

# IERG 4210

# Web Programming and

# Security

2017-2018 Term 1  
Tutorial 1

# About this course

Instructor: Prof. Kehuan ZHANG [[khzhang@ie](mailto:khzhang@ie), SHB716]

Lecture:

- ▶ Tue 12:30-14:15 @ MMW 703
- ▶ Wed 10:30-11:15 @ LKK 101

Tutorial:

- ▶ T1: Tue, 16:30-17:15 @ Lady Shaw Building, LT2
- ▶ T2: Wed, 11:30-12:15 @ Li Dak Sum Building 214 (**unconfirmed**)

Tutors:

Office hour (tentative):

- ▶ DONG Shuaike [[ds016@ie](mailto:ds016@ie), SHB726]    Thu, 14:30-15:30
- ▶ XU Fenghao [[xf016@ie](mailto:xf016@ie), SHB726]                      Thu, 15:30-16:30
- ▶ An appointment ahead of time is preferred.

# About this course

All information can be found at

- ▶ Blackboard
- ▶ <http://course.ie.cuhk.edu.hk/~ierg4210> (CU/IE  
VPN needed)

Facebook:

- ▶ <https://www.facebook.com/groups/11550109187650>
- ▶ CUHK IERG4210 2017

# About this course

## Assessment Methods

- |                      |     |
|----------------------|-----|
| ▶ <b>Assignments</b> | 40% |
| ▶ In-class Quizzes   | 5%  |
| ▶ Middle Term Exam   | 20% |
| ▶ Final Exam         | 35% |

# About the assignment

Develop an e-Commerce website with PayPal payment gateway

- ▶ Example: parknshop.com, walmart.com

Your website should be able to prevent well-known attacks (e.g. CSRF, SQL injection)

# About this course

There are 7 phases.

- ▶ For each phase, you need to push your code to your github repository.

Last phase is peer-hacking.

- ▶ You get marks when you can hack your classmate, and he will lose marks
- ▶ Get back some marks when the bug is fixed.

Deadlines are firm. Late submission penalty.

If you submit earlier in the first 4 phases, you may enjoy deadline extension in the later phase. (**96 hours max**)

- ▶ 48 hours in advance = 24 hours extension ( $\text{ext} = 24 \ (\text{t} \div 48)$  )

Details to be announced later

# Assignment inspection

To make sure you can and have completed the basic parts.

Tutors will randomly pick some of the students' works for inspection.

Code Cross checking with previous submissions

- ▶ To prevent code copying and reuse

# Assignment inspection

If we find that you cannot get **50%** of the requirements in a submission, you **MUST** give a **midterm demo**.

Try your best to finish the assignments on time.

Time of midterm demo to be decided

# Git

- ▶ In our course, you need to maintain your code via Git.
  - ▶ What is Git.
  - ▶ Installment on Linux, Mac and Windows.
    - ▶ Give an example on windows.
      - ▶ msysGit (<https://git-scm.com/download/win>)
    - ▶ Recent versions of Linux and Mac have pre-installed Git.
  - ▶ Basic terms and operations

# Git & Github

DO NOT LEAK  
id\_rsa (not  
id\_rsa.pub)



Configure Git

Generate  
Key Pairs

- ▶ Configure Git first.
  - ▶ Launch GitBash and input
- ▶ Create a Github account.
- ▶ Set SSH Key
  - ▶ Use the email you register
  - ▶ id\_rsa private key; id\_rsa.pub public key
  - ▶ Add SSH key
    - ▶ Settings -> SSH and GPG Keys
    - > New SSH

Keys -> copy your public key here.

- ▶ Try communicate with Github
  - ▶ ssh -T git@github.com

login to your  
github repo

```
dsk@DESKTOP-VLA4SUI MINGW64 /d/Github
$ git config --global user.name "Shuaike DONG"

dsk@DESKTOP-VLA4SUI MINGW64 /d/Github
$ git config --global user.email "xxxxxxxxx@gmail.com"
```

```
dsk@DESKTOP-VLA4SUI MINGW64 /d/Github
$ ssh-keygen.exe -t rsa -C "xxxxxxxxx@gmail.com"
```

```
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/dsk/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/dsk/.ssh/id_rsa.
Your public key has been saved in /c/Users/dsk/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:1L...DMSL7E2V7Q16...@DESKTOP-VLA4SUI
The key's randomart image is:
+---[RSA 2048]---+
| . o . oo |
| o + .++. |
| o . o* |
```

```
dsk@DESKTOP-VLA4SUI MINGW64 /d/Github
$ ssh -T git@github.com
The authenticity of host 'github.com (192.30.253.112)' can't be established.
RSA key fingerprint is SHA256:1L...DMSL7E2V7Q16...@DESKTOP-VLA4SUI
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'github.com,192.30.253.112' (RSA) to the list of known hosts.
Hi DongShuaike! You've successfully authenticated, but GitHub does not provide shell access.
```

# Github Student Pack

- ▶ Many developer tools free for students
- ▶ Add your school email for verification
- ▶ After verification, you can get unlimited private repositories !!

- ▶ Use different school emails (link/your department emails) to try
- ▶ More deliberate plan:
  - ▶ For web development course assignment... etc.

The image shows two screenshots of the GitHub interface. The top screenshot is the 'Student Developer Pack' landing page, featuring a yellow backpack icon and text about giving students free access to developer tools. The bottom screenshot shows the 'Emails' settings page, where a new email address 'ds016@ie.cuhk.edu.hk' has been added to the list.

Student Developer Pack  
The best developer tools, free for students

Learn to ship software like a pro

Get your pack

Emails

Add email address

ds016@ie.cuhk.edu.hk

Add

# Create your private repository

- ▶ Repository name : IERG4210\_2017
- ▶ Choose Private (to prevent copying among students)
- ▶ Initialization with README
  - ▶ Write your name and student ID in README.md

The screenshot shows a GitHub repository page for 'IERG4210\_2017'. At the top left, there's a dropdown for 'Branch: master'. Below it, the repository name 'IERG4210\_2017 / README.md' is displayed. Underneath, there's a commit history entry by 'DongShuaike' with the message 'Update README.md'. It shows '2 contributors' and two small profile icons. At the bottom, it indicates '2 lines (1 sloc) | 24 Bytes'.

### Create a new repository

A repository contains all the files for your project, including the revision history.

---

Owner  / Repository name

Great repository names are short and memorable. Need inspiration? How about [congenial-invention](#).

Description (optional)

---

Public  
Anyone can see this repository. You choose who can commit.

Private  
You choose who can see and commit to this repository.

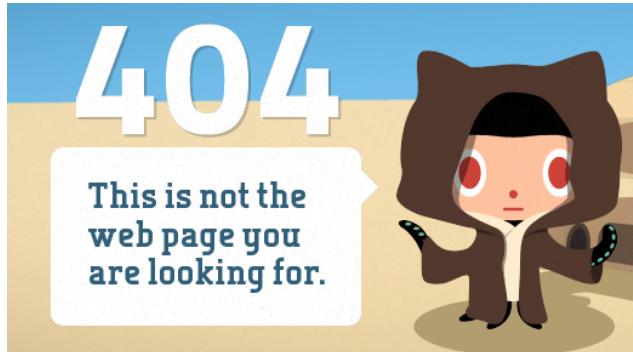
---

Initialize this repository with a README  
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore:  Add a license:

# Invite TA as your Collaborators

- ▶ Enter your private repository and click on Settings→Collaborators
- ▶ Add TA's usernames as collaborators
  - ▶ *DongShuaike*
  - ▶ *xfhie*
- ▶ Now we can access your private repository.
  - ▶ Open an Incognito page and access your repository.
    - ▶ 404... so it's completely private.

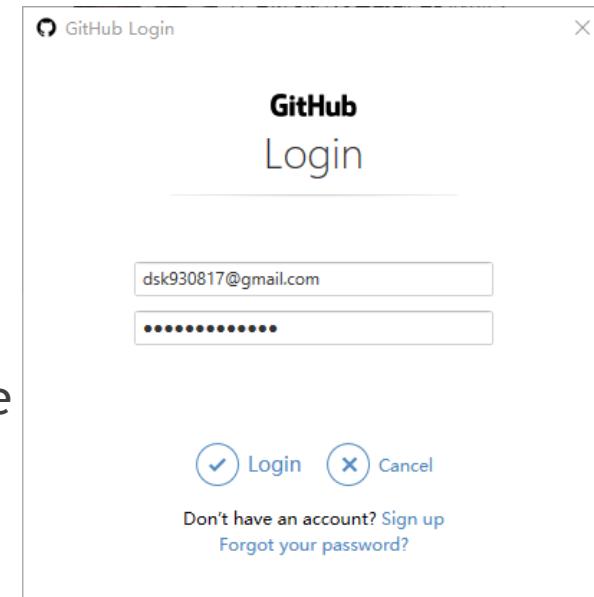


The screenshot shows a GitHub repository interface. At the top, the repository name is "DongShuaike / IERG4210\_2017" and it is marked as "Private". Below the repository name, there are tabs for "Code", "Issues 0", "Pull requests 0", "Projects 0", "Wiki", "Settings", and "Insights". The "Settings" tab is highlighted with a yellow box. A tooltip "Push access to the repository" is shown above the "Settings" tab. The main content area is titled "Collaborators". It lists two users: "Awaiting dskprivate's response" and "Awaiting nextnumbers's response". Each entry has a "Copy invite link" button and a "Cancel invite" button. Below the list is a search bar with placeholder text "Search by username, full name or email address" and a note: "You'll only be able to find a GitHub user by their email address if they've chosen to list it publicly. Otherwise, use their username instead." At the bottom right of the search area is a "Add collaborator" button.

# Try it

- ▶ Clone your private repository to your machine.
  - ▶ `git clone`  
[https://github.com/username/IERG4210\\_2017.git](https://github.com/username/IERG4210_2017.git)
  - ▶ A login form will appear. (Only for private repository) (for linux/mac, you need to input your username and password in shell)
  - ▶ After successful verification, the repository will be copied to the current path.
- ▶ Create a file named *hello\_world.php*

```
<?php  
echo "Hello world!";  
?>
```



# Try it

- ▶ Commit newly-added file to your repository
  - ▶ git add hello\_world.php
  - ▶ git commit -m “Add world world script by php”
- ▶ Push your modification
  - ▶ git push
- ▶ Now check your repository.
  - ▶ Php file has been added.

```
$ git commit -m "Add world world script by php"
[master 7577e84] Add world world script by php
 1 file changed, 3 insertions(+)
 create mode 100644 hello_world.php
```

```
dsk@DESKTOP-VLA4SUI MINGW64 /d/Github/IERG4210_2017 (master)
$ git push
Counting objects: 3, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 335 bytes | 335.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/DongShuaike/IERG4210_2017.git
 f694a03..7577e84 master -> master
```

The screenshot shows a GitHub repository interface. At the top, there are buttons for 'Branch: master' (with a dropdown arrow), 'New pull request', 'Create new file', 'Upload files', 'Find file', and a green 'Clone or download' button. Below this is a commit history table:

| File                                      | Commit Message                | Time                                 |
|---|-------------------------------|--------------------------------------|
| DongShuaike Add world world script by php | Initial commit                | Latest commit 7577e84 11 minutes ago |
| README.md                                 |                               | 6 hours ago                          |
| hello_world.php                           | Add world world script by php | 11 minutes ago                       |

# Branch

- ▶ Your git repository is organized as a tree.
  - ▶ You can have different **branches**.
  - ▶ **In this course, each phase of assignment should be created as a branch.**
- ▶ How to create a branch?
  - ▶ **git checkout master** (Switch to master)
  - ▶ **git checkout -b phase1** (create and copy from master)
  - ▶ echo "Now I have created a new branch phase1" > phase1.md
  - ▶ **(git status)** → show the status of your repo
  - ▶ **git add phase1.md**
  - ▶ **(git status)**
  - ▶ **git commit -m “phase1 committed”**
  - ▶ **git push -u origin phase1**
  - ▶ Now you can find newly added codes on github

```
→ IERG4210_2017 git:(master) git checkout master
Already on 'master'
Your branch is up-to-date with 'origin/master'.
```

```
→ IERG4210_2017 git:(master) git checkout -b phase1
Switched to a new branch 'phase1'
→ IERG4210_2017 git:(phase1) ls
README.md      hello world.php
```

```
→ IERG4210_2017 git:(phase1) x git status
On branch phase1
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    phase1.md
```

```
→ IERG4210_2017 git:(phase1) x git add phase1.md
→ IERG4210_2017 git:(phase1) x git status
On branch phase1
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   phase1.md
```

# Branch

- ▶ In our assignment, each phase should have an independent branch in your repository.
- ▶ Your master branch should always keep the whole project.
  - ▶ It means:
    - ▶ when phase 1 finished, master := phase1
    - ▶ when phase 2 finished, master := phase1 *merge* phase2
    - ▶ when phase 3 finished, master := phase1 *merge* phase2 *merge* phase3
    - ▶ ...
  - ▶ Use **git merge**.
    - ▶ git checkout master
    - ▶ git merge --no-ff phase1
    - ▶ git push -u origin master
    - ▶ Now check your master
- ▶ Every time you add new features to your website, append a brief introduction to your README.md

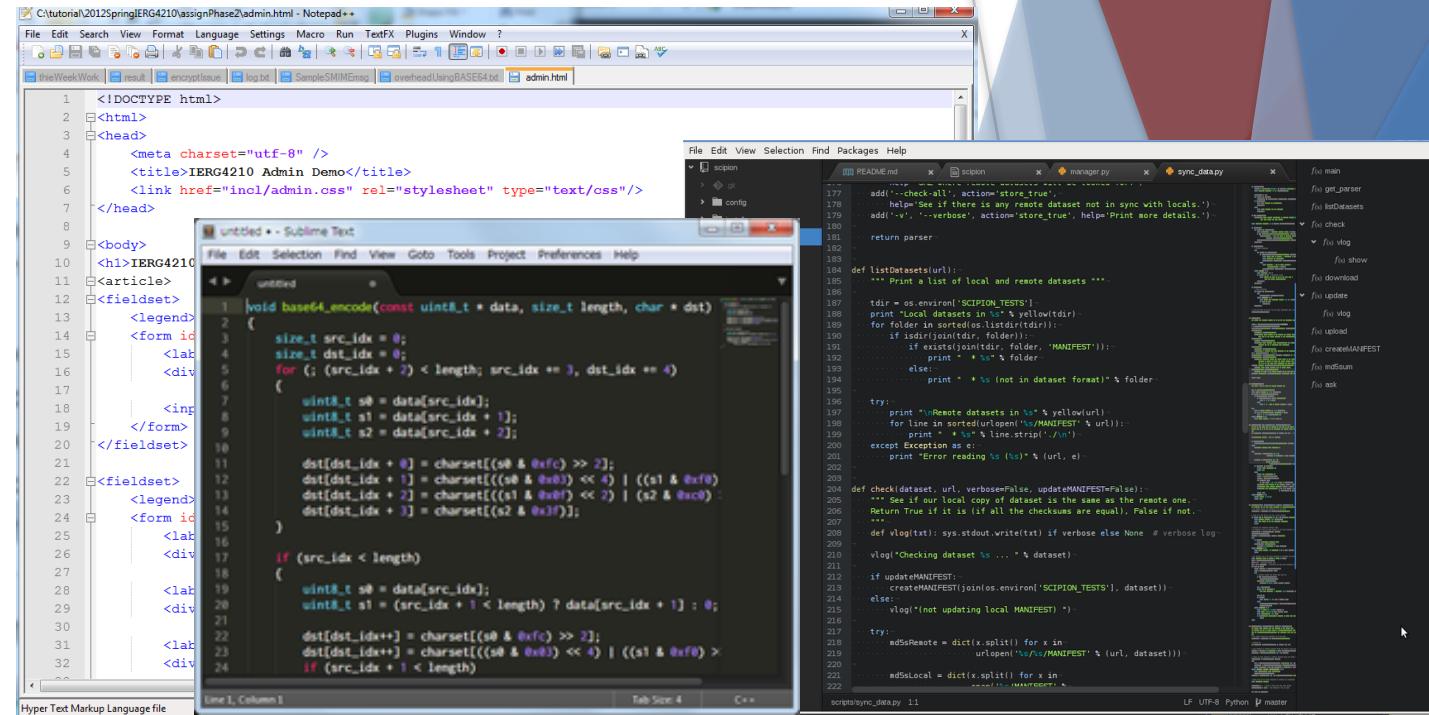
# Some useful tools

There are many tools that help you to write codes and debug

- ▶ Choose your favorite one

Tools on writing codes:

- ▶ notepad, vi
- ▶ Notepad++
- ▶ Sublime Text
- ▶ Atom
- ▶ ...etc



# Some useful tools (IDE)

PhpStorm (<https://www.jetbrains.com/phpstorm/>)

- ▶ PhpStorm = WebStorm + PHP + DB/SQL
- ▶ Works with Xdebug and Zend Debugger
- ▶ **Free student license**

Deployment to AWS

And many more features...

Some tutorials:

[This](#) and [this](#) and [this](#) and [that](#) ...

The screenshot shows the PhpStorm IDE interface. The top navigation bar includes tabs for 'Project', 'WebProcessor.php', 'src...', 'ConsoleHandlerTest.php', and 'MonoLog\Processor\WebProcessor'. The main area displays PHP code for a Symfony application. The code includes imports for Doctrine and a custom Person class, and a foreach loop that outputs group information. To the left is the 'Project' tool window showing the directory structure of the 'symfony2' project. At the bottom are several tool windows: 'Frames' (showing the current stack frame), 'Variables' (listing variables like \$groups, \$name, \$person, etc.), and 'Watches' (listing global variables like \$groups, \$ENV, \$SERVER, \$GLOBALS). The status bar at the bottom right shows the time as 21:24 and the encoding as UTF-8.

# Some useful tools

To check if your website is performed as expected, finding errors, or debugging, we may use:

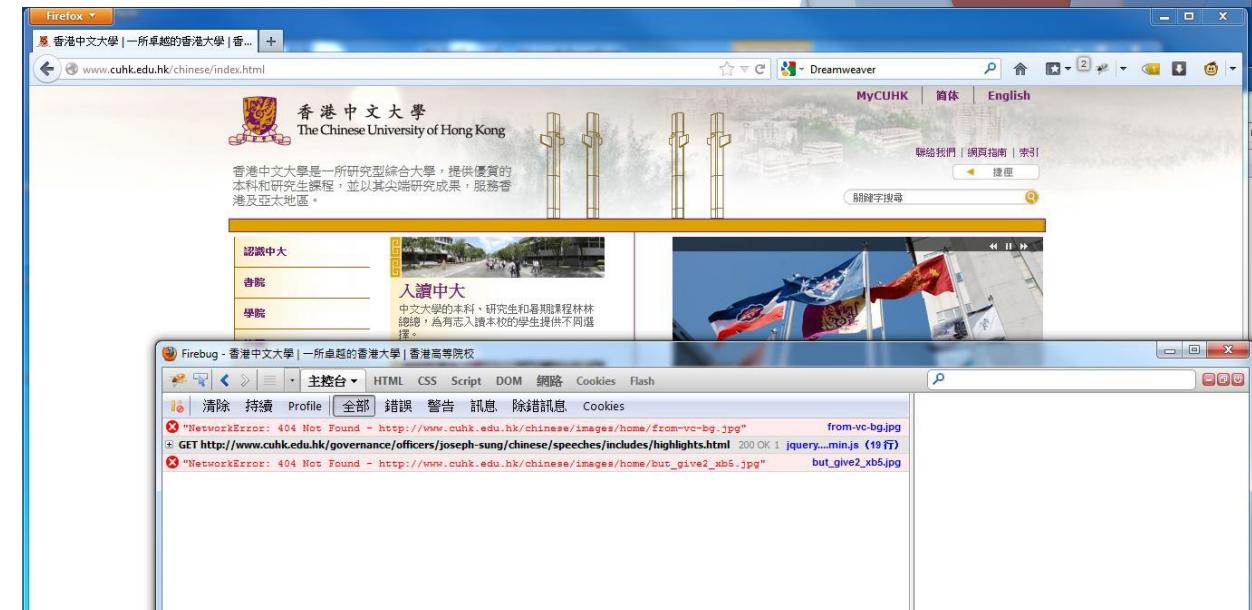
Firefox + Firebug

Chrome Developer Tools (press F12)

Safari Web Inspector

IE/Edge Developer Tools (press F12)

...



# Some useful tools

You can check the source codes, files downloaded, requests and responses made, etc.

It also has many debugging features like setting break points.

You may also use it to modify HTML code, CSS style or try Javascript code

# Some useful tools

JSFiddle (<https://jsfiddle.net/>)

Can show how your Javascript, HTML, CSS codes work and look like in real-time

Can include different libraries (e.g. jQuery)

The screenshot shows the JSFiddle interface with the following details:

- Header:** JSFIDDLE, Run, Update, Fork, TidyUp, JSHint, Share, Login/Sign up.
- Sidebar (Choose Framework):** onLoad, jQuery 1.9.1, Migrate 1.1.0 (unchecked), jQuery UI 1.9.2 (checked), jQuery Mobile 1.3.0b1 (unchecked), jQuery Mobile 1.2.0 (unchecked).  
Script tag attributes: <script> tag attributes, Normalized CSS (checked).
- HTML:** <h1>jQuery Expandable Plugin Example2</h1>  
<p>This example uses default settings. Just start typing in the text area below and watch it expand.</p>  
<textarea name="example"></textarea>
- CSS:** body { font: 15px "Lucida Grande", "Lucida Sans Unicode", Helvetica, Arial, Verdana, sans-serif; color:red; background: #ffff; }  
h1 { font-size: 30px; }  
textarea { font-size:16px; color: #ff0000; }
- JavaScript:** /\* Copyright (c) 2013 Brandon Aaron (<http://brandonaron.net>)  
 \* Licensed under the MIT License ([LICENSE.txt](#)).  
 \* Version 1.1.3  
 \* Contributions by:  
 \* - Karl Swedberg  
 \* - Pistos  
 \*/  
(function (factory) {  
 if (typeof define === 'function' && define.amd) {  
 // AMD. Register as an anonymous module.  
 define(['jquery'], factory);  
 } else {  
 // Browser globals  
 factory(jQuery);  
 }  
})(jQuery);
- Result:** jQuery Expandable Plugin Example2  
This example uses default settings. Just start typing in the text area below and watch it expand.  
will this respond and it does to changes in CSS and HTML . What will happen as I type to the end of the text area - drat - only scroll bar shows - no automatic expansion as implied

# Some useful websites

Mozilla Developer Network (<https://developer.mozilla.org/>)

Stack overflow

Course's Facebook group

**Google** is your friend. Justifies the answer you find before following

Note: Do not ask instructor/tutors for help at late night before trying to solve it yourself

# Q&A

Any questions?

Try the tools yourself

Work hard in this course