CSCI 311 Spring 2019: Lab 3

Learning Objectives:

- use html forms to collect user data
- layout form so that it makes sense to the user
- style form so that it looks good
- use tab index to facilitate keyboard access
- use appropriate form element attributes to constrain user input

What to hand in:

submit the following files to VIU Learn no later than February 2, 23:59:

- self-assessment form
- Zip together:
    - Lab3.html
    - Style.html
    - index.html
    - all css files used
- provide a working link to your index.html (wwwstu.csci.viu.ca/~username/csci311/Lab3/index.html)
- late submissions will be penalized 20% per day

All work must be individual.
Plagiarized work will result in a mark of 0. Further penalties may apply.

Marking Scheme:

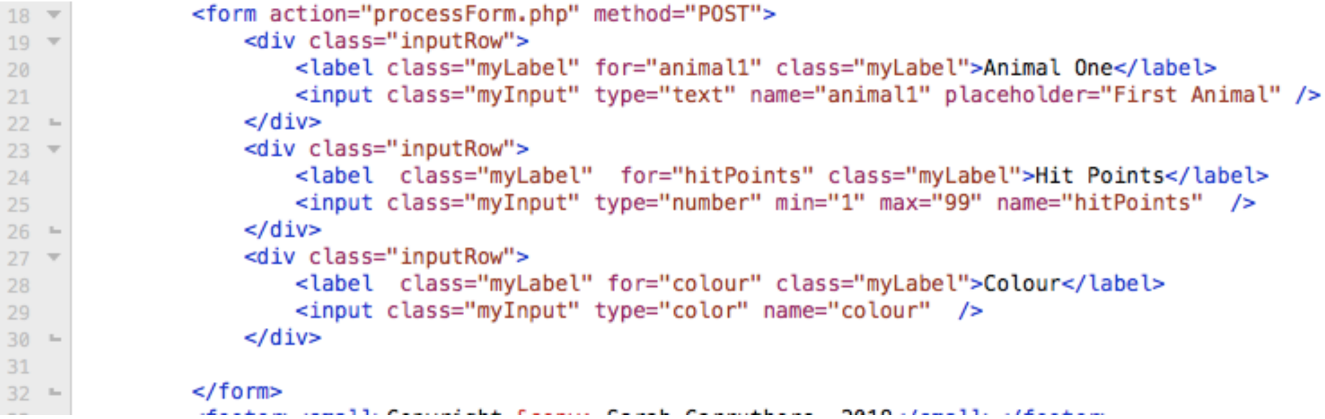
- all files meet given specifications and requirements – 4 marks
- code readability and comments – 1 mark
- self-assessment is consistent with quality of work submitted – 1 mark

Instructions:

For this lab, we will build a basic HTML form that collects and validates user input. The form will be submitted using a basic php file (that you will not modify) called processForm.php. Don't worry (yet!) about what this form is doing, you just need to copy it into your Lab3 directory and set its permissions. We will get there soon!

1. Create the Lab3.html file:
    - You can start with the template you created last week, and update it so that it matches this week's work:
        - change the navigation to include links to Style.html and Lab3.html
    - We will build a form in lab with 5 fields and a submit button. Each field has an associated label.
    - Start by adding a form element to the html as shown in Figure 1. Note:
        - the action attribute is set to "processForm.php". This file needs to be placed in the same directory as your html files.
        - we have 3 inputs here, one text input, one numerical input, and one colour input.
    - Add 3 more input elements:
        - after the first animal input, create a second one with the following attributes:
            - name: animal2
            - type: text
            - class: myInput

        - placeholder: Second Animal
        - ensure its label is correctly created as well
    - after the hitPoints input, create a second one for the Dexterity with the following attributes:
        - name: dexterity
        - type: number
        - min: 1
        - max: 15
        - class: myInput
    - a submit button:
        - name: go
        - type: submit
        - class: myInput
        - value: Create!
        - **NOTE: this is not a button element, it is an input element of type submit!**

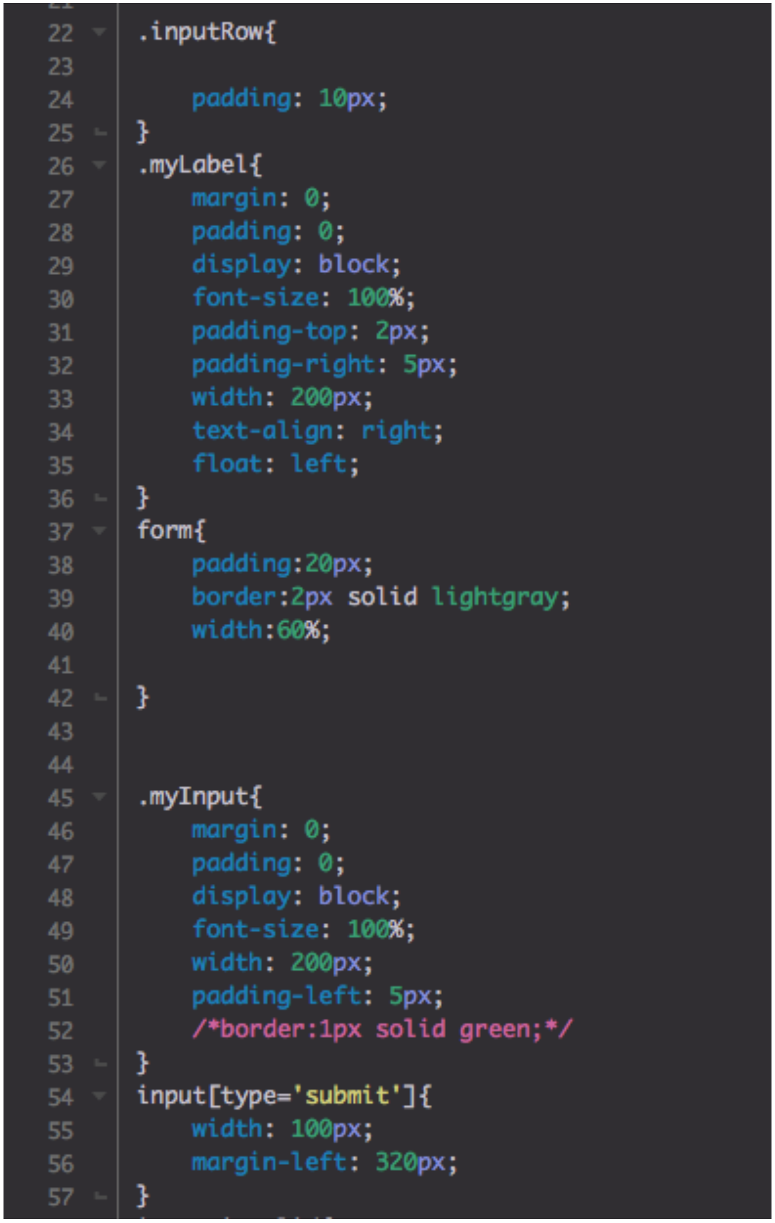


```
        <form action="processForm.php" method="POST">
            <div class="inputRow">
                <label class="myLabel" for="animal1" class="myLabel">Animal One</label>
                <input class="myInput" type="text" name="animal1" placeholder="First Animal" />
            </div>
            <div class="inputRow">
                <label  class="myLabel"  for="hitPoints" class="myLabel">Hit Points</label>
                <input class="myInput" type="number" min="1" max="99" name="hitPoints"  />
            </div>
            <div class="inputRow">
                <label  class="myLabel" for="colour" class="myLabel">Colour</label>
                <input class="myInput" type="color" name="colour"  />
            </div>

        </form>
```

Figure 1: Start of basic form in Lab3.html

2. Style the form. We will create an external style sheet. You can copy yours from last week if you'd like to keep some of the style consistent.
    - We're going to add some basic styling, as follows:
        - add some padding to the inputRow divs
        - change the display property to block
        - Add some padding and margin
        - adjust the alignment so that the labels are right aligned, and the inputs are left aligned



```
.inputRow{

    padding: 10px;
}
.myLabel{
    margin: 0;
    padding: 0;
    display: block;
    font-size: 100%;
    padding-top: 2px;
    padding-right: 5px;
    width: 200px;
    text-align: right;
    float: left;
}
form{
    padding:20px;
    border:2px solid lightgray;
    width:60%;

}


.myInput{
    margin: 0;
    padding: 0;
    display: block;
    font-size: 100%;
    width: 200px;
    padding-left: 5px;
    /*border:1px solid green;*/
}
input[type='submit']{
    width: 100px;
    margin-left: 320px;
}
```

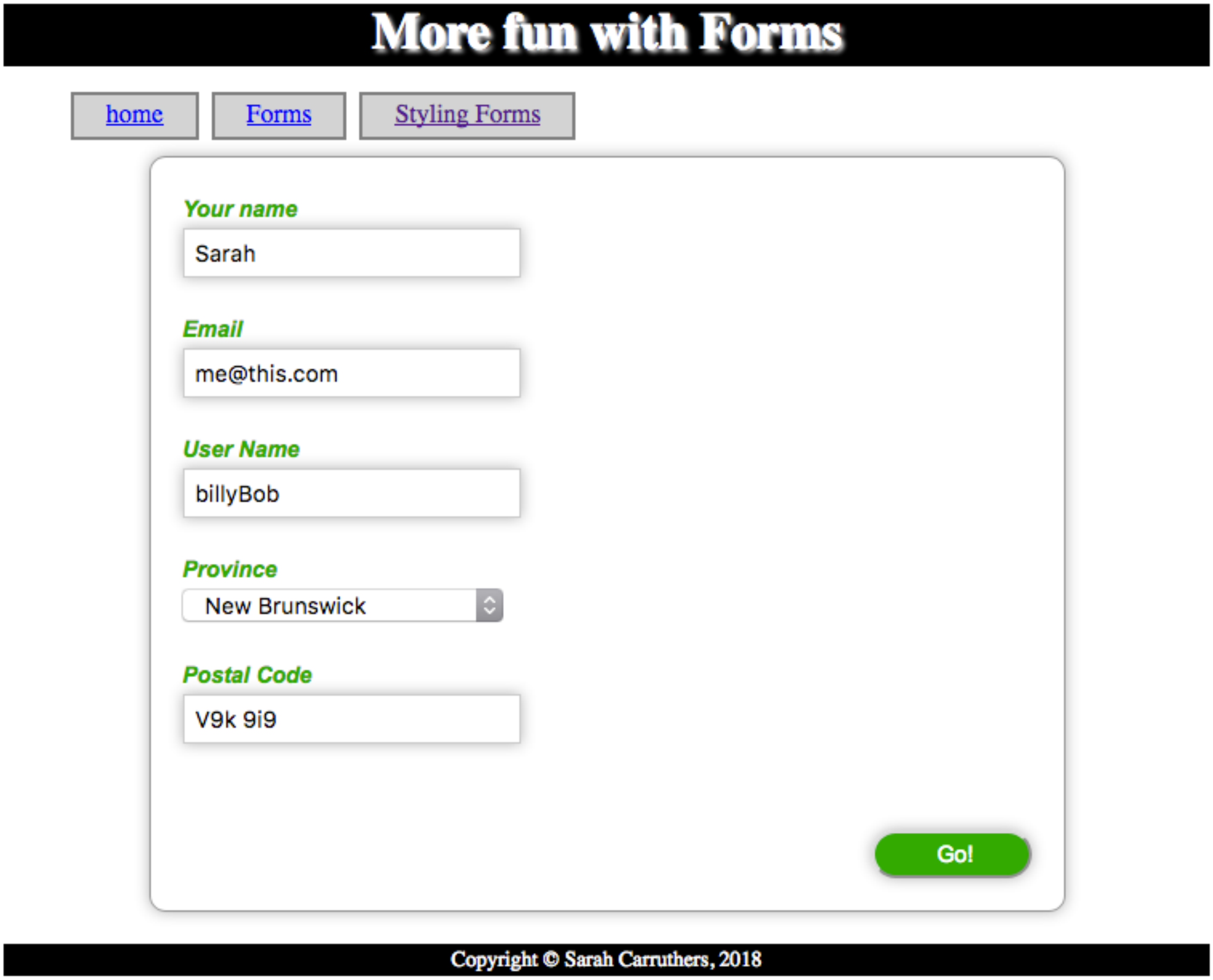
Figure 2: Basic form styling

3. Create the Style.html file:
    - You will work on this on your own. You will create a form that looks like Figure 3. Some important features of this form:
        - **Contents**:
            - you must include the following input elements (<input>) and select elements (<select>):

Input	Type	required?	comments
Your name	text	yes	has pattern
Email	text	yes	has pattern
User name	text	yes	has pattern
Province	select	n/a	include all provinces and territories in alphabetical order, set BC as default
Postal Code	text	yes	has pattern
Submit	submit	n/a	has over effect

- **Data validation.** You will use the pattern attribute in the following elements to validate the input. If the user enters invalid input, you will indicate that by outlining the entry in red. If it is valid, the red outline will disappear. This will be done using CSS and HTML only. (BTW: These patterns are not meant to be realistic. They are an exercise.)
  - Name: consists of 2 or more letters (upper case or lower case). No other characters permitted
  - Email: 1 or more letters, followed by @, followed by 1 or more letters, followed by a single period, then 2 or 3 letters
    - me@this.com is valid
    - a.b@this.asdf is not valid
  - User Name: one or more letters, followed by 0 or more numbers
  - postal code: letter number letter (optional space) number letter number
- **Tab order**. Ensure that the tab order is as follows:
  - 1: Home (link)
  - 2: Name
  - 3: Email
  - 4: UserName
  - 5: Province
  - 6: Postal Code
  - 7: Go!
  - 8: Forms (link)
  - 9: Styling Forms (link)
- **Styling** the form:
  - You may style the form as you please, but please include the following:
    - Ensure width of all inputs is consistent
    - Ensure all inputs are aligned
    - Either:
      - place labels above inputs, or
      - right align labels, and left align inputs
    - Place button on lower right
    - use shadows on some elements
    - use border-radius on some elements
    - have a hover effect on at least the submit button (using only css)

4. Update index.html files in Lab3 folder and CSCI311 folder so that all links are up to date.



More fun with Forms

Your name

Sarah

Email

me@this.com

User Name

billyBob

Province

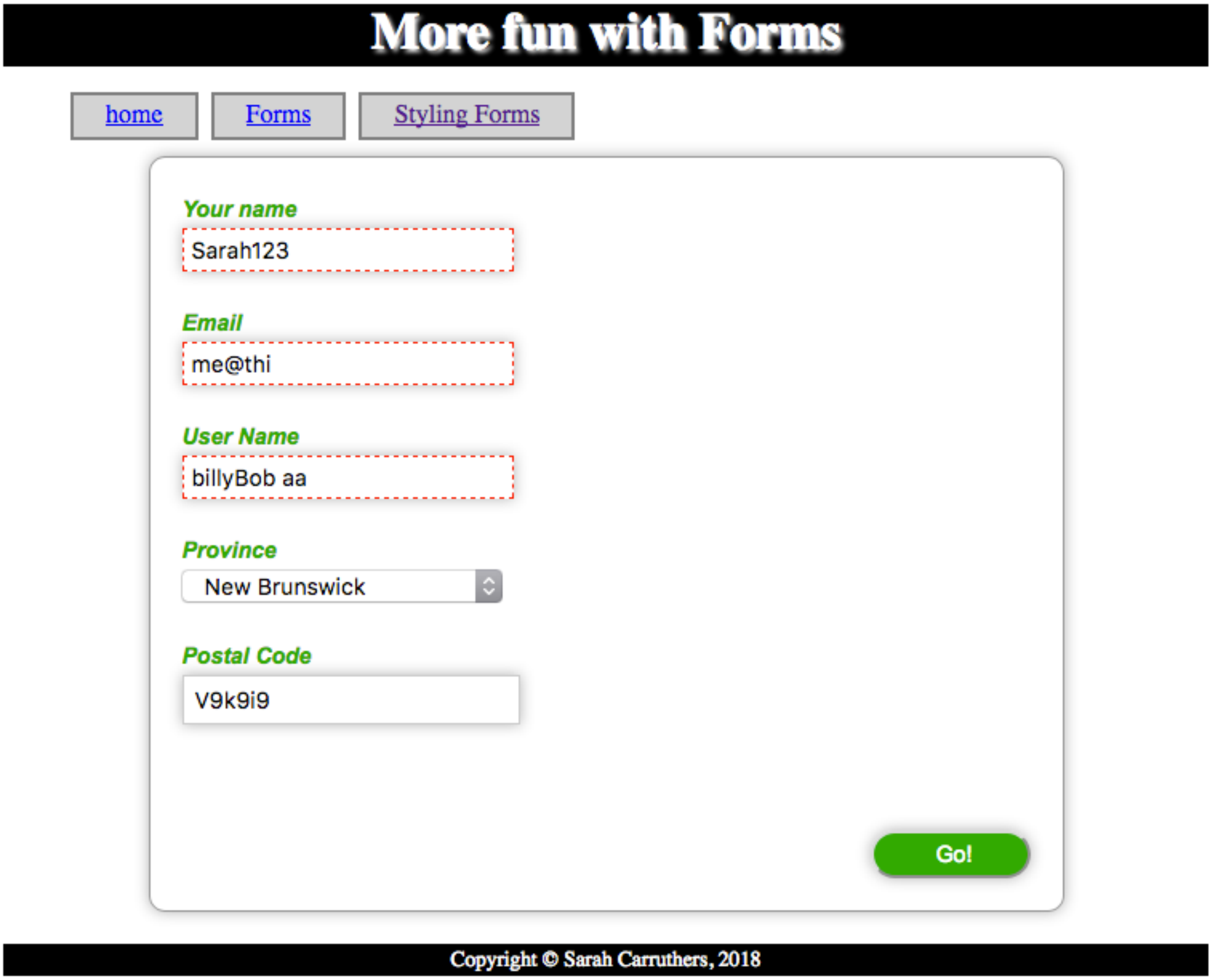
New Brunswick

Postal Code

V9k 9i9

Copyright © Sarah Carruthers, 2018

Figure 3: Style.html form with valid input



More fun with Forms

Your name

Sarah123

Email

me@thi

User Name

billyBob aa

Province

New Brunswick

Postal Code

V9k9i9

Copyright © Sarah Carruthers, 2018

Figure 4: Style.html file with invalid input

Specifications:

- Link to index.html is provided with VIU Learn submission
- index.html file is located in Lab 3 directory, and includes links to all Lab 3 html files
- Navigation is consistent on Lab3.html and Style.html
- Lab3.html file:
  - includes inputs for 2 animals, dexterity (as a number), hit points (as a number), and a colour
  - correctly submits data to processForm.php file without any errors
  - meets all stated requirements above
- Style.html file:
  - includes inputs for name, email, username, province, and postal code
  - tab order is correct
  - all validation is done as specified with pattern attribute
  - form is styled as specified

Requirements:

- All html files must be correctly formatted html5 according to: https://validator.w3.org/
- All css is correctly formatted
- All pages must have an appropriate title set

Code Readability, and Comments:

- Consistent use of indentation (see link in resources below)
- All course code-standards adhered to
- Every file includes at minimum the following comments:
    - your name
    - csci id
    - the date
    - links to any files you used online as inspiration or as resources

Self-Assessment:

You will conduct and submit a self-assessment by creating and checking the following checklists:

- All Specifications are met
- All Requirements are met
- All Code Readability, and Comments are met
- Identify, for each of the items above, based on the specifications, requirements, and code readability and comments in this document:
    - ALL that are correctly achieved
    - ALL that are not correctly achieved.
- Your assessment will be marked based on its objective correctness, so be honest!
- If you are unable to achieve one of the specifications, requirements, etc., this is the place to reflect upon why. Reasons may include:
    - ran out of time
    - didn't understand how
    - instructions were unclear (be specific, and constructive)
    - it was not possible (describe what steps you took to determine this)
    - found a better/alternate way to complete it (if you didn't get prior permission, you may lose marks for doing this)

Resources:

- CSS: http://www.w3schools.com/css/
- HTML: http://www.w3schools.com/html/
- HTML Forms: https://developer.mozilla.org/en-US/docs/Learn/HTML/Forms
- HTML Forms: https://www.w3schools.com/html/html_forms.asp
- invalid property: https://developer.mozilla.org/en-US/docs/Web/CSS/:invalid