

LAB Setup

In this lab, we will be getting our computers set up with Typescript, LESS, and Webstorm. We will cover the “why these?” question throughout the class.

Download Instructions:

1. Download and unzip all the lab files from the following link. Take a look at the structure of the class.
 1. <https://github.com/shawea/bikeSheding/blob/master/downloads/jul.zip?raw=true>
2. Using console/terminal, navigate to the folder you just downloaded and type in:
 1. `python -m SimpleHTTPServer`
3. Open up a Chrome Browser and open the developer tools/console
 1. In Chrome this can be found under View/Developer/JavaScript Console
4. In your terminal install LESS by typing in:
 1. `npm -g install less`
 2. If this does not work, you may need to get superuser access to the account:
 1. Mac: In the terminal type the command: `sudo su`
 2. PC: Click the start button/Accessories/Command Prompt/Run as Administrator
 3. Redo the previous step
5. In your terminal install Typescript by typing in:
 1. `npm -g install typescript`
6. Open WebStorm
7. In Webstorm we want to enable the watchers to compile the CSS and JS. Under File/Settings - 'watcher': delete TypeScript and LESS, then re-add TypeScript and LESS

Test Watchers

1. Using Webstorm, Open Directory of project in: **atut1/aCDN**
2. Open **style.less**, add a space or any css just to 'touch' the file. It should update and compile style.css
3. Open **app.ts**, add space or any .js just to 'touch' the file. It should update and compile app.js.
4. It is set up if the 'file watcher' in WebStorm updates the files.

Test Server

1. Now open Chrome in incognito mode, and open JavaScript console (bonus: memorize this short cut)
2. Surf to <http://localhost:8000> and find atut1/ROOT to open index.html
3. In Webstorm edit app.ts line 1:
 1. It says `console.log("ready 0.1")`
 2. Change it to say (“ready 0.11”) or anything you like.
4. Refresh your browser.
 1. The browser console should match what you have updated
5. Note that python module called SimpleHTTPServer is serving whatever folder it's in. You should be in **atut1** folder.

Note: If you cannot get any of this to work, do not worry, the first lab is to set aside some time to get this working, but the more we get done ahead of time, the more time we can spend on useful teaching!

Optional:

Glance through the List of CSS Framework Resources

Optional:

Rename ROOT/old.js to .ts and see if you can get type scripting to kick in.

Optional:

Dreamweaver: open view, add CSS

Optional:

MaxCDN: account and deploy

LAB #1: Learn to Open A View

Lets do some programming! Pay special attention to the bolded functions:

In app.ts TYPE (do not copy / paste) the following code:

```
//LAB #1
viewDir = '../aCDN/views/'

document.querySelector('#load1').addEventListener('click', onclickone)
function onclickone() {
    console.log ("I just got clicked thank you")
    open('view1', 'kontainer', iloaded1)
    console.log("is it loaded?")
}

function iloaded1(){
    console.log ("loaded1")
}

var load2But = document.getElementById('load2')
load2But.addEventListener('click', function() {
    console.log ("Test")
    open('view2', 'kontainer', iloaded2)
})

function iloaded2(){
    console.log ("loaded2")
}
```

Now run your app and see how the Open command works to fill the “kontainer” div with your the views.