



SiNGULAR 奇點創意

程式創客教室

機器人 / AI人工智慧 / 程式語言

*Join Singular!
Be a super inventor!*

主題：Python MTA 3

主講人：Peter 日期：2021/06/02



Maker + Coder =



Singular Super Inventor



取得今天日期時間

```
import datetime
```

```
print(datetime.date.today())
```



存取日期的年、月、日

```
import datetime
```

執行結果:>>>

```
Date = datetime.date.today()
```

```
print(Date) #年月日
```

2021-05-30

```
print(Date.year) #年
```

2021

```
print(Date.month) #月
```

5

```
print(Date.day) #日
```

30



Strftime格式化日期

參數	說明	%Y	以 4 位數表示西元年份
%d	日期	%y	以 2 位數表示西元年份
%m	月份阿拉伯數字(1-12)	%A	星期正寫
%B	月份正寫	%a	星期縮寫
%b	月份縮寫		

strftime (format[])

```
import datetime
#datetime to string
Date = datetime.date.today()
print (Date.strftime('%d %b %B %Y %y %A %a'))
```

執行結果

```
>>> 30 May May 2021 21 Sunday Sun
```



SINGULAR
奇點創意
程式創客教室
機器人 / AI人工智慧 / 程式語言

strptime (string, format[])

#字串轉datetime

```
import datetime
```

```
day = input("What is your birthday? ")
```

```
print(day)
```

```
birth = datetime.datetime.strptime(day, "%m/%d/%Y")
```

```
print(birth.date())
```

計算距離下次生日的天數

```
import datetime

next = datetime.datetime.strptime('8/3/20
21', '%m/%d/%Y').date()
now = datetime.date.today()
diff = next - now
print(diff)
```


strftime格式化時間

參數	說明
%M	分鐘
%S	秒數
%H	24小時制(1-24)
%I	12小時制
%p	AM or PM

存取時、分、秒

```
import datetime
```

```
Time = datetime.datetime.now()
```

```
print(Time)
```

```
print(Time.hour)
```

```
print(Time.minute)
```

```
print(Time.second)
```



比較運算子

Maker + Coder = Singular Super Inventor



SiNGULAR
奇點創意

程式創客教室

機器人 / AI人工智慧 / 程式語言

== 運算子

比較輸入是否相等

- 輸入
 - 任意型態的兩個數值
- 輸出
 - True 如果兩者相等
 - False 如果兩者不相等

練習

- 1 == 1
- 1 == 1.0
- 1 == "1"

指令格式

數值1 == 數值2

```
>>> 1 == 1
True
>>> 1 == 1.0
True
>>> 1 == "1"
False
>>>
```



!= 運算子

比較輸入是否不相等

- 輸入
 - 任意型態的兩個數值
- 輸出
 - True 如果兩者不相等
 - False 如果兩者相等

練習

- `1 != 0`
- `1 != 1.0`
- `1 != "1"`
- `1.7 != 2.7 - 1.0`

指令格式

數值1 != 數值2

```
>>> 1 != 0
True
>>> 1 != 1.0
False
>>> 1 != "1"
True
>>> 1.7 != 2.7 - 1.0
True
>>>
```



< 運算子

比較輸入是否前者小於後者

- 輸入
 - 兩個數值
- 輸出
 - True 如果前者小於後者
 - False 如果前者不小於後者

練習

- 1 < 2
- 1 < 1
- 1 < 0
- 1 < "2"

指令格式

數值1 < 數值2

```
>>> 1 < 2
```

```
True
```

```
>>> 1 < 1
```

```
False
```

```
>>> 1 < 0
```

```
False
```

```
>>> 1 < "2"
```

```
Traceback (most recent call  
last):
```

```
File "<stdin>", line 1, in  
<module>
```

```
TypeError: '<' not supported  
between instances of 'int' and  
'str'
```



SINGULAR
奇點創意

程式創客教室

機器人 / AI人工智慧 / 程式語言

> 運算子

比較輸入是否前者大於後者

- 輸入
 - 兩個數值
- 輸出
 - True 如果前者大於後者
 - False 如果前者不大於後者

練習

- 1 > 2
- 1 > 1
- 1 > 0
- 1 > "2"

指令格式

數值1 > 數值2

```
>>> 1 > 2
True
>>> 1 > 1
False
>>> 1 > 0
False
>>> 1 > "2"
Traceback (most recent call
last):
  File "<stdin>", line 1, in
<module>
TypeError: '>' not supported
between instances of 'int' and
'str'
```



<= 運算子

比較輸入是否前者小於或等於後者

- 輸入
 - 兩個數值
- 輸出
 - 真 如果兩者相等
 - 假 如果兩者不相等

練習

- 1 <= 2
- 1 <= 1
- 1 <= 0
- 1 <= "2"

指令格式

數值1 <= 數值2

```
>>> 1 <= 2
True
>>> 1 <= 1
True
>>> 1 <= 0
False
>>> 1 <= "2"
Traceback (most recent call
last):
  File "<stdin>", line 1, in
<module>
TypeError: '<= >' not supported
between instances of 'int' and
'str'
```



SINGULAR
奇點創意

程式創客教室

機器人 / AI人工智慧 / 程式語言

>= 運算子

比較輸入是否前者大於或等於後者

- 輸入
 - 兩個數值
- 輸出
 - 真 如果兩者相等
 - 假 如果兩者不相等

練習

- `1 >= 2`
- `1 >= 1`
- `1 >= 0`
- `1 >= "2"`

指令格式

數值1 >= 數值2

```
>>> 1 >= 2
False
>>> 1 >= 1
True
>>> 1 >= 0
True
>>> 1 >= "2"
Traceback (most recent call
last):
  File "<stdin>", line 1, in
<module>
TypeError: '<=' not supported
between instances of 'int' and
'str'
```



SINGULAR
奇點創意

程式創客教室

機器人 / AI人工智慧 / 程式語言



邏輯運算子

Maker + Coder = Singular Super Inventor



SiNGULAR
奇點創意

程式創客教室

機器人 / AI人工智慧 / 程式語言

not 運算子

反向布林輸入

- 輸入
 - 一個布林數值
- 輸出
 - True 如果輸入為 False
 - False 如果輸入為 True

指令格式

```
not 布林數值
```

練習

- not False
- not True
- not 1 == 1
- not not False

```
>>> not False
True
>>> not True
False
>>> not 1 == 1
False
>>> not not False
False
```



SINGULAR
奇點創意

程式創客教室

機器人 / AI人工智慧 / 程式語言

and 運算子

檢查是否兩者皆為真

- 輸入
 - 兩個布林數值
- 輸出
 - True 如果兩者皆為真
 - False 反之

練習

- True and True
- True and False
- False and True
- False and False

指令格式

布林數值1 and 布林數值2

```
>>> True and True
True
>>> True and False
False
>>> False and True
False
>>> False and False
False
>>>
```



or 運算子

檢查是否兩者其中一個為真

指令格式

布林數值1 or 布林數值2

- 輸入
 - 兩個布林數值
- 輸出
 - True 如果兩者其中一個為真
 - False 反之

練習

- True or True
- True or False
- False or True
- False or False

```
>>> True or True
True
>>> True or False
True
>>> False or True
True
>>> False or False
False
>>>
```





單向判斷式

Maker + Coder = Singular Super Inventor



SiNGULAR
奇點創意

程式創客教室

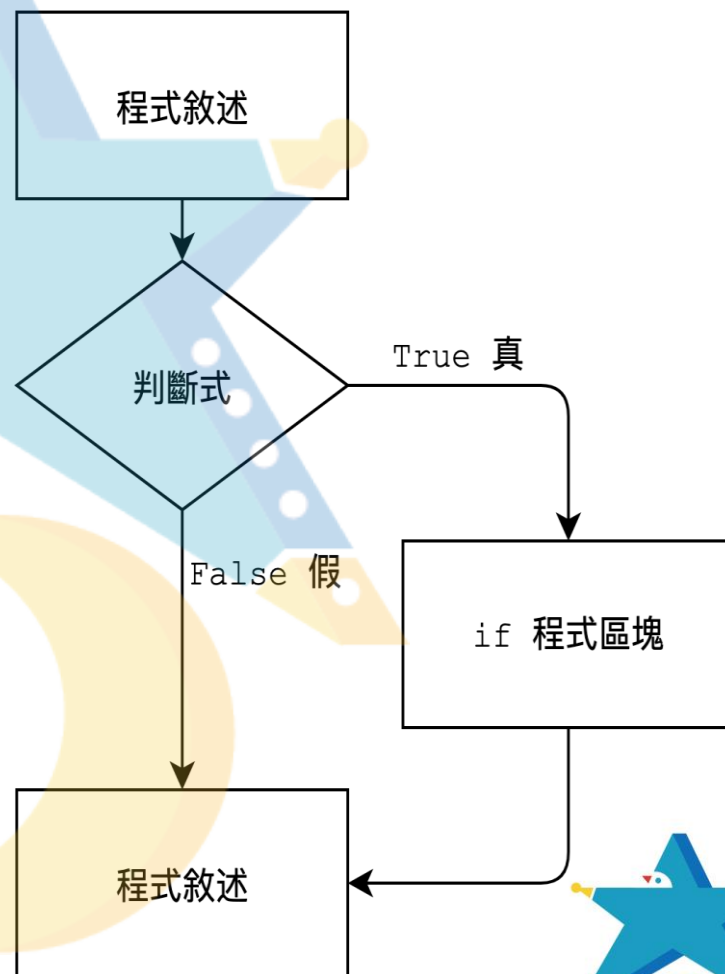
機器人 / AI人工智慧 / 程式語言

if 敘述

當判斷式為真，執行程式的指令
當判斷式為假，跳過不執行

指令格式

```
if 判斷式:  
    程式指令1  
    程式指令2  
    ...
```



SINGULAR
奇點創意

程式創客教室

機器人 / AI人工智慧 / 程式語言



雙向判斷式

Maker + Coder = Singular Super Inventor



SiNGULAR
奇點創意

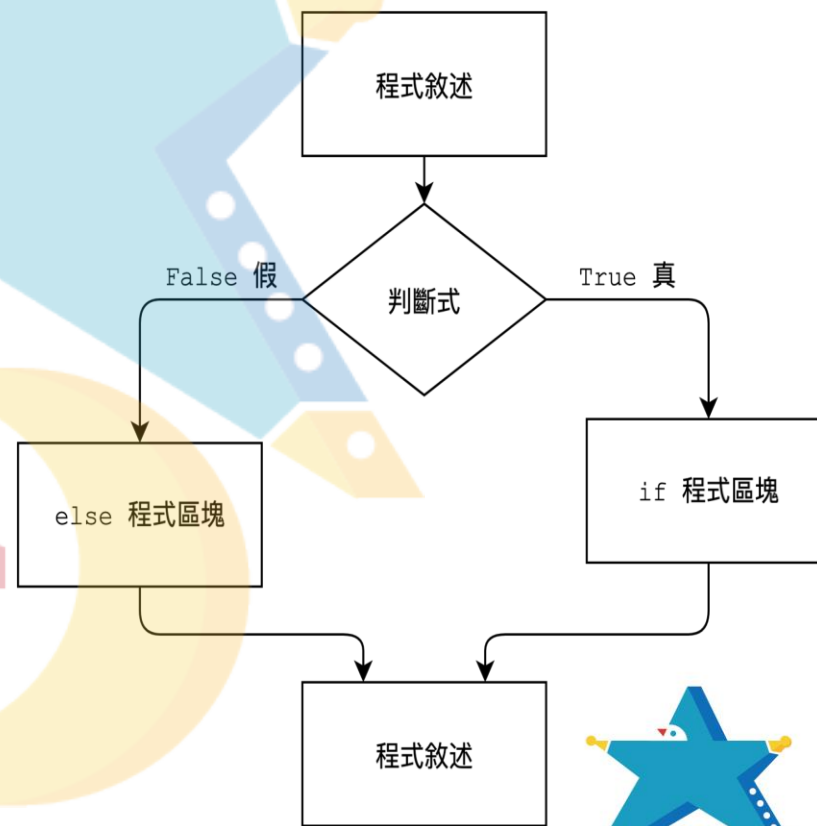
程式創客教室

機器人 / AI人工智慧 / 程式語言

if else 敘述

指令格式

```
if 判斷式:  
    程式指令1  
    ...  
else:  
    程式指令2  
    ...
```



SINGULAR
奇點創意

程式創客教室

機器人 / AI人工智慧 / 程式語言



多向判斷式

Maker + Coder = Singular Super Inventor



SiNGULAR
奇點創意

程式創客教室

機器人 / AI人工智慧 / 程式語言

if elif else 敘述

指令格式：

if 判斷式：
 程式指令1

...

elif 判斷式：
 程式指令2

...

else：
 程式指令3

...

