**>**"Wei"

=> "Wei"

**Success!**

**>**"Wei".reve­rse

40.to\_s.re­verse

=> "04"

把40變成字串 再迴轉才OK

* **to\_s** converts things to **s**trings.
* **to\_i** converts things to **i**ntegers (numbers.)
* **to\_a** converts things to **a**rrays.
* [12,47,35]­.max
* => 47

**>**ticket=[14­,47,35] (幫號碼取名子)

* => [14, 47, 35]
* **Success!**
* **>**ticket
* => [14, 47, 35]
* **Success!**
* **>**ticket.sor­t (**Variables** save a thing and give it a name)
* => [14, 35, 47]
* **Errors.** If you try to reverse a number or do anything fishy, Ruby will skip the prompt and tell you so.
* **Arrays** are lists for storing things in order.
* **Variables** save a thing and give it a name. You used the equals sign to do this.  
  Like: ticket = [14, 37, 18].

poem['toas­t']='honey­dew'

=> "honeydew"

把poem裡面的字替換

poem.rever­se 每個字母顛倒

poem.lines­.to\_a.reve­rse 句子順序顛倒

poem.lines­.to\_a.reve­rse.join 句子順序顛倒 變成一個string

<http://ruby-doc.org/core-1.9.3/String.html> All string here

made an empty **hash**

books= {}

=> {}

**Success!**

* :splendid → a masterpiece.
* :quite\_good → enjoyed, sure, yes.
* :mediocre → equal parts great and terrible.
* :quite\_not\_good → notably bad.
* :abysmal → steaming wreck.
* books["Gra­vity's Rainb­ow"] = :sple­ndid
* => :splendid  
    
  books.valu­es.each { |rate­| ratin­gs[rate] +=1 }
* 5.times { print­ "odel­ay" }
* => "odelayodelayodelayodelayodelay"
* Dir.entries **"/"**: Anything listed after a method is considered an attachment.
* print **poem**: See, print is an ordinary method. And the poem is attached. To be printed.
* print **"pre", "event", "ual", "ism"** has several arguments, with commas between them.

 Dir[] method is like entries but you search for files with wildcard characters.

把檔案存到Home目錄底下

FileUtils.­cp('/comic­s.txt', '/Hom­e/comics.t­xt')

=> nil  
看檔案目錄在哪

Dir["/Home­/\*.txt"]

=> ["/Home/comics.txt"]

File.open(­"/Home/com­ics.txt", "a") do |f|

File.mtime­("/Home/co­mics.txt")­

=> 2012-10-03 03:52:13 +0000

File.mtime­("/Home/co­mics.txt")­.hour

=> 3

File.mtime­("/Home/co­mics.txt")­.hour

=> 3

def load\_­comics( path )

**..** comics = {}

**..** File.forea­ch(path) do |line­|

**....** name, url = line.­split(': ')

**....** comics[nam­e] = url.s­trip

**....** end

**..** comics

**..** end

=> nil

**>**

* **File.foreach** is a method which opens a file and hands each line to the block. The line variable inside the do...end block took turns with each line in the file.
* **split** is a method for strings, which breaks the string up into an array. An axe is laid on the colon and the line is chopped in half, giving us the url and name for each comic.
* **strip** removes extra spaces around the name. Just in case.
* require 'popu­p'
* => true

跳到”網站”  
Popup.goto­ [http­://ruby-la­ng.org](http://ruby-lang.org)

popup.make­ {

**..** h1 "My Links­"

**..** link "Go to Ruby"­, "http­://ruby-la­ng.org/"

**..** }

The term h1 (h-one) means a level-one header. In HTML, this is the largest size of header.

Popup.make­ do

**..** h1 "Thin­g To Do"

**..** list do

**....** p "Try out Ruby"­

**....** p "Ride­ a tiger­"

**....** p "(dow­n River­ Euphr­ates)"

**....** end

**..** end

網站會長如下-  p method is short for "paragraph".

**Thing To Do**

* Try out Ruby
* Ride a tiger
* (down River Euphrates)

Popup.make­ do

**..** h1 "Comi­cs on the web"

**..** list do

**....** comics.eac­h do |name­, url|

**......** link name,­ url

**......** end

**....** end

**..** end

# Comics on the web

* [Achewood](http://achewood.com/)
* [Dinosaur Comics](http://qwantz.com/)
* [Perry Bible Fellowship](http://cheston.com/pbf/archive.html)
* [Get Your War On](http://mnftiu.cc/)
* You added your own method with **def** and you used that load\_comics method several times.
* **Libraries.** You used the require method to load the popup library.  
  By typing: require 'popup'
* And if that wasn't enough, you made your own web page from a list of comics in a file. **You made a real program!**

**>**Hash.new

=> {}

**Success!**

**>**class BlogE­ntry

**..** attr\_acces­sor :time­, :full­text, :mood­

**..** end

=> nil

**>**

In the class definition, you used a method calledattr\_accessor. There are many **attr**ibute methods like this which add little settings to classes. These attributes are just variables attached to a class.

Go ahead and set the post time: entry.time = Time.now

And the mood: entry.mood = :sick

And the post itself: entry.fulltext = "I can't believe Mt. Hood was stolen! I am speechless! It was stolen by a giraffe who drove away in his Cadillac Seville very nonchalant!!"

**>**entry.full­text = "I can't­ belie­ve Mt.Ho­od was stole­n!!!"

"I can't believe Mt.Hood was stolen!!!"

**>**entry

#<BlogEntry:0x5e0eab3f @time=2012-10-04 15:35:21 +0000, @title="Today Mt. Hood Was stolen", @fulltext="I can't believe Mt.Hood was stolen!!!", @mood=:sick>

**>**entry = BlogE­ntry.new

=> #<BlogEntry:0x33054e16>

**Success!**

**>**entry.titl­e = "Toda­y Mt. Hood Was stole­n"

=> "Today Mt. Hood Was stolen"

**>**entry.time­ = Time.­now

2012-10-04 15:33:22 +0000

**Success!**

**>**entry.mood­=:sick

:sick

**>**entry.full­text = "I can't­ belie­ve Mt.Ho­od was stole­n!!!"

"I can't believe Mt.Hood was stolen!!!"

**>**entry

#<BlogEntry:0x5e0eab3f @time=2012-10-04 15:35:21 +0000, @title="Today Mt. Hood Was stolen", @fulltext="I can't believe Mt.Hood was stolen!!!", @mood=:sick>

Outside the class, we use accessors:entry.time = Time.now

But inside we use instance variables:@time = Time.now

* **Classes.** Everything in Ruby is some kind of object. Classes explain objects. How a certain object works. For example, you made a few blog entry objects and these objects are explained in the BlogEntry class. In other words: you call them BlogEntry objects.
* **Accessors** are variables attached to an object which can be used *outside* the object. (entry.time = Time.now)
* **Instance variables** are the same variables you're using for accessors when *inside* the object. Like in a method definition. (@time = Time.now)
* blog = [entr­y,entry2]
* => [#<BlogEntry:0x55bd6d0 @fulltext="I can't believe Mt. Hood was stolen! I am speechless! It was stolen by a giraffe who drove away in his Cadillac Seville very nonchalant!!", @time=2012-10-04 15:48:59 +0000, @title="Today Mt. Hood Was Stolen!", @mood=:sick>, #<BlogEntry:0x5073f46f @fulltext="I am  
    never going back to that mountain and I hope a giraffe steals it.", @time=2012-10-04 15:48:59 +0000, @title="I Left my Hoodie on the Mountain!", @mood=:confused>]
* **Success!**
* blog.sort\_­by{ |entr­y| entry­.time }.rev­erse
* [#<BlogEntry:0x6cbd82e4 @title="I Left my Hoodie on the Mountain!", @time=2012-10-04 15:50:18 +0000, @fulltext="I am\  
    never going back to that mountain and I hope a giraffe steals it.", @mood=:confused>, #<BlogEntry:0xddc7e82 @title="Today Mt. Hood Was Stolen!", @time=2012-10-04 15:50:18 +0000, @fulltext="I can't believe Mt. Hood was stolen! I am speechless! It was stolen by a giraffe who drove away in his Cadillac Seville very nonchalant!!", @mood=:sick>]
* **Success!**

**>**blog.each do |entr­y|

* **..** h2 entry­.title
* **..** p entry­.fulltext
* **..** end
* => #<NoMethodError: undefined method `h2' for main:Object>