...+10 的總和的實驗(for, while，與無窮迴圈)，實驗時，請把 10 改為「自己的座號加 100」。
  - Ex2-5Add2:一個簡單問題同時寫出 py 與 js 的程式：先自訂小型數據假設有兩個學生的名字與一系列成績，請計算平均並印出。
  - Ex2-5Add3:一個簡單問題同時寫出 py 與 js 的程式：宣告一個容器，假定原來的數據是 1,2,3,4，後來加入字串'foo'到容器又拿出來。然後印出這個容器的全部內容。
- 

Ex2-5Add1:一個簡單問題同時寫出 py 與 js 的程式：重覆結構的實驗，計算 1+2+...+10 的總和的實驗(for, while，與無窮迴圈)，實驗時，請把 10 改為自己的座號。

Ans:

以使用 for 為例：參考下列程式與附圖。

```
tot=0
for i in range(10+1):
    tot+= i
print(tot)
```



```
var tot=0;
var i;
for (i=1;i<=10; i++){
    tot+=i
}
console.log(tot)
```

Ex2-5Add2:一個簡單問題同時寫出 py 與 js 的程式：先自訂小型數據假設有兩個學生的名字與一系列成績，請計算平均並印出。

Ans:

參考附圖。

```
student_data=[
  {'name':'Bob',
   'scores':[90,80,70,60]},
  {'name':'Alice',
   'scores':[45,75,45,96]},
]
```

```
s_data=student_data
print("before:",s_data)
for data in s_data:
    av=sum(data['scores'])/ float(len(data['scores']))
    data['average']=av
print("after:",s_data)
```

```
JavaScript
studentData=[
  {'name':'Bob',
   'scores':[90,80,70,60]},
  {'name':'Alice',
   'scores':[45,75,45,96]},
]

console.log("before-----")
studentData.forEach(function(sdata){
  console.log(sdata.name+ "--scored-->" + sdata.average);
})

studentData.forEach(function(sdata){
  var av=sdata.scores
    .reduce(function(prev,current){
      return prev+current;
    },0) / sdata.scores.length;

  sdata.average=av;
})

console.log("after-----")
studentData.forEach(function(sdata){
  console.log(sdata.name+ "--scored-->" + sdata.av);
})
```

```
Console
"before-----"
"Bob--scored-->undefined"
"Alice--scored-->undefined"
"after-----"
"Bob--scored-->75"
"Alice--scored-->65.25"
```

```
student_data=[
  {'name':'Bob',
   'scores':[90,80,70,60]},
  {'name':'Alice',
   'scores':[45,75,45,96]},
]

s_data=student_data
print("before:",s_data)
for data in s_data:
    av=sum(data['scores'])/ float(len(data['scores']))
    data['average']=av
print("after:",s_data)
```

```
Python 3.8.2 Shell
before: [{'name': 'Bob', 'scores': [90, 80, 70, 60]}, {'name': 'Alice', 'scores': [45, 75, 45, 96]}]
after: [{'name': 'Bob', 'scores': [90, 80, 70, 60], 'average': 75.0}, {'name': 'Alice', 'scores': [45, 75, 45, 96], 'average': 65.25}]
>>>
```

```
studentData=[
  {'name':'Bob',
   'scores':[90,80,70,60]},
  {'name':'Alice',
   'scores':[45,75,45,96]},
]
```

```
console.log("before-----")
studentData.forEach(function(sdata){
```

```

console.log(sdata.name+ "--scored-->" + sdata.average);
})

studentData.forEach(function(sdata){
    var av=sdata.scores
        .reduce(function(prev,current){
            return prev+current;
        },0) / sdata.scores.length;

    sdata.average=av;
})

console.log("after-----")
studentData.forEach(function(sdata){
    console.log(sdata.name+ "--scored-->" + sdata.average);
})

```

Ex2-5Add3:一個簡單問題同時寫出 py 與 js 的程式：宣告一個容器，假定原來的數據是 1,2,3,4，後來加入字串'foo'到容器又拿出來。然後印出這個容器的全部內容。

Ans:

參考附圖。

```

l=[1,2,3,4]      ; print(l)
l.append('foo'); print(l)
l.pop()          ; print(l)
print(l[1:3])
print(l[-3:-1])
print(l[0:4:2])

z=[o*o for o in l]
print('z=>', z)

d={'a': 1, 'b':2,'c':3}
print('d=>', d)

print(d['a'])
print(d.get('a'))

print(d.get('z')) #None
#print(d['z'])    #KeyError: 'z'

for key,value in d.items():
    print(key,value)
for key in d:
    print(key)

for value in d.values():
    print(value)

```

## Lab#2-5-3a\_python的容器list放入移走印出的實驗

■ 按一下

```

l=[1,2,3,4] ; print(l)
l.append('foo'); print(l)
l.pop() ; print(l)
print(l[1:3])
print(l[-3:-1])
print(l[0:4:2])

z=[o*o for o in l]
print('z=>', z)

d={'a': 1, 'b':2, 'c':3}
print('d=>', d)

print(d['a'])
print(d.get('a'))

print(d.get('z')) #None
#print(d['z']) #KeyError:

for key,value in d.items():
    print(key,value)
for key in d:
    print(key)

for value in d.values():
    print(value)
  
```

## Lab#2-5-3b\_JS的容器list放入移走印出的實驗

方法一:用jsfiddle.net/測試

```

JavaScript + Processing.js 1.4.7
1 var l=[1,2,3,4] ; console.log(l);
2 l.push('foo') ; console.log(l);
3 l.pop() ; console.log(l);
4 console.log(l.slice(1,3) )
5 console.log(l.slice(-3,-1) )
6 console.log(l.slice(0,4,2) )
7
8 z=l.map(function(o){ return o*o;})
9 console.log('z=>', z)
10
11 d={'a': 2, 'b':5, 'c':9}
12 console.log('d=>', d)
13
14 console.log(d['a'])
15 console.log(d['a'] == d.a) //兩種寫法 true
16 console.log(d.z) //undefined
17 console.log(d['z']) //undefined
18
19 //較舊的寫法：會取出key的index字串 的寫法
20 console.log("print dictionary...")
21 for ( var key in d ) {
22     if (d.hasOwnProperty(key)){
23         var item=d[key];
24         console.log(key, ":", item);
25     }
26 }
27
28 //較新的寫法
29 console.log("obj=>", Object.keys(d));
30 console.log("values =>", Object.values(d));
31
32 Object.keys(d).forEach(function (item,i) {
33     console.log(i,":")
34     console.log(item); // key
35     console.log(d[item]); // value
36 });
  
```

方法二:用JSbin.com測試

```

JavaScript
var l=[1,2,3,4] ; console.log(l);
l.push('foo') ; console.log(l);
l.pop() ; console.log(l);
console.log(l.slice(1,3) )
console.log(l.slice(-3,-1) )
console.log(l.slice(0,4,2) )

z=l.map(function(o){ return o*o;})
console.log('z=>', z)

d={'a': 2, 'b':5, 'c':9}
console.log('d=>', d)

console.log(d['a'])
console.log(d['a'] == d.a) //兩種寫法 true
console.log(d.z) //undefined
console.log(d['z']) //undefined

//較舊的寫法：會取出key的index字串 的寫法
console.log("print dictionary...")
for ( var key in d ) {
    if (d.hasOwnProperty(key)){
        var item=d[key];
        console.log(key, ":", item);
    }
}

//較新的寫法
console.log("obj=>", Object.keys(d));
console.log("values =>", Object.values(d));

Object.keys(d).forEach(function (item,i) {
    console.log(i,":")
    console.log(item); // key
    console.log(d[item]); // value
});
  
```

```

var l=[1,2,3,4] ; console.log(l);
l.push('foo') ; console.log(l);
l.pop() ; console.log(l);
console.log(l.slice(1,3) )
console.log(l.slice(-3,-1) )
console.log(l.slice(0,4,2) )
  
```

```

z=l.map(function(o){ return o*o;})
console.log('z=>', z)
  
```

```

d={'a': 2, 'b':5, 'c':9}
console.log('d=>', d)
  
```

structor: Jiun-Ting Jiang

```
console.log(d['a'])
console.log(d['a'] === d.a) //兩種寫法 true
console.log(d.z )    //undefined
console.log(d['z'] ) //undefined
```

//較舊的寫法 ，會取出 key 的 index 字串 的寫法

```
console.log("print dictionary...")
for ( var key in d ) {
  if (d.hasOwnProperty(key)){
    var item=d[key];
    console.log(key, ":", item);
  }
}
```

//較新的寫法

```
console.log("obj=>", Object.keys(d));
console.log("values =>", Object.values(d));
```

```
Object.keys(d).forEach(function (item,i) {
  console.log(i,":")
  console.log(item); // key
  console.log(d[item]); // value
});
```

---