

## 套件編譯語法 - package

### (A) 套件語法 - package

基本上，製作套件與類別/物件方法非常相似。比較特殊的地方是，某些方法被包裝成套件後，大多不會限制使用者的身分如何，也都將其宣告成 public 屬性較為普遍，宣告 package 的語法如下：

套件宣告語法：	範例：
<pre>package package_name; public class class_name {     public data_type variable_name;     ....     public return_type method_name{         ....     }     .... }</pre>	<pre>package dateTool; public class Date {     public int year;     public int month;     public int day; }</pre>

### (B) 編譯套件 - javac -d

套件原始檔案製作完成之後，則可將其編譯成套件方法庫。編譯之前必須考慮套件的類別檔案儲存於何目錄下。基本上，在原始檔案內需指定套件名稱，編譯後類別檔案將儲存於該套件名稱的目錄下，但該套件名稱的目錄又在何處？除了編譯時必須指定儲存目錄外，也必須設定 classpath 環境變數，命令格式如下：

設定 classpath 環境變數：	範例：
> set classpath=path_name	> set classpath=G:\Examples\timeDate;.
編譯套件命令：	範例：
> javac -d path_name file_name	> javac -d G:\Examples\timeDate Date.java

### (C) 主程式導入 - import

當然自行開發的套件也可儲存於原 java 套件底下，但為了方便管理大多不會這樣；如果沒有儲存原 java 預定目錄下，欲導入運用又有稍微麻煩。欲導入自行開發的套件，也須利用 classpath 指名套件（如 set classpath=G:\Examples\timeDate;.）儲存目錄，導入語法如下：

套件導入語法：	範例：
<pre>import class_name; ..... public class main_class {     ....     .... }</pre>	<pre>import dateTool.dateTool; import dateTool.Date;  public class Ex8_3 {     .... }</pre>