

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

## 步驟解說

執行檔案，產生 main()  
放入 Call Stack

main()

Event Loop



Call Stack

Web APIs

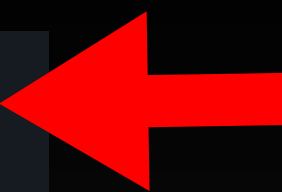


Terminal



Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```



## 步驟解說

進入迴圈產生

```
Global EC: {  
  VO: {  
    l: undefined  
  }  
}
```

```
Global EC: {  
  VO: {  
    l: undefined  
  }  
}
```

main()

Event Loop



Call Stack

Web APIs



Terminal



Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

## 步驟解說

設定 i = 0

Global EC: {

VO: {

  i: 0

}

}

main()

Event Loop



Call Stack

Web APIs

Terminal

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

步驟解說  
執行 line 2，放入 Call  
Stack

```
Global EC: {  
  VO: {  
    i: 0  
  }  
}
```

console.log('i:' + i)  
main()

Event Loop 

Call Stack

Web APIs

Terminal

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

## 步驟解說

Console 執行完畢，在  
Call Stack pop，在  
Terminal 印出

Global EC: {  
 VO: {  
 i: 0  
 }  
}

i: 0

Event Loop



main()

Call Stack

Web APIs

Terminal

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

```
Global EC: {  
  VO: {  
    i: 0  
  }  
}
```

i: 0

Terminal

## 步驟解說

執行 line 3 ，放入 Call Stack

setTimeout(cb, 0)  
main()

Call Stack

Web APIs

Event Loop 

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

```
Global EC: {  
  VO: {  
    l: 0  
  }  
}
```

i: 0

Terminal

## 步驟解說

發出要求，要瀏覽器設置 0 ms 的倒數計時器，時間到了，就呼叫第一個參數的 arrow function

main()

Call Stack

setTimeout(cb, 0)

Web APIs

Event Loop 

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

## 步驟解說

倒數完成，將第一個參數的 arrow function 放入 Task Queue

```
Global EC: {  
  VO: {  
    l: 0  
  }  
}
```

i: 0

Terminal

Event Loop



Call Stack

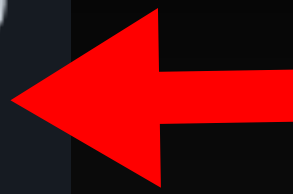
Web APIs

() => {console.log(i)}

Task Queue



```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```



步驟解說  
Event Loop 的機制是等到 Call Stack 空了之後，才會讓 Task Queue 的東西放入 Call Stack。  
因為 Call Stack 還有東西，所以 Task Queue 裡面的東西要繼續等待

```
Global EC: {  
  VO: {  
    i: 0  
  }  
}
```

Event Loop 

main()

Call Stack

Web APIs

i: 0

Terminal

() => {console.log(i)}

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

## 步驟解說

i++ 變成 1

Global EC: {

VO: {

i: 1

}

}

main()

Event Loop



Call Stack

Web APIs

i: 0

() => {console.log(i)}

Terminal

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

步驟解說  
console.log('l:' + i)  
放入 Call Stack

```
Global EC: {  
  VO: {  
    l: 1  
  }  
}
```

console.log('l:' + i)  
main()

Event Loop 

Call Stack

Web APIs

i : 0

Terminal

() => {console.log(i)}

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

## 步驟解說

Console 執行完畢，在  
Call Stack pop，在  
Terminal 印出

Global EC: {  
 VO: {  
 l: 1  
 }  
}

i: 0  
i: 1

Terminal

Event Loop



main()

Call Stack

Web APIs

() => {console.log(i)}

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

```
Global EC: {  
  VO: {  
    l: 1  
  }  
}
```

```
i: 0  
i: 1
```

Terminal

## 步驟解說

執行 line 3 ，放入 Call  
Stack

```
setTimeout(cb,  
1000)  
main()
```

Call Stack

Web APIs

Event Loop 

```
() => {console.log(i)}
```

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

步驟解說  
發出要求，要瀏覽器設置 1000 ms 的倒數計時器，時間到了，就呼叫第一個參數的 arrow function

```
Global EC: {  
  VO: {  
    l: 1  
  }  
}
```

main()

setTimeout(cb,  
1000)

Event Loop 

Call Stack

Web APIs

```
i: 0  
i: 1
```

Terminal

() => {console.log(i)}

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

## 步驟解說

i++ 變成 2

Global EC: {

VO: {

i: 2

}

}

main()

setTimeout(cb,  
1000)

Event Loop



Call Stack

Web APIs

i: 0  
i: 1

() => {console.log(i)}

Terminal

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

步驟解說  
console.log('l:' + i)  
放入 Call Stack

```
Global EC: {  
  VO: {  
    l: 2  
  }  
}
```

console.log('l:' + i)  
main()

setTimeout(cb,  
1000)

Event Loop 

Call Stack

Web APIs

```
i: 0  
i: 1
```

Terminal

() => {console.log(i)}

Task Queue



```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

## 步驟解說

Console 執行完畢，在  
Call Stack pop，在  
Terminal 印出

```
Global EC: {  
  VO: {  
    i: 2  
  }  
}
```

```
i: 0  
i: 1  
i: 2
```

Terminal

Event Loop



Call Stack

Web APIs

() => {console.log(i)}

Task Queue

main()

setTimeout(cb,  
1000)

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

```
Global EC: {  
  VO: {  
    l: 2  
  }  
}
```

```
i: 0  
i: 1  
i: 2
```

Terminal

## 步驟解說

執行 line 3 ，放入 Call Stack

setTimeout(cb,  
2000)  
main()

setTimeout(cb,  
1000)

Call Stack

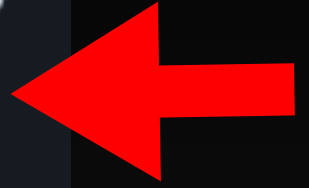
Web APIs

Event Loop 

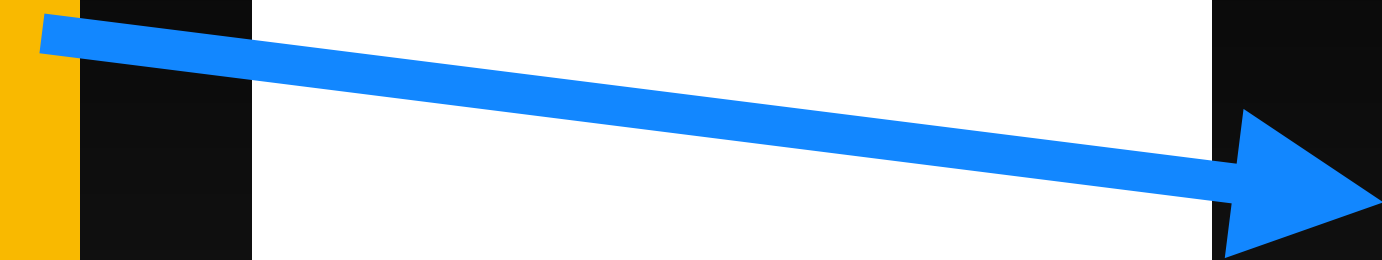
() => {console.log(i)}

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```




步驟解說  
發出要求，要瀏覽器設置 2000 ms 的倒數計時器，時間到了，就呼叫第一個參數的 arrow function



main()

setTimeout(cb, 2000)  
setTimeout(cb, 1000)

```
Global EC: {  
  VO: {  
    l: 2  
  }  
}
```

Event Loop 

Call Stack

Web APIs

```
i: 0  
i: 1  
i: 2
```

() => {console.log(i)}

Terminal

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

## 步驟解說

i++ 變成 3

Global EC: {

VO: {

i: 3

}

}

main()

setTimeout(cb,  
2000)

setTimeout(cb,  
1000)

Event Loop



Call Stack

Web APIs

i: 0

i: 1

i: 2

() => {console.log(i)}

Terminal

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

```
Global EC: {  
  VO: {  
    l: 3  
  }  
}
```

```
i: 0  
i: 1  
i: 2
```

Terminal

步驟解說  
console.log('l:' + i)  
放入 Call Stack

```
console.log('l:' + i)  
main()
```

Call Stack

```
setTimeout(cb,  
2000)  
setTimeout(cb,  
1000)
```

Web APIs

Event Loop 

```
() => {console.log(i)}
```

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

步驟解說  
Console 執行完畢，在  
Call Stack pop，在  
Terminal 印出

Global EC: {  
 VO: {  
 l: 3  
 }  
}

i: 0   i: 3  
i: 1  
i: 2

Terminal

Event Loop 

main()

Call Stack

setTimeout(cb,  
2000)  
setTimeout(cb,  
1000)

Web APIs

() => {console.log(i)}

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

```
Global EC: {  
  VO: {  
    l: 3  
  }  
}
```

```
i: 0   i: 3  
i: 1  
i: 2
```

Terminal

## 步驟解說

執行 line 3 ，放入 Call Stack

setTimeout(cb,  
3000)  
main()

Call Stack

setTimeout(cb,  
2000)  
setTimeout(cb,  
1000)

Web APIs

Event Loop 

() => {console.log(i)}

Task Queue



```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

步驟解說  
發出要求，要瀏覽器設置 3000 ms 的倒數計時器，時間到了，就呼叫第一個參數的 arrow function

```
Global EC: {  
  VO: {  
    l: 3  
  }  
}
```

main()

```
setTimeout(cb, 3000)  
setTimeout(cb, 2000)  
setTimeout(cb, 1000)
```

Event Loop 

Call Stack

Web APIs

```
i: 0   i: 3  
i: 1  
i: 2
```

Terminal

```
() => {console.log(i)}
```

Task Queue



```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

## 步驟解說

i++ 變成 4

```
Global EC: {  
  VO: {  
    i: 4  
  }  
}
```

main()

```
setTimeout(cb,  
3000)  
setTimeout(cb,  
2000)  
setTimeout(cb,  
1000)
```

Event Loop



Call Stack

Web APIs

```
i: 0   i: 3  
i: 1  
i: 2
```

Terminal

```
() => {console.log(i)}
```

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

```
Global EC: {  
  VO: {  
    l: 4  
  }  
}
```

```
i: 0   i: 3  
i: 1  
i: 2
```

Terminal

步驟解說  
console.log('l:' + i)  
放入 Call Stack

```
console.log('l:' + i)  
main()
```

Call Stack

```
setTimeout(cb,  
3000)  
setTimeout(cb,  
2000)  
setTimeout(cb,  
1000)
```

Web APIs

Event Loop 

```
() => {console.log(i)}
```

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

步驟解說  
Console 執行完畢，在  
Call Stack pop，在  
Terminal 印出

Global EC: {  
 VO: {  
 i: 4  
 }  
}

```
i: 0 i: 3  
i: 1 i: 4  
i: 2
```

Terminal

Event Loop



main()

Call Stack

```
setTimeout(cb,  
3000)  
setTimeout(cb,  
2000)  
setTimeout(cb,  
1000)
```

Web APIs

() => {console.log(i)}

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

```
Global EC: {  
  VO: {  
    l: 4  
  }  
}
```

```
i: 0   i: 3  
i: 1   i: 4  
i: 2
```

Terminal

## 步驟解說

執行 line 3 ，放入 Call Stack

setTimeout(cb,  
4000)  
main()

Call Stack

setTimeout(cb,  
3000)  
setTimeout(cb,  
2000)  
setTimeout(cb,  
1000)

Web APIs

Event Loop 

() => {console.log(i)}

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

步驟解說  
發出要求，要瀏覽器設置 4000 ms 的倒數計時器，時間到了，就呼叫第一個參數的 arrow function

```
Global EC: {  
  VO: {  
    l: 4  
  }  
}
```

main()

```
setTimeout(cb, 4000)  
setTimeout(cb, 3000)  
setTimeout(cb, 2000)  
setTimeout(cb, 1000)
```

Event Loop 

Call Stack

Web APIs

```
i: 0   i: 3  
i: 1   i: 4  
i: 2
```

```
() => {console.log(i)}
```

Terminal

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

## 步驟解說

i++ 變成 5  
跳脫迴圈

```
Global EC: {  
  VO: {  
    i: 5  
  }  
}
```

main()

```
setTimeout(cb, 4000)  
setTimeout(cb, 3000)  
setTimeout(cb, 2000)  
setTimeout(cb, 1000)
```

Event Loop 

Call Stack

Web APIs

```
i: 0   i: 3  
i: 1   i: 4  
i: 2
```

() => {console.log(i)}

Terminal

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

```
Global EC: {  
  VO: {  
    l:5  
  }  
}
```

## 步驟解說

main() 執行完畢，pop

Event Loop 

Call Stack

Web APIs

```
i:0   i:3  
i:1   i:4  
i:2
```

Terminal

() => {console.log(i)}

Task Queue

```
setTimeout(cb,  
4000)  
setTimeout(cb,  
3000)  
setTimeout(cb,  
2000)  
  
setTimeout(cb,  
1000)
```



```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

```
Global EC: {  
  VO: {  
    l:5  
  }  
}
```

```
i:0   i:3  
i:1   i:4  
i:2
```

Terminal

## 步驟解說

Call Stack 空了，因為  
Event Loop 是先進先  
出，所以放行第一個進  
來 Task Queue 的東西去  
Call Stack

Event Loop 

```
() => {  
  console.log(i)  
}
```

Call Stack

```
setTimeout(cb, 4000)  
setTimeout(cb, 3000)  
setTimeout(cb, 2000)  
setTimeout(cb, 1000)
```

Web APIs

Task Queue



```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

## 步驟解說

執行 arrow function  
去找到 i，並印出來，從  
Call Stack pop

Global EC: {  
 VO: {  
 i: 5  
 }  
}

i: 0 i: 3  
i: 1 i: 4  
i: 2 5

Event Loop



Call Stack

Web APIs

Terminal

Task Queue

setTimeout(cb,  
4000)  
setTimeout(cb,  
3000)  
setTimeout(cb,  
2000)  
setTimeout(cb,  
1000)

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

```
Global EC: {  
  VO: {  
    l:5  
  }  
}
```

## 步驟解說

以上都是幾乎一瞬間完成

```
setTimeout(cb,  
4000)  
setTimeout(cb,  
3000)  
setTimeout(cb,  
2000)  
setTimeout(cb,  
1000)
```

Event Loop 

Call Stack

Web APIs

```
i:0  i:3  
i:1  i:4  
i:2  5
```

Terminal

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

```
Global EC: {  
  VO: {  
    l:5  
  }  
}
```

## 步驟解說

一秒鐘過後，大家計時  
器一起少一秒

setTimeout(cb,  
3000)  
setTimeout(cb,  
2000)  
setTimeout(cb,  
1000)  
setTimeout(cb,  
0)

Event Loop 

Call Stack

Web APIs

```
i:0  i:3  
i:1  i:4  
i:2  5
```

Terminal

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

```
Global EC: {  
  VO: {  
    l:5  
  }  
}
```

## 步驟解說

有一個計時器倒數完成，將第一個參數的 arrow function 放入 Task Queue

Event Loop 

Call Stack

Web APIs


```
i:0  i:3  
i:1  i:4  
i:2  5
```

Terminal

() => {console.log(i)}

Task Queue

```
setTimeout(cb, 3000)  
setTimeout(cb, 2000)  
setTimeout(cb, 1000)
```



```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

```
Global EC: {  
  VO: {  
    l:5  
  }  
}
```

```
i:0  i:3  
i:1  i:4  
i:2  5
```

Terminal

## 步驟解說

Call Stack 空了，因為  
Event Loop 是先進先  
出，所以放行 Task  
Queue 的東西去 Call  
Stack

Event Loop 

```
() => {  
  console.log(i)  
}
```

Call Stack

```
setTimeout(cb,  
3000)  
setTimeout(cb,  
2000)  
setTimeout(cb,  
1000)
```

Web APIs

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

## 步驟解說

執行 arrow function  
去找到 i，並印出來，從  
Call Stack pop

Global EC: {

VO: {

i:5

}

}

```
i:0  i:3  5  
i:1  i:4  
i:2  5
```

Terminal

Event Loop



Call Stack

Web APIs

Task Queue

```
setTimeout(cb,  
3000)  
setTimeout(cb,  
2000)  
setTimeout(cb,  
1000)
```

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

```
Global EC: {  
  VO: {  
    l:5  
  }  
}
```

## 步驟解說

一秒鐘過後，大家計時  
器一起少一秒

setTimeout(cb,  
2000)  
setTimeout(cb,  
1000)  
setTimeout(cb,  
0)

Event Loop 

Call Stack

Web APIs

```
i:0   i:3   5  
i:1   i:4  
i:2   5
```

Terminal

Task Queue



```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

步驟解說

有一個計時器倒數完成，將第一個參數的 arrow function 放入 Task Queue

```
Global EC: {  
  VO: {  
    l:5  
  }  
}
```

Event Loop 

Call Stack

Web APIs

```
i:0   i:3   5  
i:1   i:4  
i:2   5
```

Terminal

  
( ) => {console.log(i)}

Task Queue



```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

```
Global EC: {  
  VO: {  
    l:5  
  }  
}
```

```
i:0   i:3   5  
i:1   i:4  
i:2   5
```

Terminal

## 步驟解說

Call Stack 空了，因為  
Event Loop 是先進先  
出，所以放行 Task  
Queue 的東西去 Call  
Stack

Event Loop 

```
() => {  
  console.log(i)  
}
```

Call Stack

```
setTimeout(cb,  
2000)  
setTimeout(cb,  
1000)
```

Web APIs

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

## 步驟解說

執行 arrow function  
去找到 i，並印出來，從  
Call Stack pop

Global EC: {

VO: {

i:5

}

}

```
i:0  i:3  5  
i:1  i:4  5  
i:2  5
```

Terminal

Event Loop



Call Stack

Web APIs

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

```
Global EC: {  
  VO: {  
    l:5  
  }  
}
```

## 步驟解說

一秒鐘過後，大家計時  
器一起少一秒

setTimeout(cb,  
1000)  
setTimeout(cb,  
0)

Event Loop 

Call Stack

Web APIs

```
i:0   i:3   5  
i:1   i:4   5  
i:2   5
```

Terminal

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

```
Global EC: {  
  VO: {  
    l:5  
  }  
}
```

## 步驟解說

有一個計時器倒數完成，將第一個參數的 arrow function 放入 Task Queue

Event Loop 

Call Stack

Web APIs

```
i:0   i:3   5  
i:1   i:4   5  
i:2   5
```

Terminal

  
( ) => {console.log(i)}

Task Queue

setTimeout(cb,  
1000)

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

```
Global EC: {  
  VO: {  
    l:5  
  }  
}
```

```
i:0   i:3   5  
i:1   i:4   5  
i:2   5
```

Terminal

## 步驟解說

Call Stack 空了，因為  
Event Loop 是先進先  
出，所以放行 Task  
Queue 的東西去 Call  
Stack

Event Loop 

```
() => {  
  console.log(i)  
}
```

Call Stack

```
setTimeout(cb,  
1000)
```

Web APIs

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

## 步驟解說

執行 arrow function  
去找到 i，並印出來，從  
Call Stack pop

Global EC: {

VO: {

i:5

}

}

i:0	i:3	5
i:1	i:4	5
i:2	5	5

Terminal

Event Loop



Call Stack

Web APIs

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

```
Global EC: {  
  VO: {  
    l:5  
  }  
}
```

## 步驟解說

一秒鐘過後，大家計時  
器一起少一秒

setTimeout(cb,  
0)

Event Loop 

Call Stack

Web APIs

```
i:0   i:3   5  
i:1   i:4   5  
i:2   5     5
```

Terminal

Task Queue



```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

```
Global EC: {  
  VO: {  
    l:5  
  }  
}
```

## 步驟解說

有一個計時器倒數完成，將第一個參數的 arrow function 放入 Task Queue

Event Loop 

Call Stack

Web APIs

```
i:0  i:3  5  
i:1  i:4  5  
i:2  5    5
```

Terminal

  
( ) => {console.log(i)}

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

```
Global EC: {  
  VO: {  
    l:5  
  }  
}
```

```
i:0  i:3  5  
i:1  i:4  5  
i:2  5    5
```

Terminal

## 步驟解說

Call Stack 空了，因為  
Event Loop 是先進先  
出，所以放行 Task  
Queue 的東西去 Call  
Stack

Event Loop 

```
() => {  
  console.log(i)  
}
```

Call Stack

Web APIs

Task Queue

```
for(var i=0; i<5; i++) {  
  console.log('i: ' + i)  
  setTimeout(() => {  
    console.log(i)  
  }, i * 1000)  
}
```

## 步驟解說

執行 arrow function  
去找到 i，並印出來，從  
Call Stack pop

Global EC: {

VO: {

i:5

}

}

i:0	i:3	5	5
i:1	i:4	5	
i:2	5	5	

Terminal

Event Loop



Call Stack

Web APIs

Task Queue