

argparse

argparse介紹

我們常常在使用程式的時候會帶入參數, `argparse`就是來幫你把這些參數解析的

參數大致分成兩種: 1) 位置型參數 2) 選項型參數

位置型參數

位置型參數就像是我們平常在使用函式的時候帶入的參數, 照著順序帶入下來, 並且一定要帶入

In [1]:

```
1 import argparse
2
3 # prog: program名稱, 如果沒帶入就是你py的名字
4 parser = argparse.ArgumentParser(prog="MyGame")
5 # 增加一組位置參數name
6 parser.add_argument("name", help="Your Name")
7 # 增加一組位置參數surname
8 parser.add_argument("surname", help="Surname")
9 # --help是內建的, 會自動產生
10 parser.print_help()
11 # 你可以藉由parse_args來做試驗
12 p1 = parser.parse_args("Elwing Chou".split())
13
14 print("[Name]:", p1.name)
15 print("[Surname]:", p1.surname)
```

usage: MyGame [-h] name surname

positional arguments:

name	Your Name
surname	Surname

optional arguments:

-h, --help show this help message and exit

[Name]: Elwing

[Surname]: Chou

選項型參數

選項型參數通常會提供兩種方法讓你使用, 1) `-`(一個橫線): 短選項 2) `--`(兩個橫線): 長選項, 你可以兩者都提供, 也可以只提供一個

In [3]:

```
1 import argparse
2
3 parser = argparse.ArgumentParser(prog="MyGame")
4
5 # 透過type可以直接讓你的值變成對的type
6 parser.add_argument("-r", "--round", help="round to n digits", type=int)
7
8 parser.add_argument("height", help="height", type=float)
9 parser.add_argument("weight", help="weight", type=float)
10
11 parser.print_help()
12
13 # 短選項
14 p = parser.parse_args("-r 2 175 75".split())
15 print("bmi:", round(p.weight / (p.height / 100) ** 2, p.round))
16
17 # 長選項
18 p = parser.parse_args("--round 2 185 75".split())
19 print("bmi:", round(p.weight / (p.height / 100) ** 2, p.round))
20
21 # 如果沒有帶入, 會是 None
22 p = parser.parse_args("185 75".split())
23 print("round:", p.round)
24 print("bmi:", round(p.weight / (p.height / 100) ** 2, p.round))
```

usage: MyGame [-h] [-r ROUND] height weight

positional arguments:

height	height
weight	weight

optional arguments:

-h, --help	show this help message and exit
-r ROUND, --round ROUND	round to n digits

bmi: 24.49

bmi: 21.91

round: None

bmi: 22

選項型預設值

你可以提供預設值給選項型的參數

In [4]:

```
1 import argparse
2
3 parser = argparse.ArgumentParser(prog="MyGame")
4
5 parser.add_argument("-r", "--round", help="round to n digits", type=int, default=2)
6 p = parser.parse_args("").split()
7 print("round:", p.round)
```

round: 2

額外小技巧

限制值

你可以把值限制在你想要的範圍內

In [6]:

```
1 import argparse
2
3 parser = argparse.ArgumentParser(prog="MyGame")
4
5 parser.add_argument("-t", "--type", help="Game Type", type=int, choices=[0, 1, 2])
6
7 parser.print_help()
8 p = parser.parse_args("-t 2".split())
9 print("type:", p.type)
10 # 這個會出錯
11 # parser.parse_args("-t 3".split())
```

usage: MyGame [-h] [-t {0,1,2}]

optional arguments:

-h, --help show this help message and exit
-t {0,1,2}, --type {0,1,2} round to n digits

type: 2

多個參數

你可以在 nargs 表明你要接受多少個 n(帶入整數) *(任意個) +(至少一個), 如果有表明 nargs, 回來的型態必定為 list

In [8]:

```
1 import argparse
2
3 parser = argparse.ArgumentParser(prog="MyGame")
4
5 parser.add_argument("-s", "--sum", help="sum all numbers", type=float, nargs="*")
6
7 parser.print_help()
8 p = parser.parse_args("--sum 2.3 3.2 4.3".split())
9 print("numbers:", p.sum)
10 # 沒有帶入會是空的List
11 p = parser.parse_args("--sum".split())
12 print("numbers:", p.sum)
```

usage: MyGame [-h] [-s [SUM [SUM ...]]]

optional arguments:

-h, --help show this help message and exit
-s [SUM [SUM ...]], --sum [SUM [SUM ...]] sum all numbers

numbers: [2.3, 3.2, 4.3]

numbers: []

無值型

如果你需要無值型, 有帶入是True, 沒帶入是False, 你可以配合action來使用

In [9]:

```
1 import argparse
2
3 parser = argparse.ArgumentParser(prog="MyGame")
4
5 # store_false就是store_true的相反
6 parser.add_argument("-e", "--enabled", help="enable hard mode", action="store_true")
7
8 parser.print_help()
9 # 有帶入就是True
10 p = parser.parse_args("--enabled".split())
11 print("enabled:", p.enabled)
12 # 沒帶入就是False
13 p = parser.parse_args("").split())
14 print("enabled:", p.enabled)
```

usage: MyGame [-h] [-e]

optional arguments:

-h, --help show this help message and exit
-e, --enabled enable hard mode

enabled: True

enabled: False

版本訊息

你可以使用action="version"配合version參數來打印出版本訊息, 使用version的話不會進主程式, 印出版本後直接退出

In [16]:

```
1 import argparse
2 version = 2.0
3 parser = argparse.ArgumentParser(prog='PROG')
4 parser.add_argument("-v", "--version", action="version", version='%(prog)s {}'.format(\
5 # 會直接結束程式
6 parser.parse_args("--version".split()))
```

PROG 2.0

An exception has occurred, use %tb to see the full traceback.

SystemExit: 0