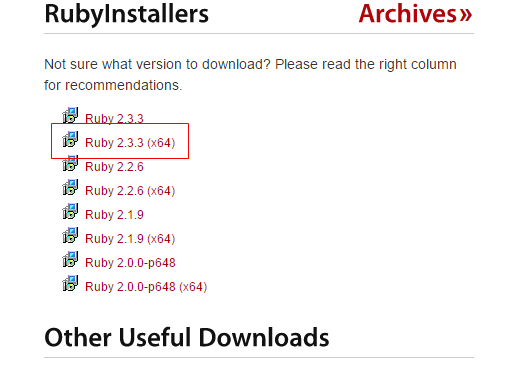
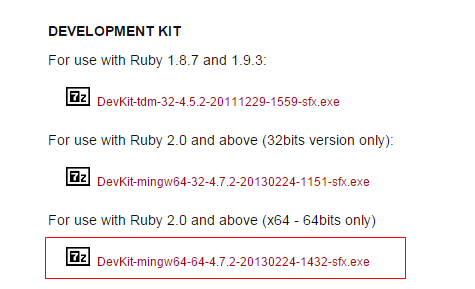
# Ruby学习分享

## 一、Ruby下载及安装

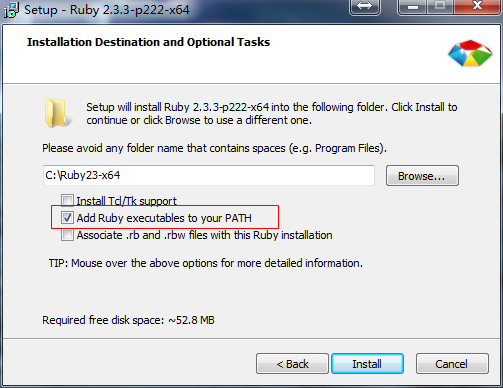
### 1.ruby on windows下载：

<http://rubyinstaller.org/downloads/>



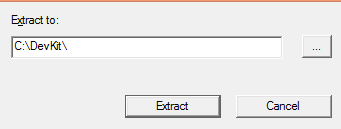


### 2.ruby安装：



勾选添加到环境变量，点击“Install”安装

### 3.安装DevKit：压缩包直接解压到自定义的文件夹内



### 4.安装完成后，进入cmd，执行如下命令：

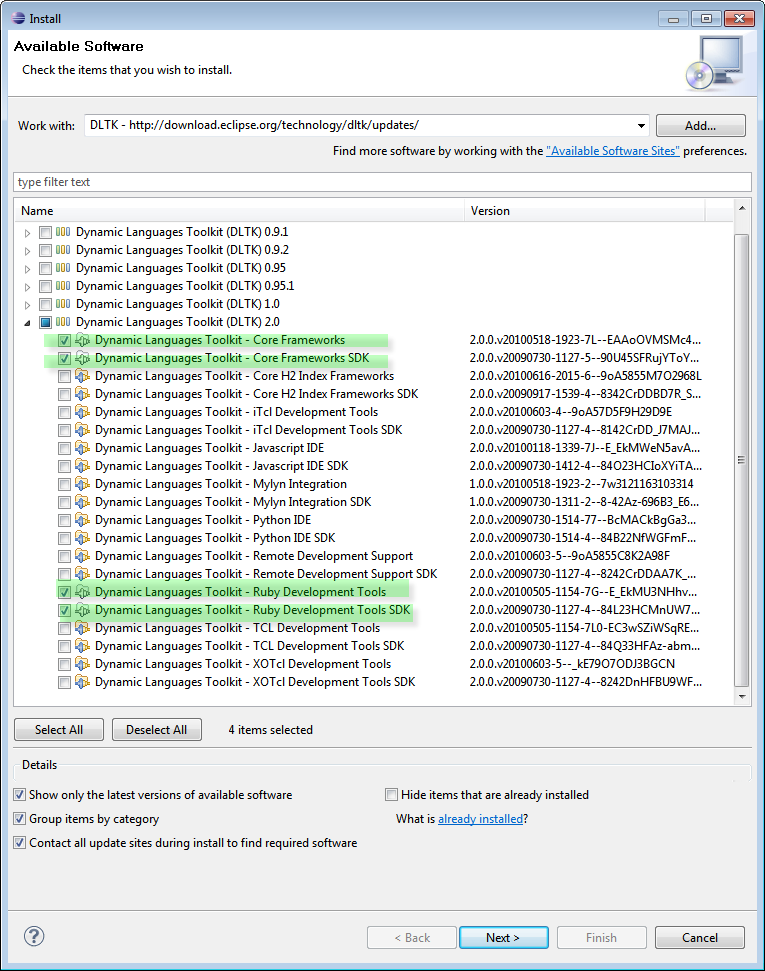
ruby -v

查看ruby版本号

## 二、eclipse的Ruby插件下载及安装（在线）：

### **1.安装eclipse的ruby插件。**

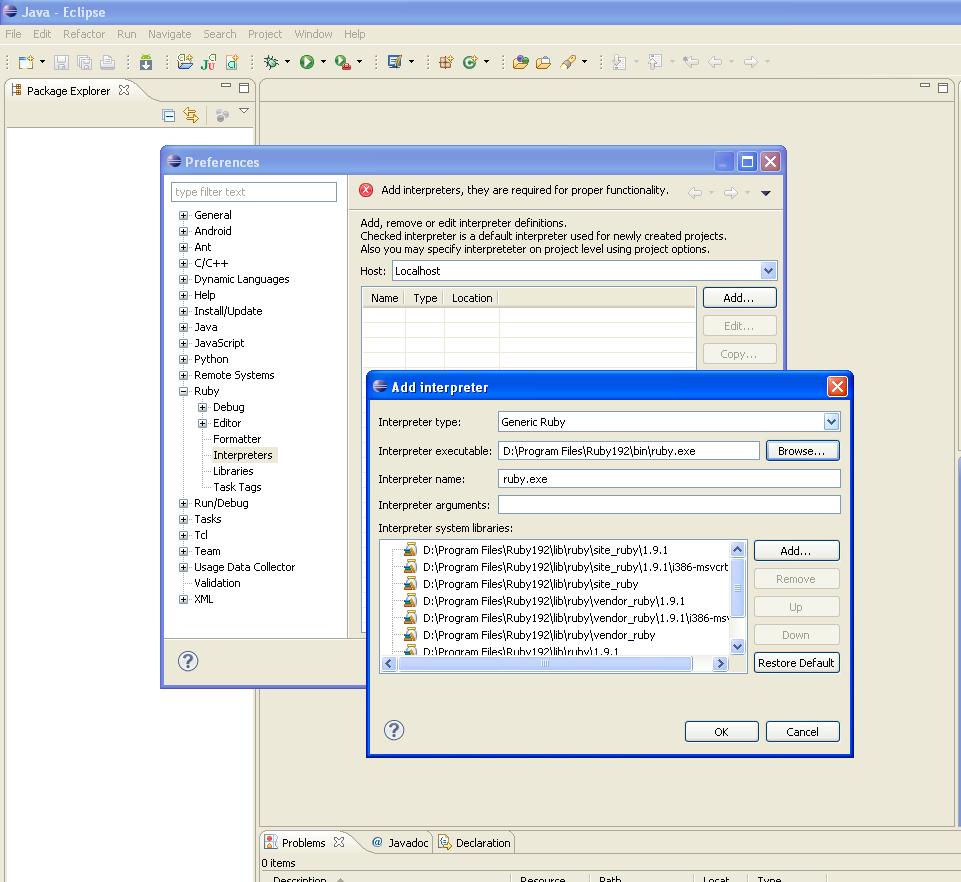
我安装的是eclipse的DLTK（Dynamic Languages Toolkit），Eclipse > help > install new software，添加这个URL是[**http://download.eclipse.org/technology/dltk/updates/**](http://download.eclipse.org/technology/dltk/updates/)，这个里面包含了几乎所有的常用插件，这里选择ruby相关的就行了，见下图：



安装完成之后重启eclipse就OK了。

### 2.设置Interpreter：

Window --> preferences --> ruby --> interpreters ，点击Add按钮，找到你的ruby安装目录，并选择ruby.exe



### 3.创建一个新的Ruby project：

File --> New --> New Project --> Ruby Project。然后添加一个ruby script文件，File --> New --> Empty Ruby Script。

在Script文件中，输入以下文字：

puts "hello world"

然后Run As > Ruby Script，便能在Console中看到输出的hello world。

### 4.尝试打开某个软件：QQ.exe

**class** Login

system 'start C:\"Program Files (x86)"\QQ\Bin\QQ.exe'

**end**

PS：在路径中，如果有“空格”，需要将有“空格的文件夹名称”用“双引号”括起来

### 5.尝试打开某个浏览器

a）打开浏览器需要使用require ‘selenium-webdriver’（使用前需要在“gem”程序包管理器内，先安装“selenium-webdriver”包，安装方法见：“b）”）

PS：RubyGems在ruby内简称“gem”是一个方便而强大的Ruby程序包管理器（ package manager）

b）打开windows命令行窗口，输入gem install selenium-webdriver，等待在线安装完成

c）输入gem list检查selenium-webdriver的版本号，显示对应版本号即安装完成

d）打开firefox、ie、chrome，

PS1： 如果需要打开firefox时出现报错，可能是selenium-webdriver与geckodriver版本不支持导致报错，那就必须下载部署geckodriver，下载后放在ruby安装目录下，比如 C:\Ruby23-x64\bin 下即可。

geckodriver.exe 下载地址：<https://github.com/mozilla/geckodriver/releases>

PS2：如果需要打开chrome，就必须下载部署chromedriver.exe，下载后放在ruby安装目录下，比如 C:\Ruby23-x64\bin 下即可。

chromedriver.exe下载地址： <http://chromedriver.storage.googleapis.com/index.html>

PS3：如果需要打开chrome，就必须下载部署IEDriverServer.exe，下载后放在ruby安装目录下，比如 C:\Ruby23-x64\bin 下即可。

IEDriverServer.exe下载地址：<http://selenium-release.storage.googleapis.com/index.html>

**require** 'rubygems'

**require** 'selenium-webdriver'

*# 打开firefox*

dr = Selenium::WebDriver.for *:firefox*

dr = Selenium::WebDriver.for *:ff*

*# 打开ie*

dr = Selenium::WebDriver.for *:ie*

dr = Selenium::WebDriver.for *:internet\_explorer*

*# 打开chrome*

dr = Selenium::WebDriver.for *:chrome*

（但是本人尝试该方法打开IE，无法打开会报错：

C:/Ruby23-x64/lib/ruby/gems/2.3.0/gems/selenium-webdriver-3.2.2/lib/selenium/webdriver/common/service.rb:80:in `stop': undefined method `poll\_for\_exit' for nil:NilClass (NoMethodError)

from C:/Ruby23-x64/lib/ruby/gems/2.3.0/gems/selenium-webdriver-3.2.2/lib/selenium/webdriver/common/service.rb:69:in `block in start'

from C:/Ruby23-x64/lib/ruby/gems/2.3.0/gems/selenium-webdriver-3.2.2/lib/selenium/webdriver/common/platform.rb:138:in `block in exit\_hook'

C:/Ruby23-x64/lib/ruby/gems/2.3.0/gems/selenium-webdriver-3.2.2/lib/selenium/webdriver/common/socket\_lock.rb:63:in `initialize': Permission denied - bind(2) for "127.0.0.1" port 5554 (Errno::EACCES)

from C:/Ruby23-x64/lib/ruby/gems/2.3.0/gems/selenium-webdriver-3.2.2/lib/selenium/webdriver/common/socket\_lock.rb:63:in `new'

from C:/Ruby23-x64/lib/ruby/gems/2.3.0/gems/selenium-webdriver-3.2.2/lib/selenium/webdriver/common/socket\_lock.rb:63:in `can\_lock?'

from C:/Ruby23-x64/lib/ruby/gems/2.3.0/gems/selenium-webdriver-3.2.2/lib/selenium/webdriver/common/socket\_lock.rb:52:in `lock'

from C:/Ruby23-x64/lib/ruby/gems/2.3.0/gems/selenium-webdriver-3.2.2/lib/selenium/webdriver/common/socket\_lock.rb:38:in `locked'

from C:/Ruby23-x64/lib/ruby/gems/2.3.0/gems/selenium-webdriver-3.2.2/lib/selenium/webdriver/common/service.rb:71:in `start'

from C:/Ruby23-x64/lib/ruby/gems/2.3.0/gems/selenium-webdriver-3.2.2/lib/selenium/webdriver/ie/bridge.rb:37:in `initialize'

from C:/Ruby23-x64/lib/ruby/gems/2.3.0/gems/selenium-webdriver-3.2.2/lib/selenium/webdriver/common/driver.rb:59:in `new'

from C:/Ruby23-x64/lib/ruby/gems/2.3.0/gems/selenium-webdriver-3.2.2/lib/selenium/webdriver/common/driver.rb:59:in `for'

from C:/Ruby23-x64/lib/ruby/gems/2.3.0/gems/selenium-webdriver-3.2.2/lib/selenium/webdriver.rb:87:in `for'

from D:/ruby/workspace/autotesting/login.rb:9:in `<class:Login>'

from D:/ruby/workspace/autotesting/login.rb:1:in `<main>'）

所以，建议不要使用IE，或使用windows命令行“system”方法打开IE：

system 'start C:\"Program Files"\"Internet Explorer"\iexplore.exe'

## 三、使用ruby打开网页并进行登录操作

### 1.打开某个url

**a）方法1：**

*#使用get方法*

url='http://im.qq.com/'

dr.get url

**b）方法2：**

*#使用navigate方法，然后再调用to方法*

url = 'http://im.qq.com/'

dr.navigate.to url

### 2.关闭浏览器

*#使用close方法 关闭本次执行打开的页面*

dr.close

*#使用quit方法 关闭所有页面*

dr.quit

PS：验证过程中使用close的方法会出现无法关闭页面的情况

### 3.进行登录操作

*#点击“登录”按钮，呼出“登录”的frame框*

dr.find\_element(*:id*=>'login').click

*#切换到“登录”的frame框*

dr.switch\_to.frame('login\_frame')

sleep 3

*#点击frame框内的“帐号密码登录”链接*

dr.find\_element(*:id*=>'switcher\_plogin').click

*#在frame框内输入帐号密码并点击【登录】*

dr.find\_element(*:id*=>'u').send\_keys("test")

dr.find\_element(*:id*=>'p').send\_keys("test")

dr.find\_element(*:id*=>'login\_button').click

### 4.完整脚本

**class** Login

*#system 'start C:\"Program Files (x86)"\QQ\Bin\QQ.exe'*

**require** 'selenium-webdriver'

*# 打开firefox*

*#dr = Selenium::WebDriver.for :firefox*

dr = Selenium::WebDriver.for *:ff*

*# 打开ie*

*#dr = Selenium::WebDriver.for :ie*

*#dr = Selenium::WebDriver.for :internet\_explorer*

*# 打开chrome*

*#dr = Selenium::WebDriver.for :chrome*

sleep 6

*#使用get方法*

*#url='http://im.qq.com/'*

*#dr.get url*

*#使用navigate方法，然后再调用to方法*

url = 'http://im.qq.com/'

dr.navigate.to url

sleep (5)

*#点击“登录”按钮，呼出“登录”的frame框*

dr.find\_element(*:id*=>'login').click

*#切换到“登录”的frame框*

dr.switch\_to.frame('login\_frame')

sleep 3

*#点击frame框内的“帐号密码登录”链接*

dr.find\_element(*:id*=>'switcher\_plogin').click

*#在frame框内输入帐号密码并点击【登录】*

dr.find\_element(*:id*=>'u').send\_keys("test") *#用test替换成真实账号*

dr.find\_element(*:id*=>'p').send\_keys("test") *#用test替换成密码*

dr.find\_element(*:id*=>'login\_button').click

dr.quit

**end**

## 四、使用命令行执行ruby文件（后缀.rb）

### 1.命令行执行ruby文件的格式

去除**class** XXX ... **end**，在命令行进入该文件的目录，使用“ruby 文件名.rb”即可执行该文件

*ps：为什么要用命令行来运行ruby文件*

*答：很多时候，使用轻盈的文本编辑器来写脚本会比用IDE来的更方便，当然这就需要一个比较好用的文本编辑器（推荐SciTE，本次分享会一并分享该编辑器）*

### 2.实例（以上述登录脚本为例）

只要去掉**class** XXX ... **end**，即可执行

**require** 'selenium-webdriver'

*# 打开firefox*

*#dr = Selenium::WebDriver.for :firefox*

dr = Selenium::WebDriver.for *:ff*

*# 打开ie*

*#dr = Selenium::WebDriver.for :ie*

*#dr = Selenium::WebDriver.for :internet\_explorer*

*# 打开chrome*

*#dr = Selenium::WebDriver.for :chrome*

sleep 6

*#使用get方法*

*#url='http://im.qq.com/'*

*#dr.get url*

*#使用navigate方法，然后再调用to方法*

url = 'http://im.qq.com/'

dr.navigate.to url

sleep (5)

*#点击“登录”按钮，呼出“登录”的frame框*

dr.find\_element(*:id*=>'login').click

*#切换到“登录”的frame框*

dr.switch\_to.frame('login\_frame')

sleep 3

*#点击frame框内的“帐号密码登录”链接*

dr.find\_element(*:id*=>'switcher\_plogin').click

*#在frame框内输入帐号密码并点击【登录】*

dr.find\_element(*:id*=>'u').send\_keys("test") *#用test替换成真实账号*

dr.find\_element(*:id*=>'p').send\_keys("test") *#用test替换成密码*

dr.find\_element(*:id*=>'login\_button').click

dr.quit

### 3.对比登录时的错误提示

在做测试的时候，很多时候需要对比“实际结果”和“预期结果”

假设我们的“预期结果”="您还没有输入验证码！"

那我们可以通过获取元素内的数据来对比（出现未输入验证码时的错误信息）：

*#获取元素“id”为“err\_m”的数据，以text定义为“err”*

err=dr.find\_element(*:id*=>'err\_m').text

*#将预期结果定义为“dd”*

dd=("您还没有输入验证码！")

*#判断“err”和“dd”*

**if**

err.eql? dd *#使用“eql?”方法用于对比两个值*

puts "P" *#打印“P”*

**else** puts "F" *#打印“F”*

**end**

将上述代码插入到“输入帐密并点击【登录】”后，“关闭浏览器”前

运行并查看输入结果：



## 五、操作EXCEL

1.