

Python

Console

command=console=terminal=終端機

cd 更換工作資料夾

python 進入interactive模式或執行py檔

interpreter 直譯器 compiler 編譯器

python 用interpreter，用直譯器的語言又稱scripting language

interactive mode 是指一行一行跑程式，行首會有">>>"稱為prompt(提示字元)的符號

script mode 是將程式打成一個檔案在丟到直譯器裡一次執行的模式

idle 是指 python GUI，也就是使用者介面，會有附終端機，e.g. visual studio

editor 是編輯器，類似記事本功能，要執行還要將檔案丟到console裡，e.g. notepad++

存成py檔可以直接在console執行，若不是則前面要打python

python 可以用"\n"換行

EOL : end of line

syntax error : 語法錯誤

函數

input() : 預設回傳型態是字串

print() :

type() :

int() :

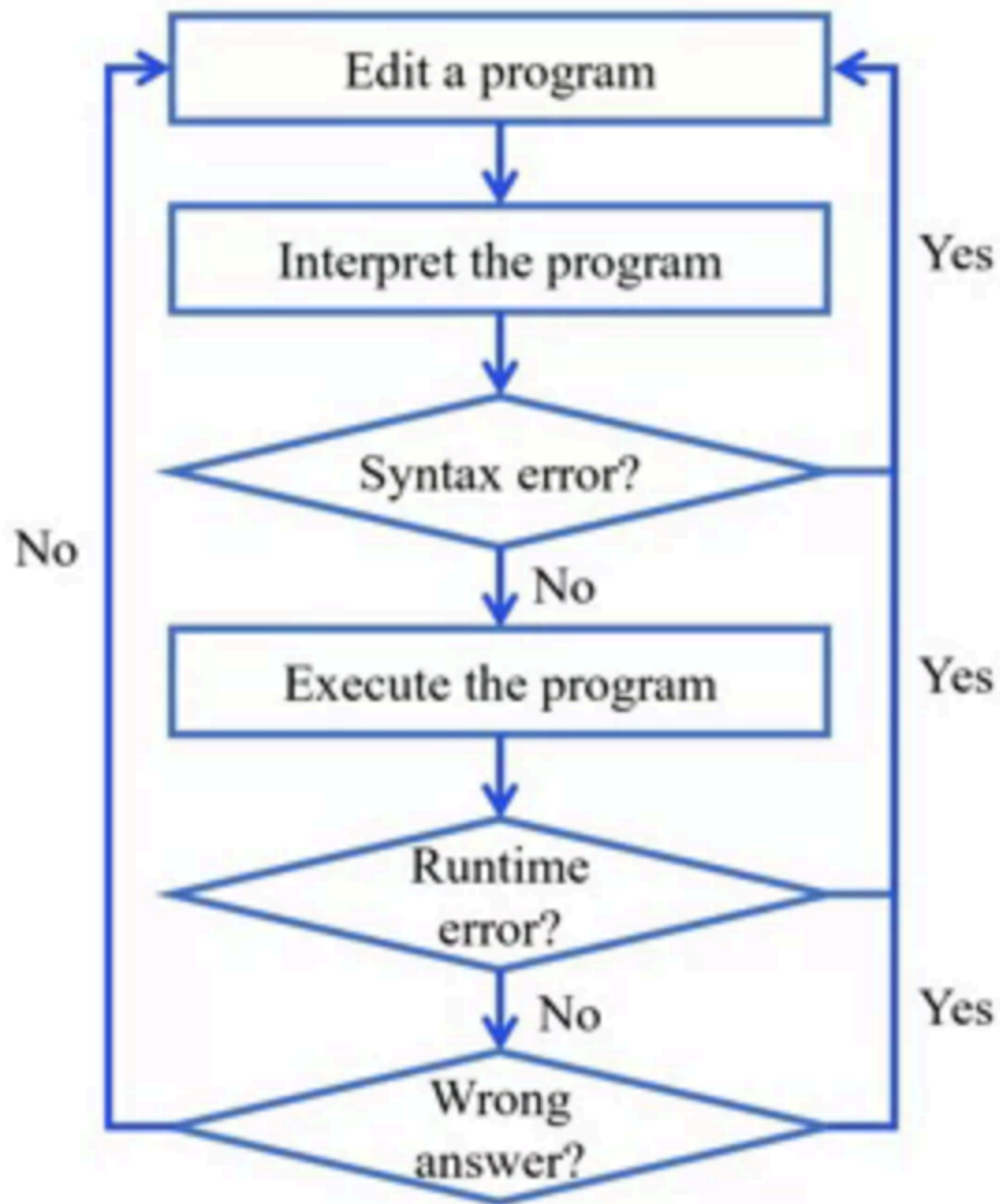
float() :

str() :

% :

// :

** :



casting : 改變資料型態

小數變整數 無條件捨去

字串變整數 無法執行

len() :

if block 的縮排不一樣會有compile error

boolean variable: True False

bool() : 預設是false

not and or:

註解: "#" """"可以當註解

Course Feedback

	整數 <code>int()</code>	浮點數 <code>float()</code>	字串 <code>str()</code>	布林 <code>bool()</code>
整數	原樣	直接轉為浮點數	直接轉為字串	<code>0</code> -> <code>False</code> 其他 -> <code>True</code>
浮點數	無條件捨去為整數	原樣	直接轉為字串	<code>0.0</code> -> <code>False</code> 其他 -> <code>True</code>
字串	只有數字表示的整數 才能直接轉換 其他 -> <code>ValueError</code>	只有數字表示的浮點數才能 直接轉換 其他 -> <code>ValueError</code>	原樣	<code>''</code> -> <code>False</code> 其他 -> <code>True</code>
布林	<code>True</code> -> <code>1</code> <code>False</code> -> <code>0</code>	<code>True</code> -> <code>1.0</code> <code>False</code> -> <code>0.0</code>	<code>True</code> -> <code>'True'</code> <code>False</code> -> <code>'False'</code>	原樣

Course Feedback

符號	概念	舉例	結果
<code>\'</code>	插入'	<code>print('I\'m fine.')</code>	I'm fine.
<code>\"</code>	插入"	<code>print('I am \"fine\".')</code>	I am "fine".
<code>\n</code>	換行	<code>print('I am \nfine.')</code>	I am fine.
<code>\t</code>	插入 Tab	<code>print('I am \tfine.')</code>	I am fine.
<code>\\</code>	插入\	<code>print('I\\am\\fine.')</code>	I\am\fine.

上面視為字串所以要加引號

ord(字元)=整數

chr(整數)=字元

range(起點, 終點, 間隔)

str.split(): 以空白鍵將字串分割成list, 或者在括弧內用其他關鍵字分割

len(list)

list乘法: a = [0]*3 > a = [0, 0, 0]

list.append(component)

Examples of list operations

- What does the following program do?

```
✓ lst = [3, 1, 4, 1, 5, 9]
✓ lst.append(2)
print(lst) # [3, 1, 4, 1, 5, 9, 2]
✓ lst.sort() ✓
print(lst) # [1, 1, 2, 3, 4, 5, 9]
✓ lst.reverse() ✓
print(lst) # [9, 5, 4, 3, 2, 1, 1]
print(lst.index(4)) # 2
```

```
lst.insert(4, "Hi")
print(lst) # [9, 5, 4, 3, 'Hi', 2, 1, 1]
print(lst.count(1)) # 2
lst.remove(1)
print(lst) # [9, 5, 4, 3, 'Hi', 2, 1]
print(lst.pop(3)) # 3
print(lst) # [9, 5, 4, 'Hi', 2, 1]
```

np問題:經驗法則heuristic algorithm解決, 無最佳演算法

lpt 演算法(least processing time):先從小到大將作業排序, 目前的作業排給loading最輕的人, 最慘會比最佳答案多1/3倍的時間

greedy sol: 目前最好的方法, 目前最短的距離

使用測資的方法:將測資用輸入的方法打成txt檔, 放在py檔同個路徑, 在console裡輸入

```
python xxx.py < yyy.txt
```

list 是一種指標: list A = [1,2,3] list B = list A list A = [1,2,2] > list B = [1,2,2]

定義二維陣列

isinstance(var, type): 確定變數是不是這個資料型態, 回傳布林值

mutable immutable: list/set/ dict

ASCII > BIG 5 > Unicode(UTF-8, UTF-16)

-*- coding: utf8 -*- 可以將python設定為使用該種版本文字編碼

字串格式(總長度+小數位數+資料類型)

總長度: 規範字串輸出的長度, 前面可加"+"代表顯示正負號 "-"代表向左對齊 "0"代表前面補零或是不限長度

小數後長度:代表小數點後要顯示幾位, 可省略

資料類型: s d f三種代表字串數字跟浮點數

當字串有跨行時, 可以再換行前加入\, 或是用小括弧將字串整個包起來, 讓python省略掉換行輸入

'{:0f}'.format(num) 四捨五入到小數點後零位

tuple

- 定義: immutable的list
- 單個元素要用(1,)
- tuple(iterable): 將變數轉換成tuple
- list(iterable): 將變數轉換成list
- sorted(tuple) = list

zip, lambda, map

- zip(iterable, iterable) = (iterator, iterator)
- 將兩個iterable結合成一個, 以較短的那個為長度
- lambda x: f(x)
- map(function, iterable): 產生一個序列每一項都是function作用在原本的元素

dictionary

- 宣告空字典: aaa = dict()
- 添加元素: aaa['一'] = 'one'
- for i in dict 代表取dict內所有的keys
- key 必須是immutable

- `dict[key]` 回傳value
- `dict.get(key)`回傳key對應的value或是None
- `dict.get(key, default)`回傳key對應的值若沒有key則回傳預設值，但不會影響原字典
- `dict.setdefault(key, default)`回傳key對應的值若沒有key則回傳預設值，會影響原字典
- `dict.keys()` 回傳所有key的list
- `dict.values()` 回傳所有value的list
- `dict.items()` 回傳所有key value 的tuple的list

set

- `set = {}`
- 會將重複的元素刪除
- `set.add(新元素)`
- `set.remove(元素)`
- 聯集 `set|set` 交集 `set&set` 差集`set-set` 錯集 `set ^ set`

copy, deepcopy

- 用在mutable上，如: list, dict, set
- `obj.copy()`
- `import copy\obj = copy.deepcopy(obj)`

datetime

- `import datetime`
- `d2 = datetime.datetime(2022, 2 ,2 , 2, 2, 2)`
- `d2.date()`
- `d2.time()`
- `d2.date().weekday()`
- `d2.day` `d2.hour` 指的是資料attribute

- `d3 = datetime.timedelta(days = 4, seconds = 900)`
- `timedelta` 可以與 `datetime` 相加， 或者 `datetime - datetime = timedelta`
- `datetime` to string: `datetime.strftime(格式)`
- 格式: `%Y%m%d%H%M%S` 年月日時分秒
- 格式: `%B%I%p%A` 英文月份 12小時制小時 AM/PM 星期
- string to `datetime`: `datetime.datetime.strptime(字串, 格式)`

開檔關檔

- 開檔 → 操作 → 關檔
- `handler = open('filename', 'mode', encoding = 'type')`
- `filename`最好是絕對路徑，有反斜線的話要再加一次反斜線或者在最前面加`r''`
- mode: `r, w, r+, rb, wb, a`
- 關檔: `handler.close()`
- `handler.write()`
- with `open()` as handler:
 - do something
 這樣就不用關檔

except

- try:
 - do something
- except `errortype`:
 - do something