Midterm Topics

Agenda

- Git
- HTML
- CSS
- JS
- Exam Logistics

4 Steps

1. Head to gitlab, copy your repository. (git clone sshLinkToRepo)
   a. Then enter the repo (cd cmsc389N-UID)

2.
```
git remote add upstream
https://gitlab.cs.umd.edu/arasevic/cmsc389Nspring2020-student.git
```

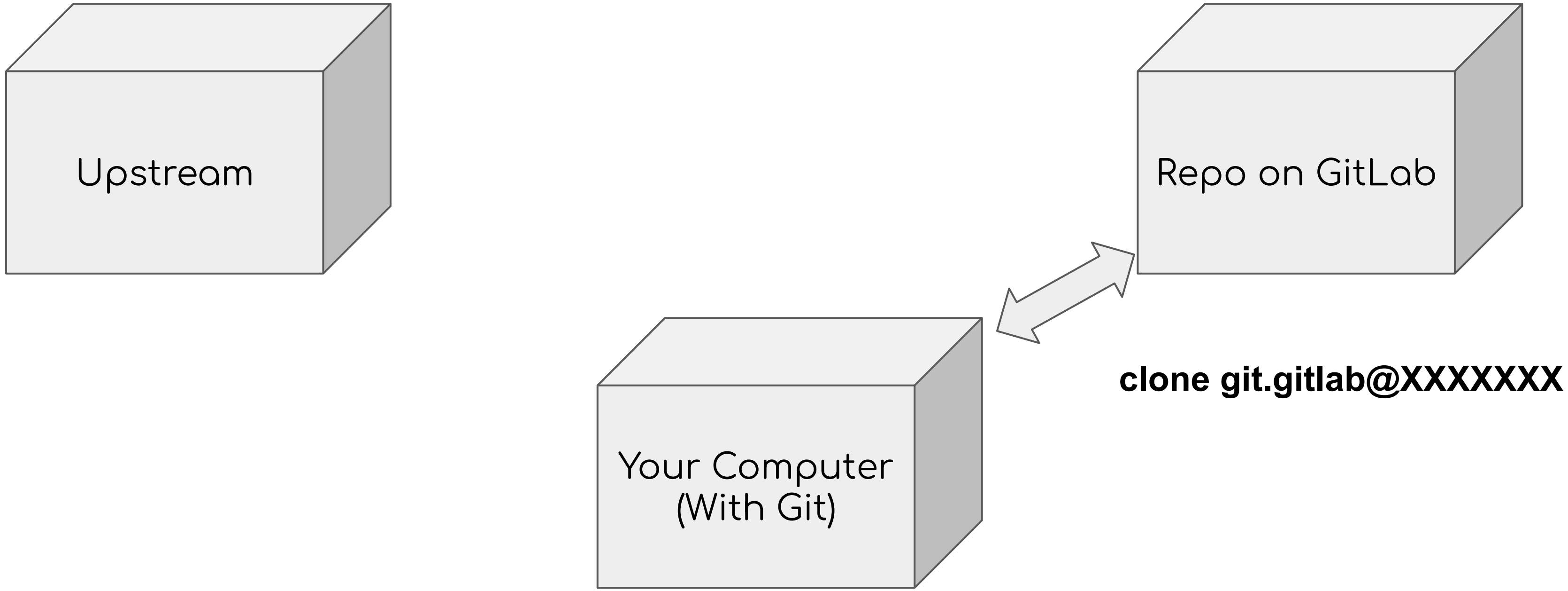
3.
```
git pull upstream master
```

4.
```
git push origin master
```

What are we doing here?

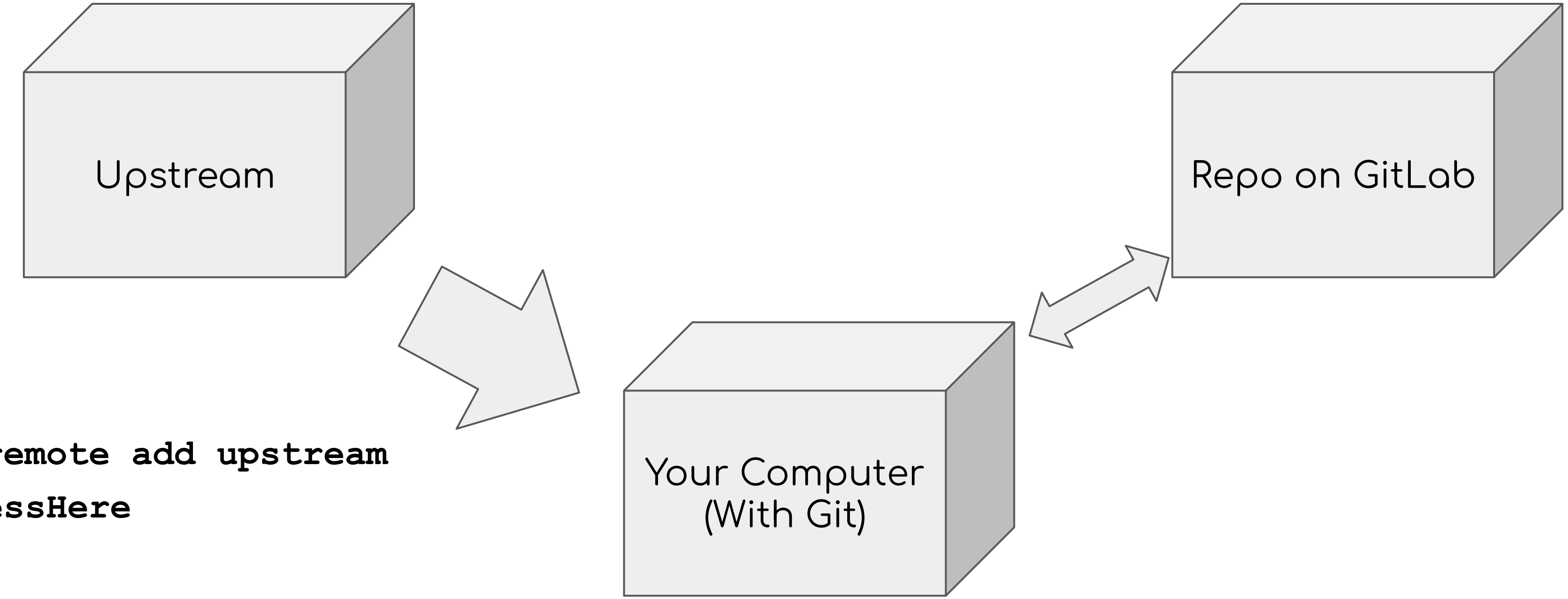
Upstream
(Us instructors)
Repo on GitLab

What are we doing here?



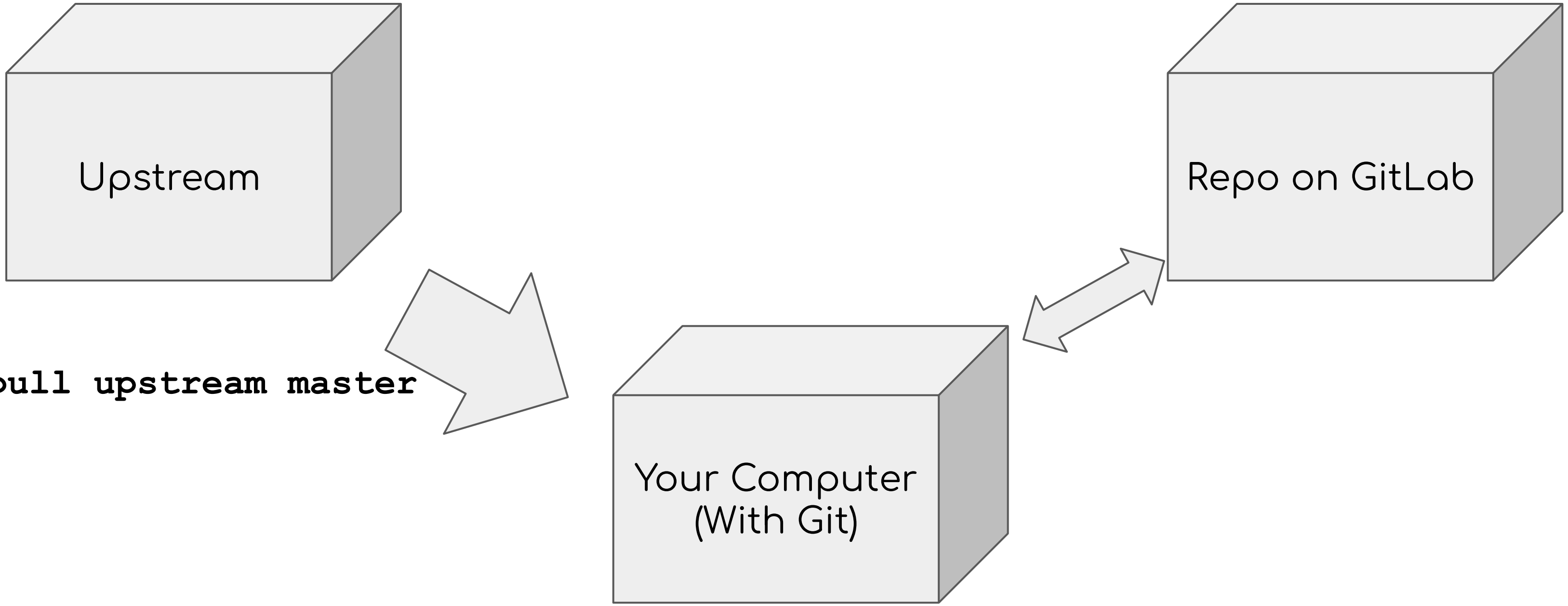
clone git.gitlab@XXXXXXX

What are we doing here?



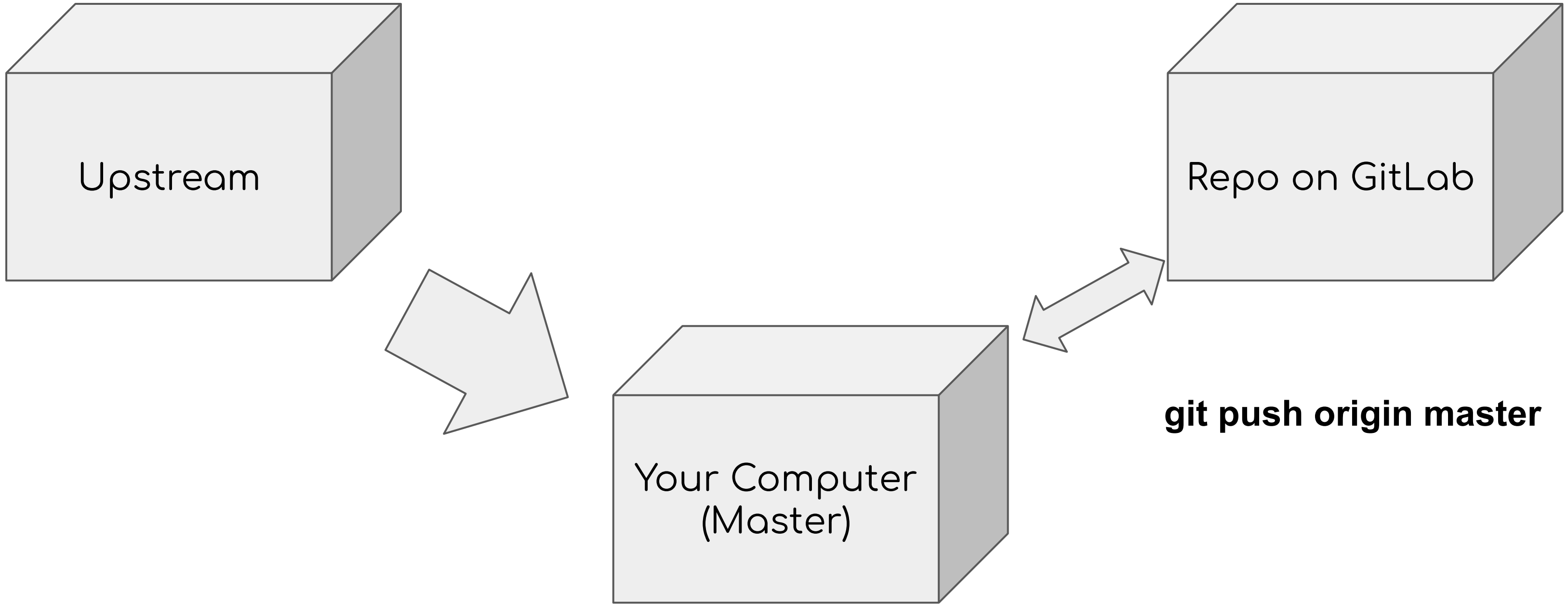
```
git remote add upstream
addressHere
```

What are we doing here?



```
git pull upstream master
```

What are we doing here?



git push origin master

To do!

Modify the README file, push the changes and make sure you can see them in gitlab!

In the README put: Name, date, and the answer to the following question: What is something you're good at that people wouldn't expect?

HTML + The Internet

IP Protocols- IPv4 and IPv6

These protocols define how data is sent between computers over packet-switched network

IPv4:

- 32-bit unsigned integer: 128.8.128.8
- Domain name: cs.umd.edu
- localhost: 127.0.0.1

IPv6:

- 128 bit addresses
- Replaces IPv4
- How many possibilities do we have now?

Basics of the Web: DNS

DNS Domain Name Systems

Protocol for translating domain names to IP addresses

Example: cs.umd.edu → 128.8.128.44

Multiple DNS servers on internet

DNS server may need to query other DNS servers

edu DNS server queries umd.edu server to find cs.umd.edu

Basics of the Web: URL Structure

A URL consists of:

- A Protocol
    - Http
    - Ftp
    - Https
    - File
    - …
- IP Address
- Port (usually left empty)
- Path

Important Tags

- <!DOCTYPE html>
- <html>
- <!-- Other tags go here, also this is a comment -->
- Lists (both types)
- Head Tag
- Table
- Image
- div vs span
- Which are block? Which are inline?

Validating

What is a Validator?

Why do we Validate?

CSS

CSS (Cascading Style Sheets)

- HTML is for controlling structure

- CSS is for controlling presentation

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

Place the above link in the <head> tag to link the two!

CSS Reasoning

- Text file with rules. It includes no html
- Style sheets files use a .css extension
- Allows you to apply typographic styles (font size, line spacing, etc.)
- Allows you to apply spacing instructions
- Allows you to have page layout control
- Smaller html files by avoiding redundancy in style specification
- Easy update a collection of pages by updating only a single file

CSS

Rule:

- Basic element of a style sheet
- Describes the formatting associated with a page element

Rule format:

Selector {declaration(s)}

- Selector: identifies what should be styled in a web document
- Declaration: describes styling information (what and how that portion of the web document should be modified)

Example

```
h1 {
    color:orange;
    text-align: left;
}
```

Types of Style Sheets

- Inline
  - Style information applied to specific tag (e.g., <p style=...”)
  - Avoid if possible (I still do it sometimes)
- Internal
  - Using the <style> tag in the header of the html document
  - Convenient to provide own style to a specific page
- External
  - External style sheet which web pages link to (see <link> tag)
  - Preferred approach

Items You Must Know How to Style

- Color
- Size
- Alignment
-

DOM

What is the DOM?

Be able to draw the Document Tree of an HTML document.

Kinds of Selectors

- Class Selectors:
  - Allow us to apply the same rules to a set of elements
  - Use when you need to apply a style many times in your document
  - Created with a period (also known as full stop)

- ID Selectors:
  - Similar to class selectors but appear only once in the document
  - Used when you need to apply a style only once in your document
  - Created using #

Kinds of Selectors

- Descendant selector
  - Override the type, class and id selector styles
  - Typically with two elements where the second is a descendant

- Examples

```
li a {font-size: 2em}

#header h2 {font-weight: normal;}

#content h2 {font-weight: bold;}
```

Kinds of Selectors

- Universal selector
  - Applies to all elements in context
  - Example: * {font-family: arial, Helvetica; }
- Pseudo-elements
  - Allows you to style an item that is not marked by elements
  - Two pseudo-elements :first-letter, and :first-line

Child Selectors

- A child selector matches when an element is the child of some element. A child selector is made up of two or more selectors separated by ">".
- Example

```
body > p { line-height: 1.3; }
```

sets the style of all p elements that are children of body:

```
div ol > li p { color: tan;}
```

What does this do?

Adjacent Sibling Selectors

- The selector matches if E1 and E2 share the same parent in the document tree and E1 immediately precedes E2, ignoring non-element nodes (such as text nodes and comments)
- Syntax: E1 + E2, where E2 is the subject of the selector
- Example

```
math + p { text-indent: 0 }

h1 + h2 { margin-top: -5mm }
```

JavaScript!

JavaScript

- Finally some programming!

- JavaScript is a programming language that allows us to:
  - Create interactive web pages
  - Control a browser application
    - Open and create browser windows
    - Download and display contents
  - Interact with the user
  - Interact with HTML Forms

JS and ECMAScript

- JavaScript implements ECMAScript

What is ECMAScript?

- A scripting language standard
- ActionScript and JScript are other implementations

Event Handling

- Relies on a single-threaded execution model
- An event queue keeps track of events that have taken place, but have not been processed (event-handler function for the event has not been called)
- All generated events (whether are user-generated or not) are placed in the event queue in the order they were detected by the browser
  - The browser mechanism that detects events and that adds them to the event queue is separate from the thread that is handling the events

How do we run JavaScript?

- Chrome (or any browser)
  - Right click -> inspect

- Node
  - Make sure you have it installed

- Within HTML

Event Handling

- Relies on a single-threaded execution model
- An event queue keeps track of events that have taken place, but have not been processed (event-handler function for the event has not been called)
- All generated events (whether are user-generated or not) are placed in the event queue in the order they were detected by the browser
    - The browser mechanism that detects events and that adds them to the event queue is separate from the thread that is handling the events

How do we run JavaScript?

- Chrome (or any browser)
  - Right click -> inspect
- Node
  - Make sure you have it installed
- Within HTML

Functions

- Functions are Objects
- Name of a function is a reference value

Classic way to create a function:

```
function name (params){
    statements
}
```

Logistical Items:

- Functions are invoked by using the () operator
- Don’t use var for parameters (e.g. function print(x, y))
- Parameters are passed by value
- There is no mandatory main function
- Returning values via *return*

How can I create a function?

1. With a function declaration

2. Function Expression

3. Using a function constructor

Arrow Functions

- Alternative to anonymous functions
    - “Lambda Expressions”
- Rely on the => operator
- Format
    - Parameters => code
    - Parenthesis for parameters are only required if the function has no parameters or 2 or more parameters. Function with one parameter do not require parenthesis surrounding the parameters
    - If code is a single expression no curly braces nor return statement are required

Arrays (One Dimensional)

- Collection of values that can be treated as a unit or individually
  - a special type of objects
  - var a = new Array(4);
- As usual, access elements using []

Arrays can be of any type, and can even contain mixed type elements.

String Methods

- Comparison based on < and >
- concat
  - returns a new string representing concatenation of strings
- includes
  - determines whether one string is found within another
- startsWith
  - whether string begins with characters from another string
- endsWith
  - whether string ends with characters of another string
- indexOf
  - index of first character in string (or -1 if not found)
-

Array Methods

- **fill** - fill elements of an array
- **concat** - returns copy of joined arrays
- **indexOf** - returns position of element in array
- **join** - returns string with all elements in the array
- **pop** - removes & returns last element
- **push** - adds to the end (returns length)
- **reverse** - reverses the array
- **shift** - removes & returns first element
- **unshift** - adds new element to the beginning

Objects

- Property – association between a name and a value
  - When the value is a function the property is referred to as a method
  - Name can be any valid JavaScript string or anything that can be converted to a String (that includes empty string)
    - Any invalid property name can only be accessed using square bracket notation

Object Constructors

- Rather than handwriting all values in an object, Javascript allows for Object Constructors

Ex:

```
function Person(first, last, age, eye) {
  this.firstName = first;
  this.lastName = last;
  this.age = age;
  this.eyeColor = eye;
}
```

Classes in JavaScript

- Use keyword class
- Constructor is no longer using function, use constructor instead
- Methods can be defined with no other keywords necessary
- Not hoisted!

Let’s create an Object!

What is `this`?

- Outside of any object, it refers to the global object window **or** is undefined (if you “use script”)
- Arrow functions have no concept of this.
- When in an object, it refers to the current Object
  - Works the same as in Java
  - This.data to access a data field in your object

Inheritance in JavaScript

- Classes *extend* each other
- References to the superclasses’ methods and constructors must use the *super* keyword
- If the superclass is not created using *class*, you must link the prototypes!

JavaScript DOM Manipulation

Accessing Information:

- document.getElementById(‘myID’);
- document.getElementsByTagName(‘p’);
- document.getElementsByClassName(‘mainMenu”);

Basics of Writing To Document from JavaScript

You may also embed variables into your html now!

For example:

```
let x = “Station Wagons”;

document.writeln(“<p>My favorite cars are ” + x + “</p>”);
```

Most of the examples posted use this, so test it out!

Advanced DOM Manipulation

- element.innerHTML = *new html*
- element.attribute = *new values*
- element.style.property = *new style*
- element.setAttribute(attribute, value);

Exam Details

- Code Writing
- Explaining in your own words
- Multiple choice
- No more than 1 hour 15 mins