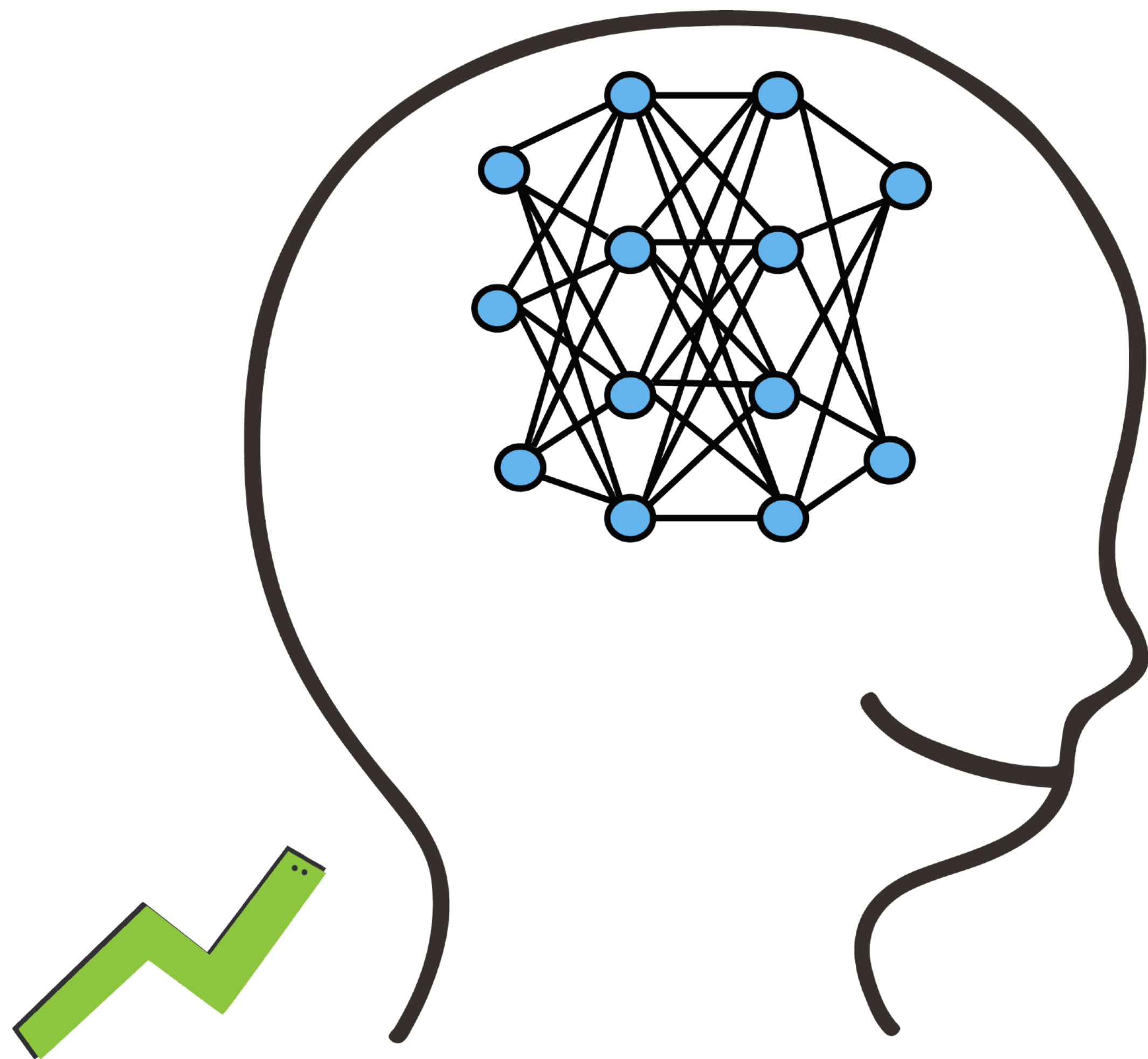


Python 基礎語法

The Basic Grammar of Python



變數型態與基本運算

Variable Types and Basic Computing

變數型態 (Variable Type)

- 布林 (Boolean)
 - True(1) / False(0)
- 整數 (Integer)
 - 24, 100 ...
- 浮點數 (Float)
 - 3.1416 ...
- 字串 (String)
 - 'Hello', "Hello" ...

Notes

- 確認變數型態：type()
e.g. type(3.14) -> float
- Python 3 整數無大小限制
- 字串可用單引號或雙引號



命名規則

- 可以用來命名的字元
 - 小寫英文字母 (a-z)
 - 大寫英文字母 (A-Z)
 - 數字 (0-9)
 - 底線 (_)
- 規則：開頭不可以是數字

Notes

▸ Python 3 統一使用Unicode (萬國碼)，包含中文字也都可用來命名 (但不建議)

e.g. 變數 = 3

```
In [1]: 變數 = 3
```

```
In [2]: 變數
```

```
Out[2]: 3
```



保留字

- 不能使用這些字來命名

False, class, finally, is, return

None, continue, for, lambda, try

True, def, from, nonlocal, while

and, del, global, not, with

as, elif, if, or, yield

assert, else, import, pass

break, except, in, raise

```
In [3]: True = 3
```

```
File "<ipython-input-3-679355094e21>", line 1
```

```
True = 3
```

```
^
```

```
SyntaxError: can't assign to keyword
```



基本運算

運算子	功能	e.g.
+	加	$2 + 3 = 5$
-	減	$2 - 3 = -1$
*	乘	$2 * 3 = 6$
/	除	$3 / 2 = 1.5$
//	取高斯符號，即[x]（不大於數x的最大整數）	$3 // 2 = 1$
%	模數/mode（取餘數）	$11 \% 3 = 2$
**	指數	$2 ** 3 = 8$

Notes

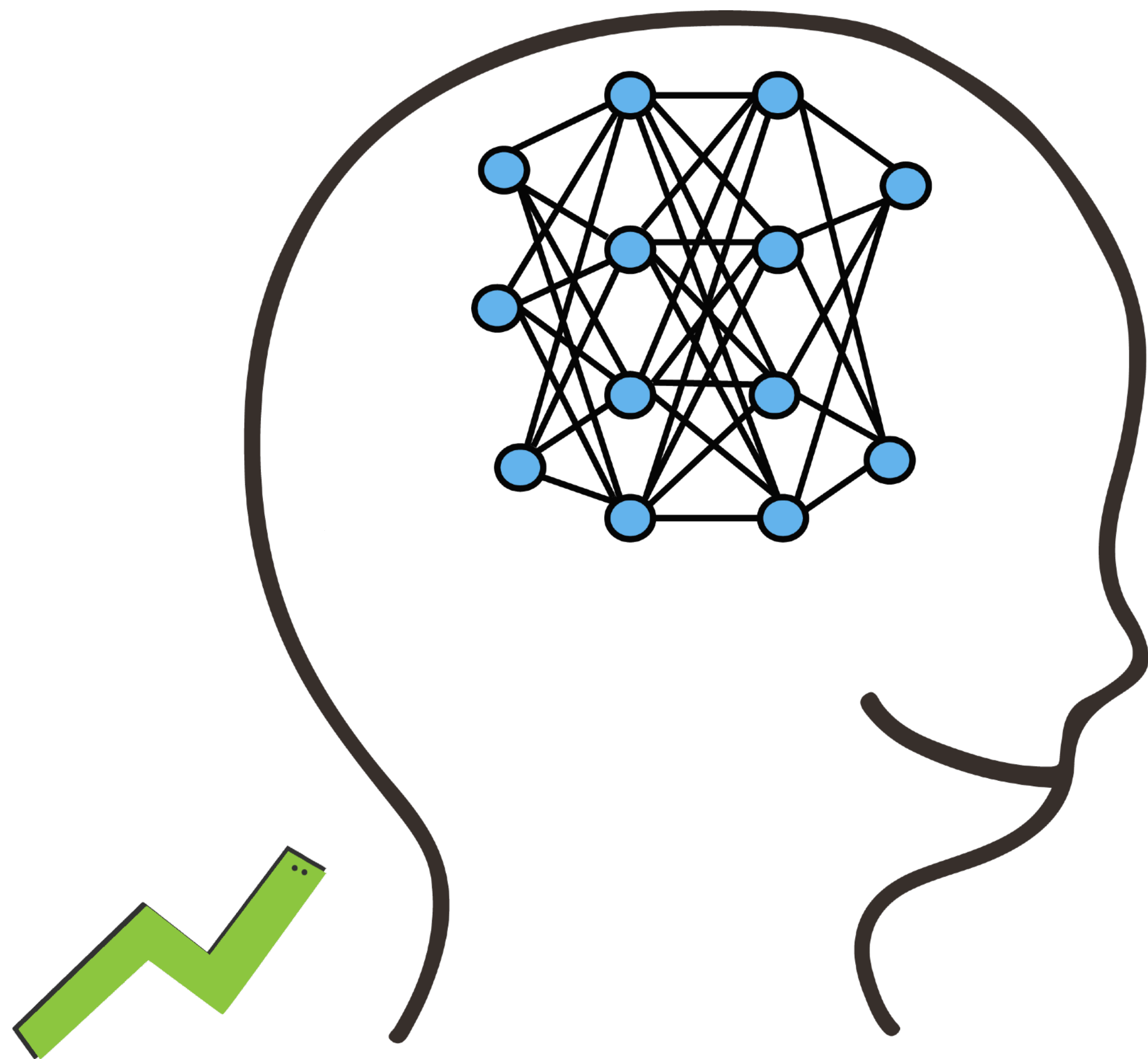
► 運算規則：先乘除後加減

e.g. $2 + 3 * 2 = 8$



資料型態轉換

- 轉換成整數：`int()`
 - e.g. `int(12.3) = 12` (無條件捨去)
- 轉換成浮點數：`float()`
 - e.g. `float('12.3') = 12.3`
- 轉換成字串：`str()`
 - e.g. `str(12.3) = '12.3'`



字串處理

String Manipulation



Print

- 輸出至螢幕

- ▶ e.g. `print("Hello")`

```
In [4]: print('Hello')
```

Hello

- 以空白串接字串或數值（會被轉成字串）

- ▶ e.g. `print("Hello","Python",123)`
- ▶ Hello Python 123

Notes

- ▶ Print in Python 2 and 3
e.g. `print "Hello"` (Python 2)
e.g. `print("Hello")` (Python 3)



引號

- 單引號 (') 和雙引號 (") 效果相同
- 若字串中需使用單或雙引號，則須用另一種引號括住
 - e.g. 'Hello' 等於 "Hello"
 - e.g. "I'm Ben."



跳脫字元

跳脫字元指的是字元本身的直接撰寫已經被程式語言用來作為語法的一部份，以下為常用跳脫字元及實作示範：

- `\'`：單引號
- `\"`：雙引號
- `\\`：單一反斜線
- `\n`：換行
- `\t`：tab

```
print('1. Hello \'World\' ')\nprint('2. Hello \"World\" ')\nprint('3. Hello \\World')\nprint('4. Hello \\nWorld')\nprint('5. Hello \\tWorld')
```

```
1. Hello 'World'\n2. Hello \"World\"\n3. Hello \\World\n4. Hello\nWorld\n5. Hello      World
```



字串串接

- 字串串接 (+)
 - e.g. `str = "Hello" + "Python"`
 - `HelloPython`
- 字串複製 (*)
 - `str = "Hello" * 3 + "Python"`
 - `HelloHelloHelloPython`



字串字元取用

- 字串長度
 - `len(str)`
- 字元取用 (i可為負值，代表從最後開始數，-1為最後一個位置)
 - `str[i]`：第i個字元 (從0開始算起)
 - `str[i:j]`：從i到j (不包含位置j)
 - `str[:j]`：從開頭 (0) 到j (不包含位置j)
 - `str[i:]`：從i到最後
 - `str[i:j:k]`：從i到j (不包含位置j)，間隔k

Notes

- Key: 包含頭(i)不包含尾(j)
- Python 的切片(Slicing)規則適用於許多地方，包含後續課程的串列(List)、矩陣、資料表(DataFrame)等資料格式，請務必熟悉。



字串切割與合併

- str.split(分隔符號)

- ▶ e.g. str.split(",")

```
In [5]: s = 'A, B, C, D'
        s.split(',')
Out[5]: ['A', ' B', ' C', ' D']
```

- str(分隔符號).join(iterable) (iterable置入可迭代的物件如list)

- ▶ e.g. “,”.join(word_list)

```
In [6]: slist = ['A', ' B', ' C', ' D']
        ','.join(slist)
Out[6]: 'A, B, C, D'
```



大小寫轉換

- `str.upper()`：轉大寫
 - e.g. `“hello”.upper()`
 - HELLO
- `str.lower()`：轉小寫
 - e.g. `“HELLO”.lower()`
 - hello



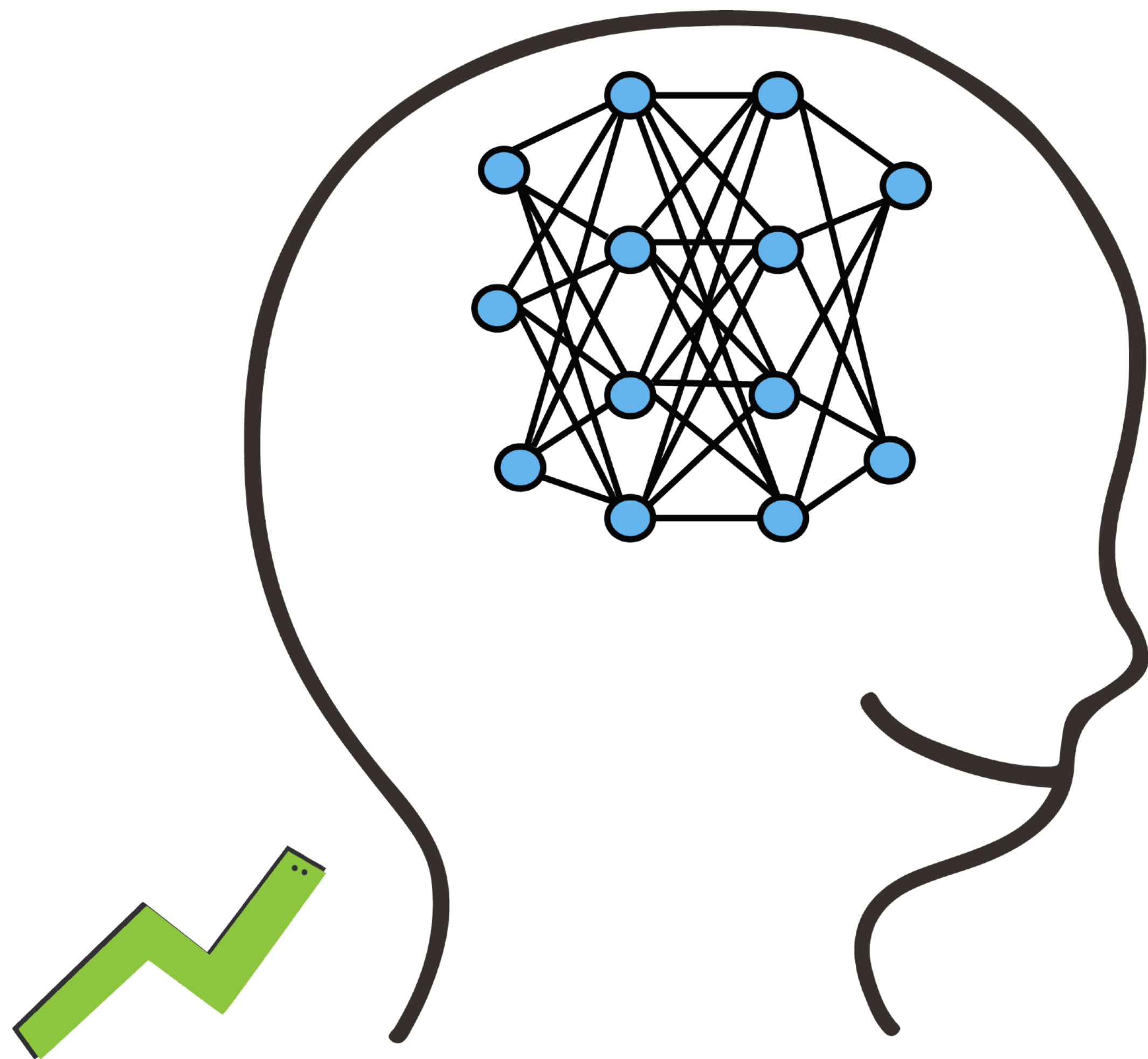
其他

- 判斷是否是大/小寫：`str.isupper()` / `str.islower()`
- 判斷是否是數字：`str.isnumeric()`
- 字串取代：`str.replace()`
-



Reference

- Python官方文件 - 4.7.1. String Methods
 - <https://docs.python.org/3.6/library/stdtypes.html>



Python 程式結構語法

The Grammar of Python Program
Structure



註解與延續多行

- 單行註解：`#`
 - e.g. `# comments`
- 多行註解：`"""`
 - e.g. `"""comment line 1`
`comment line 2 """`
- 延續多行：`\`
 - e.g. `line 1 \`
`line 2`



程式結構語法

- 皆使用縮排 (tab)
 - 不需要如begin, end、{}、;等來定義程式結構

In [2]:

```
x = 1
if x == 1:
    print('yes')
else:
    print('no')
```

yes



if-else

if (condition1):

statement1 #若滿足condition1則執行statement1

elif (condition2):

statement2 #若滿足condition2則執行statement2

else:

statement3 #若皆不滿足condition1、2則執行statement3



while-loop

#若滿足condition1，則進入迴圈中執行，
直到不滿足為止

```
while (condition1):  
  
    statement1
```

Notes

► **break : 跳出此層迴圈**

e.g. break the while-loop when c is 3

```
c = 0
```

```
while True:
```

```
    c += 1
```

```
    if (c == 3):
```

```
        break
```

► **continue : 跳過後續的程式，重頭執行迴圈**



for-loop

- 產生數字序列：range()
 - e.g. range(5)：0~4
 - e.g. range(1,5): 1, 2, 3, 4
 - e.g. range(1,10,2): 1,3,5,7,9 （間隔2）
- for-loop 計數
 - e.g. for i in range(5): #range(5)會回傳0,1,2,3,4

print (i)



for-loop (cont.)

- 存在於Tuple / 串列 (list) / 字典 (dictionary) / 集合 (set) 的項目

- e.g.

```
clist = ['A', 'B', 'C']
```

```
for i in clist:
```

```
    print (i)
```

- 串列生成 (Pythonic的撰寫方式)

- e.g. `nlist = [num for num in range(1,10) if num%2 == 0]`

- output: `[2,4,6,8]`



自訂函式

- def 函式名稱 (參數)

- e.g.

```
def add(x, y):
```

```
    print(x+y)
```

- 呼叫自訂函式：

- e.g. add(3,2)



References

- Python官方文件：<https://docs.python.org/3.6/>
- Python官方文件 - 教程：<https://docs.python.org/3.6/tutorial/>