



SINGULAR
奇點創意

程式創客教室

機器人 / AI人工智慧 / 程式語言

機器人 / AI人工智慧 / 程式語言

Python MAT 學生手冊



Maker + Coder =



Singular Super Inventor



資料型態:

資料型態	轉型	舉例
整數	int()	1, -1, 0
浮點數	float()	1.0, 3.3,
布林	bool()	True, False
字串	str()	"abc", '123'
空資料	Null	Null

資料型態常用函數:

函式	語法	說明
input()	<code>val = input("提示使用者")</code>	讓使用者由鍵盤輸入資料。 輸入：型態為字串 輸出：型態為字串
type()	<code>val = type(10)</code> <code>>>> int</code> <code>val = type(1.1)</code> <code>>>> float</code> <code>val = type(True)</code> <code>>>> bool</code> <code>val = type("abc")</code> <code>>>> str</code>	取得型態類別。 輸入：要取得的型態 輸出：型態類別
print()	<code>print("Hi Singular")</code>	顯示資訊。
#	註解	不會被程式解讀



資料格式化 %:

參數	語法	說明
%d	print('I %d' % 94)	10 進位整數方式輸出
%3d	print('I %3d' % 94)	3 個字元寬度顯示整數，不滿 3 個則補空白
%03d	print('I %03d' % 94)	3 個字元寬度顯示整數，不滿 3 個則補 0
%f	print('I %f' % 94.12)	將浮點數以 10 進位方式輸出。
%6.2f	print('I %6.2f' % 94.12)	浮點數表示 6 位浮點數，小數位數為 2 位，不滿 6 位則補空白。
%06.2f	print('I %06.2f' % 94.12)	浮點數表示 6 位浮點數，小數位數為 2 位，不滿 6 位則補 0。
%s	print('I %s' % "abc")	使用 str() 將字串輸出

資料格式化 Format:

格式：{0:d}.format(x)

0：輸出的位置

d：輸出的型態

x：輸出的值

參數	語法	說明
%d	print("I {0:d}".format(94))	10 進位整數方式輸出
%3d	print("I {0:3d}".format(94))	3 個字元寬度顯示整數，不滿 3 個則補空白
%03d	print("I {0:03d}".format(94))	3 個字元寬度顯示整數，不滿 3 個則補 0
%f	print("{0:f}".format(94.12))	將浮點數以 10 進位方式輸出。
%6.2f	print("{0:6.3f}".format(94.12))	浮點數表示 6 位浮點數，小數位數為 2 位，不滿 6 位則補空白。
%06.2f	print("{0:06.3f}".format(94.12))	浮點數表示 6 位浮點數，小數位數為 2 位，不滿 6 位則補 0。
%s	print('I %s' % "abc")	使用 str() 將字串輸出



運算式子:

$a = 100$

運算子	運算	範例	相當於	結果
+	加法	$b = a + 2$		$b = 102$
-	減法	$b = a - 2$		$b = 90$
*	乘法	$b = a * 2$		$b = 200$
/	除法	$b = a / 2$		$b = 50$
+=	加法	$a += 2$	$a = a + 2$	102
-=	減法	$a -= 3.$	$a = a - 3.$	97.0
*=	乘法	$a *= 4$	$a = a * 4$	400
/=	小數除法	$a /= 4$	$a = a / 3$	25.0
//=	整數除法	$a //= 3$	$a = a // 3$	33
%=	取餘數	$a \% = 3$	$a = a \% 3$	1
**=	次方	$a ** = 0.5$	$a = a ** 0.5$	10.0

字串處理:

運算	語法	結果
字串加法	<code>"abc" + "123"</code>	<code>"abc123"</code>
字串乘法	<code>"abc" * 2</code>	<code>"abcabc"</code>



串列處理:

運算	語法	結果
串列加法	<code>[1, 2, 3] + [4, 5, 6]</code>	<code>[1, 2, 3, 4, 5, 6]</code>
串列乘法	<code>[1, 2, 3] * 2</code>	<code>[1, 2, 3, 1, 2, 3]</code>
修改元素	<code>L = [1, 2, 3]</code> <code>L[0] = 'A'</code>	<code>['A', 2, 3]</code>
取子串列	串列[起始值:中止值:跳幾格] <code>L = [0, 1, 2, 3, 4]</code> <code>L[0:3]</code> <code>L[3:5]</code> <code>L[-3:]</code> <code>L[0:3:2]</code> <code>L[::-2]</code>	 <code>[0, 1, 2]</code> <code>[3, 4]</code> <code>[2, 3, 4]</code> <code>[0, 2]</code> <code>[4, 2, 0]</code>

比較運算子:

比較運算子	說明
<code>A == B</code>	判斷 A 是否 相於 B
<code>A != B</code>	判斷 A 是否 不等於 B
<code>A > B</code>	判斷 A 是否 大於 B
<code>A < B</code>	判斷 A 是否 小於 B
<code>A >= B</code>	判斷 A 是否 大於等於 B
<code>A <= B</code>	判斷 A 是否 小於等於 B
比較運算子	說明

邏輯運算子:

邏輯運算子	說明
<code>C = not A</code>	C 的結果為反相 bool。
<code>C = A and B</code>	A, B 同時為 True, C 才等於 True, 其他條件 False。
<code>C = A or B</code>	A, B 其中之一為 True, C 就等於 True, 其他條件 False。



運算式優先順序

順序	運算子
1	()括號
2	**次方
3	+(正數)、-(負數)
4	*(乘法)、/(除法)、%(餘數)、/(取商)
5	+(加法)、-(減法)
6	==、!=、>、<、>=、<= (比較運算)
7	not
8	and、or
9	=、+=、-=、*=、/=、%=、//=、**=

判斷式:

格式	語法格式	舉例
單項判斷式	if 判斷條件: 判斷式成立要執行的程式	if a==0: print("good")
雙項判斷式	if 判斷條件: 條件成立要執行的程式 else: 條件不成立要執行的程式	if a==0 : print("good") else : print("bad")
多項判斷式	if 判斷條件 1: 條件 1 成立要執行的程式 elif 判斷條件 2: 條件 2 成立要執行的程式 else: 條件不成立要執行的程式	if a==0 : print("great") elif b==0 : print("good") else: print("bad")
元素判斷式	if a in L: a 有在 L, 回傳 True, a 沒有在 L, 回傳 False	L = [a, b, c] if a in L: print("good")
字串判斷式	if a in str: a 有在字串裡, 回傳 True, a 沒有在字串裡 L, 回傳 False	str = "abcdef" if a in str: print("good")



迴圈:

格式	說明	舉例
while	重複執行迴圈內的敘述程式，直到判斷式不成立，則跳出回圈。	while True: print("Hi Singular")
for	指定重複執行的次數。	for num in range(3): print("Hi Singular")
for	指定回圈 起始值，中止值，每次都迴圈變數都會增加 1，迴圈變數等於中止值，則跳出回圈。	for num in range(1, 3): print("Hi Singular")
for	指定回圈 起始值，中止值，遞增值，每次都迴圈變數都會依遞增值增加，迴圈變數等於中止值，則跳出回圈。	for num in range(0, 3, 2): print("Hi Singular")
while else	執行 while 迴圈，當迴圈正常結束時，執行 else 區塊裡面的程式。	while num < 3: print(num) else: print("迴圈正常結束")
for else	執行 for 迴圈，當迴圈正常結束時，執行 else 區塊裡面的程式。	for num in range(1, 4, 2): print(num) else: print("迴圈正常結束")



迴圈結構特殊指令：

指令	說明	舉例
break	執行 while or for 迴圈，當迴圈遇到 break 的時候，就跳出迴圈	<pre>i = 1 while i < 7: if i == 4: break print(i) i += 1</pre> <hr/> <pre>for num in range(1, 6, 2): print(num) if (num == 3): break</pre>
continue	執行 while or for 迴圈，當迴圈遇到 continue 的時候，就直接執行下一次迴圈	<pre>i = 0 while i < 6: if i == 3: continue print(i) i += 1</pre> <hr/> <pre>for i in range(0, 6): if i == 3: continue print(i)</pre>



錯誤處理

指令	說明	舉例
try : except :	當錯誤發生的時候，處理錯誤，程式不會因為錯誤而停止。	try: A / B except: print("wrong")
try: except err :	當錯誤發生的時候，處理特定錯誤，程式不會因為錯誤而停止。	try: A / B except ZeroDivisionError: print("wrong")

載入模組:

說明	舉例	使用方式
載入模組	import random	random.randrange(5)
載入可用的模組，並重新命名子模組名稱。	from random import randrange as rd	rd(5)

隨機模組

模組	說明	舉例
randrange	在指定範圍內依遞增值隨機取出一正整數，不含中止值。	random.randrange(中止值) random.randrange(起始值, 中止值, 遞增值) random.randrange(3) random.randrange(0, 10, 2)
randint	隨機選擇介於 a 與 b 之間的整數 N (a <= N and N <= b)，含中止值 b	random.randint(a, b) random.randint(0, 10)



讀寫檔案-

說明	語法	舉例
開啟檔案	<code>fd = open(檔案名字, 操作模式)</code>	<code>fd = open("test.txt", "r")</code>
讀檔案	<code>fd.write("要寫進檔案的資訊")</code>	<code>fd.write("Hi Singular")</code>
寫檔案	<code>rd_file = fd.read()</code>	<code>rd_file = fd.read()</code>
關閉檔案	<code>fd.close()</code>	<code>fd.close()</code>

操作模式參數

參數	敘述
r	讀檔案，檔案不存在，回報錯誤
r+	讀寫檔案，檔案不存在，回報錯誤
w	寫檔案，檔案不存在，創建檔案
w+	讀寫檔案，檔案不存在，創建檔案
a	續寫檔案，檔案不存在，創建檔案
a+	續讀寫檔案，檔案不存在，創建檔案

建立與使用函式:

說明	舉例
建立函式 <code>def 函式名稱(輸入變數 1, 輸入變數 2 ..):</code> 輸入: 變數 1, 變數 2 輸出: <code>return</code> 的數值	<pre>def name_add(name1, name2): return name1 + name2</pre>
使用函式 回傳值 = 函式名稱(變數 1, 變數 2..)	<pre>last_name = name_add('a', 'b') print(last_name) # >>> last_name = ab</pre>



MATH 函式:

說明	舉例
取餘數含小數點	<code>math.fmod(x, y)</code>
取絕對值	<code>math.fabs(x)</code>
向下取整數	<code>math.floor(x)</code>
向上取整數	<code>math.ceil(x)</code>
分解指數	<code>math.frexp(x)</code>

Strftime 格式化日期:

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

