

HTML CSS GIT GITHUB



CSCI 4140 Tutorial 1

Jan. 21

Qin Chuan

Table of Content

1. HTML basics
2. CSS
3. Git Introduction
4. Github

HTML Basics

- Browser retrieve HTML files (and related files) from web server
- Browser then render HTML code into a page
 - Show text with styles
 - Show images
 - Collect user inputs

HTML Elements

- HTML file contains HTML elements

Format: `<tag>content</tag>`

- Content can be text, image, HTML elements or mix
- Note that the order is important

Example: `<tag1>hello <tag2>world!</tag2></tag1>`

- Element with empty content

Example: `
`

HTML Attributes

- Additional information to an element
- Name-value pairs: name="value"

```
<a href="index.html">Index Page</a>
```

- Attribute name: href
- Attribute value: index.html

HTML Basics

- Some common HTML tags

HTML tags	Usage
<code><html></code>	Define the root of HTML document
<code><body></code>	Define the start of page body
<code>
</code>	Link break
<code><table></code>	Define a table (use with <code><tr></code> and <code><td></code>)
<code><div></code>	A division / section of page
<code></code>	Show an image (src attribute to define location)

- The tags define page structure
- Class / id attributes and CSS define style

HTML Forms

- Send data (user input) to server
- Server receive and process the data

<code><form></code>	Declare a HTML form
<code><input></code>	Checkbox, textbox... (type attributes)
<code><button></code>	Buttons
<code><textarea></code>	Text input field
<code><select></code>	Drop-down list
<code><option></code>	

- Use name attribute to distinguish and access from script

CSS Basics

- CSS = **Cascading Style Sheets**
- A **style sheet language** used to describe the **presentation** of a document written in HTML or XML
- CSS describes how the structured element must be **rendered** on screen, on paper, in speech, or on other media
- CSS has various **levels** and profiles:
 - CSS1: Published on December 17, 1996
 - CSS2: Published in May 1998
 - CSS2.1: Published on June 7, 2011
 - CSS3: Earliest drafts published in June 1999; published as modules

CSS Example

```
<html>
<head>
  <title>CSS Example: H1</title>
</head>
<body>
  <h1>Hello world!</h1>
</body>
</html>
```



Hello world!

CSS Example

```
<html>
<head>
  <title>CSS Example: H1</title>
  <style>
    h1 {
      color: #FF0000;
      border: 1px dashed #333333;
    }
  </style>
</head>
<body>
  <h1>Hello world!</h1>
</body>
</html>
```



Hello world!

CSS Syntax

```
h1 {  
    color: #FF0000;  
    border: 1px dashed #333333;  
}  
.text {  
    background-color: #AAAAAA;  
}
```

- **Red**: selector
- **Yellow**: declaration
- **Blue**: declaration block

CSS Syntax

```
h1 {  
    color: #FF0000;  
    border: 1px dashed #333333;  
}  
.text{  
    background-color: #AAAAAA;  
    /* You can write comments. */  
}
```

- **Green:** property
- **Orange:** value
- **Purple:** comment

CSS How to insert?

- Inline style (not recommended)

```
<div style="background-color: red; border: 1px solid;">
```

This is ugly...

```
</div>
```

- internal stylesheet (still not recommended)

```
<head>
```

```
  <style> h1 { color: #FF0000; }</style>
```

```
</head>
```

```
<body>
```

```
  <h1>Hello world!</h1>
```

```
</body>
```

CSS How to insert?

- External stylesheet(s)

```
<link href="style.css" rel="stylesheet" type="text/css" />
```

- Write the CSS declaration blocks in a **separate file**
- **Import** the CSS inside the <head> section
- You can import **multiple CSS**
- When the browser renders the page, the CSS styles are applied in the **same order of their appearance**
- This is recommended!

Sidetrack: How to store the code?

- Although there are no standards, most website uses similar structure to store the code (HTML, CSS, JavaScript, ...)
- One of the goals of CSS: **Separation of document content from document presentation**
- If you use inline styles or internal stylesheets and you want to change the color of h1, how many file you need to edit if there are 1000 .html files!?

```
1 css
  2 style.css
  2 print.css
1 js
  2 global.js
  2 ga.js
  2 jquery.js
1 images
  2 tywong.jpg
1 index.html
```

CSS Selector

- To apply CSS on an element, we need to select it with selectors
Element selector

```
div { ... }
```

```
<div> #!@# </div>
```

ID selector ('#' + ID of the element)

```
#sosad { ... }
```

```
<div id="sosad">?</div>
```

Class selector ('.' + class of the elements)

```
.oops { ... }
```

```
<div class="oops">?</div>
```


CSS Selector

- To apply CSS on an element, we need to select it with selectors
Attribute selector

```
a[target] { ... }
```

```
<a target="_blank">A</a>
```

```
a[target="right"] { ... }
```

```
<a target="right">B</a>
```

```
a[target*="ig"] { ... }
```



```
<a name="link">C</a>
```

Read http://www.w3schools.com/css/css_attribute_selectors.asp for more examples!

CSS Selector

- To apply CSS on an element, we need to select it with selectors

Multiple selectors (use “,” – works like “OR”)

```
div, #sosad, .oops { ... }
```

```
<div> #!@# </div>
```

```
<p id="sosad">?</p>
```

```
<i class="oops">?</i>
```

Multiple selectors (works like “AND”)

```
div.oops { ... }
```

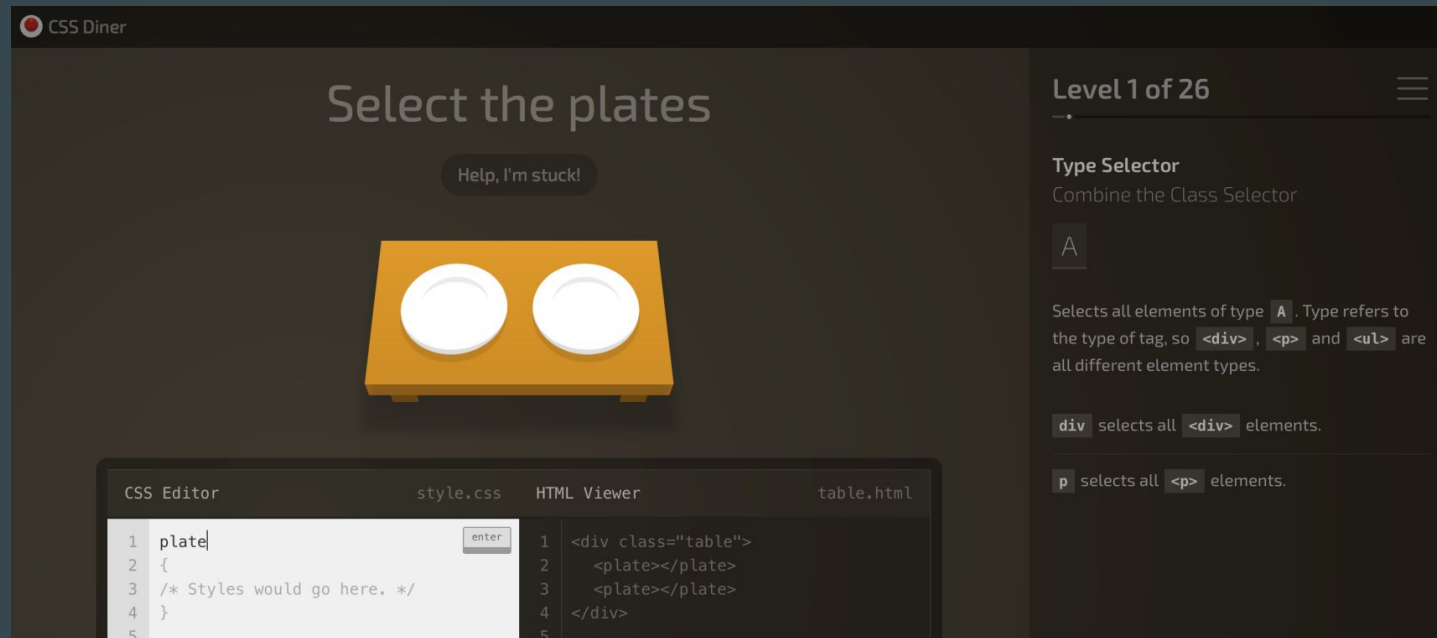
```
<div class="oops">?</div>
```



```
<i class="oops">?</i>
```

CSS Selector

- There are more variations in using CSS selectors
- Recommendation: <http://flukeout.github.io>
- An addictive game for learning CSS selectors!



CCS Properties

- There are too many of them...
- Start from http://www.w3schools.com/css/css_background.asp to learn the available properties yourself
- Yet another good reference (a “man page” of CSS): <https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>
- You can almost customize everything (and forget about IE)
- Demo: Using **Chrome Developer Tools** to modify and try the CSS properties

Git Version control

Managing your code

- Clear picture of development
- No need to worry about modifying wrong code
- History of development
- Compare with / restore old code
- Coordinate with others

Distributed Version Control System

- Everyone having a copy of repository in local
- Communicate (push / pull) with central repository (remote)
- Flexible for project with multiple developers

Git Installation

- Mac

Comes with command line tools

Prompt pop up by typing git in terminal

Or install with Homebrew (brew.sh)

```
brew install git
```

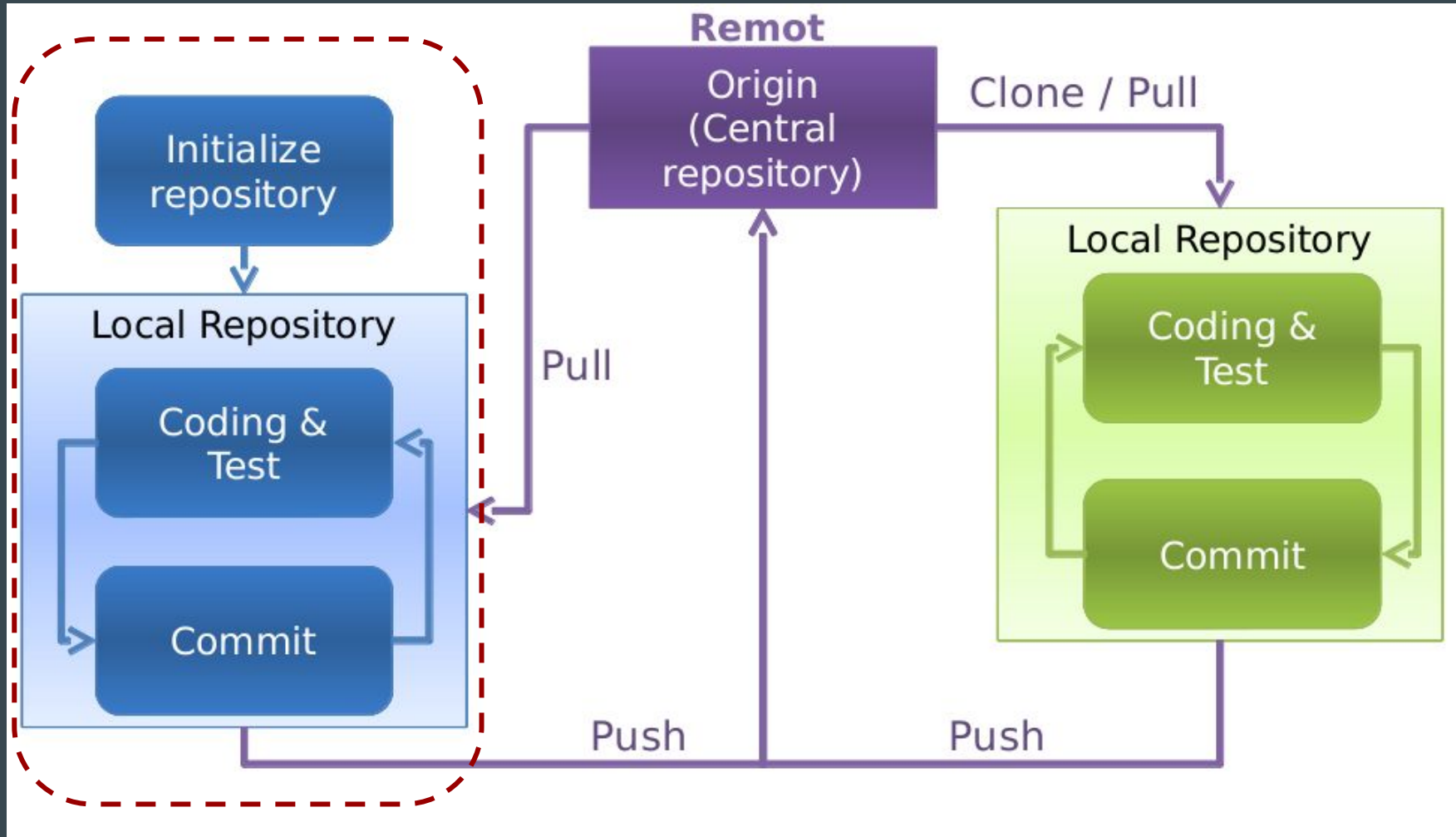
- Ubuntu

```
sudo apt-get install git-core
```

- Windows

Cygwin

Git Workflow



Git commands

- Initial git repository
Git init
- Check status of repository
 - Check modified but unstaged files**Git status**
- Add file to git repository (index)
 - Make modified file become staged (ready to commit)**Git add <files>**
- Make staged change persistent (as commit)
 - Confirm a change / add of files**Git commit [-m <message>]**
- Log of all previous commit
 - With commit hash shown**Git log**

Git File status

After **Git status** (see Demo)

- Unstaged
 - File changed from previous commit, and **not** tracked by git
- Staged
 - File changed from previous commit, which is tracked by git
- Untracked
 - File is **not** tracked by git
- Tracked
 - File **didn't changed** from previous commit, and the file is tracked by git

Git Remote Repository

Synchronize with remote repository

- Clone the remote repository to local (Instead of git init)
`Git clone <url>`
- Or pull the commit from remote to local machine (**Important!**)
`Git pull`
- Work as usual
`Git commit`
- After local edit and commit, push change to remote server
`Git push`

GITHUB


Repository hosting service

- Host your repository
- Manage team works




GitHub (<https://github.com>)

- Free user: All repositories are public
- Encourage social collaboration
- Education plan available: <https://education.github.com/>
- **We will use github for submission this year**

GITHUB Demo



Qin Chuan
chintran27

 The Chinese University of Hong Kong
 NT, HKSAR
 Joined on Nov 9, 2015

1


Follower


4


Starred


2

Following


 Contributions

 Repositories

 Public activity

 Edit profile

Popular repositories

 **CDStore**


Convergent dispersal deduplication datastore

3 ★

Contributions

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan
M													
W													
F													

Summary of pull requests, issues opened, and commits. [Learn how we count contributions.](#)

Less  More

Contributions in the last year

33 total

Jan 17, 2015 – Jan 17, 2016

Longest streak

5 days

January 4 – January 8


Current streak

0 days

Last contributed 2 days ago

Contribution activity

Period: 1 week ▾

 7 commits

Pushed 7 commits to chintran27/WADE Jan 11 – Jan 16

Thank You