

範例程式碼：

```
1  #Python 程式設計練習：銀貸償付計算(def 版)
2
3  print('#本程式為銀行貸款償付計算程式，由屏東大學周國華老師設計#')
4  print()
5  def LoanPayment(a,b,c):
6      d=a/((1-pow((1+(c/1200)),(-b*12)))/(c/1200))
7      print()
8      print('按照您的貸款金額及條件，您每月應攤還的本息是新台幣',format(d,',.2f'),'元')
9      print()
10     i=eval(input('想繼續使用請按 1，離開請按 0：'))
11     if i==1:
12         print()
13         main()
14     else:
15         print()
16         print('謝謝使用本程式！')
17 def main():
18     print('請依照以下順序，輸入貸款總金額、貸款總年數及貸款年利率：')
19     x=eval(input('1.貸款總金額(新台幣)[請勿包含千分位逗號]：'))
20     y=eval(input('2.貸款總年數：'))
21     z=eval(input('3.貸款年利率(例如：年利率 2.53%，請輸入 2.53)：'))
22     LoanPayment(x,y,z)
23 if __name__ == '__main__':
24     main()
```

程式碼解析：

1. #為單行註解符號，Python 直譯器看到#就會忽略那一行的內容。若要寫多行註解，就要在註解內容的前後用三個雙引號(""")或單引號('')包住。
2. 程式碼中的空行不會出現在程式執行畫面中，若要在程式執行畫面中出現空行，要使用 print()。
3. print 函式提供螢幕列印功能，要在螢幕上呈現的內容須在括弧內以單引號或雙引號前後包住(前後引號需一致，不能一單一雙)。如果只想在螢幕上出現空行，就打 print()。本範例在單引號內的#符號會被當成字串內容處理，而非註解符號。
4. Python 直譯器會將 Python 程式從第一行程式碼依序往下執行。但直譯器看到 def 函式時，會略過去，因為 def 函式必須透過呼叫才能執行。本範例程式，在執行完第 4 行程式碼 print()後，會略過兩個 def 函式，跳至第 23 行執行 if __name__=='__main__' 這一行程式碼。
5. __name__ 是每一支 Python 程式(稱為 Python 模組)都有的特殊內建屬性，若該模組是主程式，

則此屬性的值會設定為'__main__'；若該模組是被另一支程式載入(import)，則此屬性的值會設定為該模組的檔案名稱(不含.py 尾稱)。

6. 單一等號(=)是**指定運算子**，可將等號右邊的值指定給等號左邊的變數。

● 雙等號(==)是**比較運算子**，X==Y 代表 X 和 Y 相等。

7. Python 提供的選擇性敘述有 if、if...else、if...elif...else 三種，語法如下

(1) if

if 條件運算式:

主體敘述

(2) if...else

if 條件運算式:

主體敘述 1

else:

主體敘述 2

(3) if...elif...else

if 條件運算式:

主體敘述 1

elif 條件運算式:

主體敘述 2

else:

主體敘述 3

[註：上述的 elif 程式區塊可以有無限多個]

if、elif 的條件運算式後面及 else 的後面要加冒號(:)，主體敘述要內縮，通常是內縮 4 格。

8. 本範例程式若為單獨執行的主程式，則最後一段 if 敘述的條件運算式即成立，會進入主體敘述呼叫 main()函式。
9. 函式(function)是解決某一問題的片段程式，可以被重複呼叫使用，其定義是以 def 開頭，後接函式名稱及一組小括號，小括號內可以包含參數，小括號後是冒號(:)，換行後內縮 4 格接該函式的主體敘述。
10. Python 是以冒號後換行內縮(通常為 4 格)來區隔程式片段，其他程式語言大部分是以大括弧{...}來區隔程式片段。
11. 本範例程式進入 main()函式後，先以三個 eval(input(..))敘述來取得呼叫 LoanPayment(..)函式所需的三個參數。
12. input 函式可讓使用者在螢幕上輸入指定內容，輸入之內容(無論是數字、文字或符號)均預設為字串(str)型態。input 括弧內可填入要使用者輸入之內容的提示性文字，提示性文字要用單引號或雙引號前後包住。使用者會直接在該提示性文字後面輸入內容。若要讓使用者在提示性文字的下一行輸入內容，可在提示性文字的最後面加上\n 換行符號。\\n 須包在提示性文字的引號內。
13. eval 函式可將括弧內的字串(本範例為透過 input(..)函式取得)進行適當評價，並將其內容轉換為對應的資料型態。
14. Python 是動態定型語言(dynamically-typed language)，變數不需要事先宣告其資料型態，可在執行時期(run time)直接根據給定的值來決定其型態(故稱為**動態定型**)。本範例中的 x、y 兩個變數

(正常情況下會輸入整數)會被動態定型為 `int` 型態，`z`、`d` 兩個變數(正常情況下會輸入或計算出小數)則會被動態定型為 `float` 型態。

15. `main()`函式呼叫 `LoanPayment(..)`函式時，`x`, `y`, `z` 三個變數的值會成為對應的 `a`, `b`, `c` 三個參數的值。
16. 函式通常以英文字命名，且第一個英文字的第一個字母按慣例要小寫，後面每個英文字的第一個字母要大寫。若小寫字母易引起混淆時，也可以改成大寫。例如本範例第 5 行的 `loanPayment` 函式，`l` 容易被誤認為數字 1 或英文字母 `i` 的大寫，故改為大寫 `L`。
17. `LoanPayment(..)`函式內的第一行程式碼(本範例第 6 行)，在計算每月應攤還的本息 `d`，計算式使用到 `pow(..)`這個內建函式，以及年金現值因子。
18. Python 的內建函式

Python 直譯器有內建數十個函式，隨時都可以使用這些函式。以下按照英文字母排序列出。

內建函式				
<code>abs()</code>	<code>delattr()</code>	<code>hash()</code>	<code>memoryview()</code>	<code>set()</code>
<code>all()</code>	<code>dict()</code>	<code>help()</code>	<code>min()</code>	<code>setattr()</code>
<code>any()</code>	<code>dir()</code>	<code>hex()</code>	<code>next()</code>	<code>slice()</code>
<code>ascii()</code>	<code>divmod()</code>	<code>id()</code>	<code>object()</code>	<code>sorted()</code>
<code>bin()</code>	<code>enumerate()</code>	<code>input()</code>	<code>oct()</code>	<code>staticmethod()</code>
<code>bool()</code>	<code>eval()</code>	<code>int()</code>	<code>open()</code>	<code>str()</code>
<code>breakpoint()</code>	<code>exec()</code>	<code>isinstance()</code>	<code>ord()</code>	<code>sum()</code>
<code>bytearray()</code>	<code>filter()</code>	<code>issubclass()</code>	<code>pow()</code>	<code>super()</code>
<code>bytes()</code>	<code>float()</code>	<code>iter()</code>	<code>print()</code>	<code>tuple()</code>
<code>callable()</code>	<code>format()</code>	<code>len()</code>	<code>property()</code>	<code>type()</code>
<code>chr()</code>	<code>frozenset()</code>	<code>list()</code>	<code>range()</code>	<code>vars()</code>
<code>classmethod()</code>	<code>getattr()</code>	<code>locals()</code>	<code>repr()</code>	<code>zip()</code>
<code>compile()</code>	<code>globals()</code>	<code>map()</code>	<code>reversed()</code>	<code>__import__()</code>
<code>complex()</code>	<code>hasattr()</code>	<code>max()</code>	<code>round()</code>	

19. Python 的內建函式是可以直接呼叫使用的，本課程截至目前為止已使用過 `eval()`、`format()`、`input()`、`pow()`、`print()`這些內建函式。若要使用非內建函式，則須在 Python 模組的最上端透過 `import` 指令載入具有該函式的外部模組。
20. 年金現值公式：
$$\frac{1-(1+i)^{-n}}{i}$$
，式中 `i` 代表利率，`n` 代表期數。
21. `pow()`函式通常以 `pow(a,b)`的形式呼叫，直譯器會給出 `a` 的 `b` 次方的值。
22. 銀行貸款利率都是指年利率(`c`)，本例為每月攤還，所以在代入公式前須轉換為月利率。
23. 本範例的 `d` 值，亦可直接帶年金現值公式寫成 `d=a/((1-(1+(c/1200))**(-b*12)))/(c/1200)`
24. `print` 函式要在螢幕列印的內容(註：術語稱為物件 `object`)，可以用多個逗點隔開。物件可以是直接用兩個單(或雙)引號表達的字串，也可以是 `print` 之前已給定內容值的變數。若列印的物件不只一個，螢幕列印時每個物件之間預設會用一個空格隔開。
 - `print` 函式的預設語法：
`print(*objects, sep=' ', end='\n', file=sys.stdout, flush=False)`
 - 若想讓多個物件之間在列印時用其他方式隔開，可以設定 `sep` 參數。上述預設語法中的 `sep` 後面的單引號內包含一個空格。若改為 `sep=""`，列印時物件之間就沒有空格。若改為 `sep=','`，列印時物件之間就會以逗號隔開。
 - `print` 內要列印的物件可以用 `format` 函式加以格式化。`format` 函式的語法：`format(X,Y)`，`X`

是要格式化的對象，Y 是格式化的方式。本範例程式碼 `format(d, ',.2f')` 中，d 變數是要列印的物件，其格式化的方式是用兩個單引號包住 `','`，引號內第一個逗點是千分位格式，表示 d 變數的整數數字將以三位一逗點的千分位格式表達；引號內的 `.2f` 表示 d 變數是一個取到小數點以下兩位的浮點數。

25. `LoanPayment()` 函式的最後一段敘述是 `if...else` 選擇性敘述。若使用者輸入的 i 變數值為 1，則呼叫 `main()` 函式，重新執行一次銀貸攤還計算；若使用者輸入的 i 變數值為 0 (或任何非 1 的值)，則結束本程式。