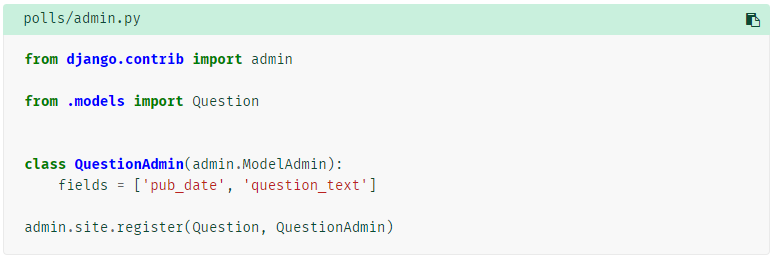
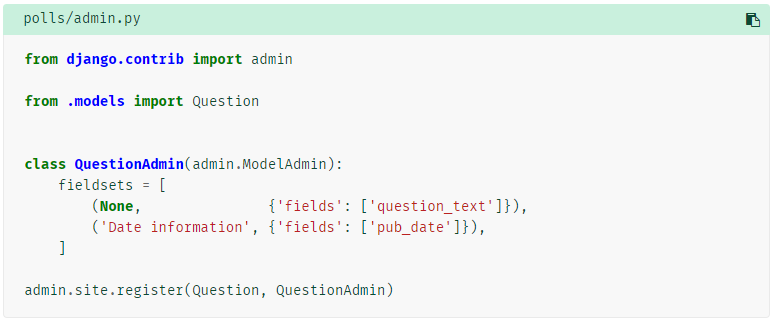
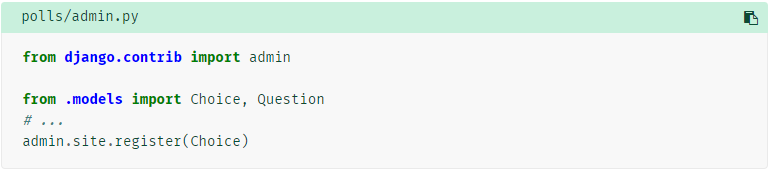
**Django app, part 7**



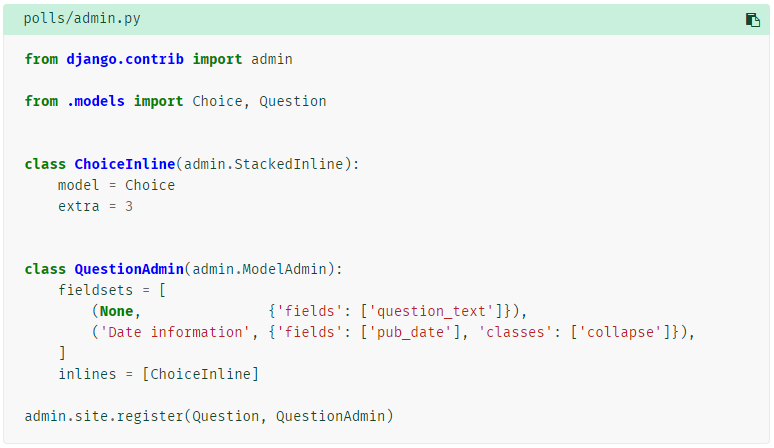
開啟admin.py，客製化後台的介面，設定QuestionAdmin這個區塊內放置發布日期(pub\_date)及問題內容(question\_text)這兩種資料，並且告訴django，Question跟QuestionAdmin需要後台的管理(註冊這兩個model)。



更改QuestionAdmin區塊資料的呈現方式，第一項資料內容為問題內容(question\_text)沒有表頭(第一個參數設為none)、第二個參數為要放的資料，第二項資料表頭顯示Date information，第二個參數放置發布日期的資料。



因為在這個app中，一個問題(Question)有多個選項(Choice)，因此我們在後台註冊Choice，但其實這樣做比較沒效率，因此要更改為當新增一個問題時，同時會出現多個選項可以編輯。



建立ChoiceInline，給予StackedInline的樣式，指定model的來源為Choice，當call ChoiceInline時，預設出現3個Choice，另外在QuestionAdmin中放入ChoiceInline，最後再將原本在後台註冊的Choice改回Question及QuestionAdmin。



修改ChoiceInline的呈現樣式為TabularInline。



修改問題清單的呈現方式，原本只會呈現question\_text，改為呈現question\_text(問題)、pub\_data(發布日期)、was\_published\_recently(是否為最近發布)。



去poll的model中修改was\_published\_recently這個model，加入判斷是否為近期發布的計算，參照的資料有pub\_date，資料型態為布林值，預設為True，資料呈現的標題為Published recently?。



加上django所提供的filter功能，讓我們可以自由切換Django所提供不同期間的資料。



加上搜尋欄位，我們可以透過鍵入問題的部分關鍵字查詢。



在settings.py中加上DIRS這行程式碼，去修改當load templates時，加入路徑，讓django找的到我們所建立的template。接著在templates下建立admin的資料夾，將複製(系統有一個預設的檔案)的base\_site.html存在這。



開啟base\_site.html檔編輯，將預設的Django administration改為Polls Administration。