

ActiveSupport

What is ActiveSupport

- ActiveSupport 是 Rails 裡的工具函式庫，它也擴充了一些 Ruby 標準函式庫。除了被用在 Rails 核心程式中，你也可以在你的程式中使用。

擴充 Object

<http://api.rubyonrails.org/classes/Object.html>

blank? and present? (I)

- blank?會傳回true的有：
 - nil跟false物件
 - 空的或是只有space的字串
 - 空的Array或Hash
- present?則與blank?相反

blank? and present? (II)

- 在console中試看看：

- `nil.blank?`
- `false.present?`
- `[].blank?`
- `{:a => ""}.present?`
- `"".blank?`
- `" ".blank?`

try

- 某個物件如果是nil，那對他呼叫method則會回傳nil。
- 很常用在Model#column_name的檢查。
- 在Console中試看看：
 - User.first.try(:email)
 - User.all[1000].try(:name)

to_param

- 會回傳可以被作為URL的字串，預設是會回傳to_s的結果，但是有些類別有另外實作。
- 在Console中試看看：
 - Time.now.to_param
 - [1, "a", [Time.now, "weijen"]].to_param
 - 以gurupu的Group model做例子。

with_options

- 如果有好幾個statements，都需要相同的選項(option)，可以利用with_options簡化程式。
- 例如在migration中，有好幾個columns都需要: null => false，我們可以：

```
with_options :null => false do
  t.string :name
  t.string :tel
  t.string :address
end
```


in?

- 測試a物件是否在b物件中。
- 在Console中試看看：
 - l.in?(1, 2)
 - l.in?(2, 3)
 - l.in?(1..10)
 - l.in?([1,3,5])

擴充String

<http://api.rubyonrails.org/classes/String.html>

html_safe & html_safe?

- 會檢查字串是否可以呈現在網頁中。
- 在Console中試看看：
 - `"".html_safe?`
 - `s = "".html_safe`
 - `s.html_safe?`
 - `s = "<script>...</script>".html_safe`
 - `s.html_safe?`

squish

- 會將字串中的開頭與結尾的空白字元刪掉。
- 會將字串中的不可見字元(ex. \n, \r)刪掉。
- 在Console中試看看：
 - " \n foo\n\r \t bar \n".squish # => "foo bar"

first & last

- 取出字串的第一個字元字串與最後一個字元字串。
- 在Console中試看看：
 - "hello".first
 - "hello".last

starts_with? & ends_with?

- 檢查字串的第一個字元字串，或是最後一個字元字串。
- 在Console中試看看：
 - `"foo".starts_with?("f")`
 - `"foo".ends_with?("o")`

at

- 取出字串中特定位置的字元字串。
- 在Console中試看看：
 - "hello".at(0)
 - "hello".at(4)
- 其實 "hello"[0]也可以達到同樣的效果，可是用at可讀性較高。

from & to

- from回傳從某個位置到結尾的子字串。
- to回傳從開頭到某個位置的子字串。
- 在Console中試看看：
 - "hello".from(0)
 - "hello".from(2)
 - "hello".to(0)
 - "hello".to(2)

pluralize & singularize

- 回傳某個字串的複數與單數。
- 在Console中試看看：
 - "ruby".pluralize
 - "news".pluralize
 - "tables".singularize
 - "news".singularize

camelize

- "product".camelize # => "Product"
- "admin_user".camelize # => "AdminUser"
- "a/b".camelize # => "A::B"

underscore

- "Product".underscore # => "product"
- "AdminUser".underscore # => "admin_user"
- "A::B".underscore #=> "a/b"

titleize

- "hello".titleize #=> "Hello"
- "hello world".titleize #=> "Hello World"

dasherize

- "name".dasherize # => "name"
- "a_b".dasherize # => "a-b"

to_date, to_time, to_datetime

- "2010-07-27".to_date
 - # => Tue, 27 Jul 2010
- "2010-07-27 23:37:00".to_time
 - # => Tue Jul 27 23:37:00 UTC 2010
- "2010-07-27 23:37:00".to_datetime
 - # => Tue, 27 Jul 2010 23:37:00 +0000

擴充Array

<http://api.rubyonrails.org/classes/Array.html>

to & from

- 與字串的用法相同，只是回傳的是子陣列。
- 在Console中試看看：
 - `%w(a b c d).to(2)`
 - `%w(a b c d).from(2)`

second, third, fourth, & fifth

- 就是取出第幾個element。

sample

- 隨機取出element
- 在Console中試看看：
 - `%w(a b c d).sample`
 - `%w(a b c d).sample(2)`

prepend & append

- 增加element到陣列的前面與後面。
- 其實Array#unshift & Array#<< 也是同樣的功能，只是為了可讀性。

to_sentence

- 轉換為字串，不過可讀性更好。
- 在Console中試看看：
 - `%w(Earth Wind Fire).to_sentence`
 - `%w(Earth Wind Fire).to_s`

擴充Date, Time, DateTime

<http://api.rubyonrails.org/classes/Date.html>

取得最近日期

- `Date.current` (`Date.today`是取得系統時間)
- `Date.yesterday`
- `Date.tomorrow`
- 這幾個methods取出的日期是以你的app所設定的時區為準。你可以在`config/application.rb`中設定時區：
`config.time_zone = 'Taipei'`

檢查日期

- `d = "2012-05-17".to_date`
- `d.today?`
- `d.past?`
- `d.future?`

年、月

- `d.prev_year`
- `d.next_year`
- `d.beginning_of_year`
- `d.end_of_year`
- `d.prev_month`
- `d.next_month`

周

- `d.beginning_of_week`
- `d.end_of_week`
- `d.monday`
- `d.sunday`
- `d.next_week`
- `d.next_week(:sunday)`
- `d.prev_week`
- `d.prev_week(:saturday)`

日期加減

- $d + 10.\text{day}$
- $d - 5.\text{day}$
- $d + 1.\text{month}$
- $d - 1.\text{month}$
- $d + 1.\text{year}$
- $d - 2.\text{year}$