HTML Tables, Forms, Inputs & Buttons

CISC-2350-R01 | Fall 2017 | Week 4

Ruta Kruliauskaite

Today’s Agenda

- Attendance
- Discuss homework readings
- Review: Github & Github Pages
- Student web review presentations
- HTML Tables
- HTML Forms, Inputs & Buttons
- Homework Assignment

Discuss readings

Review: Github & Github pages

What is Github?

- Versioning program
- GitHub is web portal, Git is what is used through terminal
- We’ll be using web portal to keep track of our assignments
- Everyone will have their own username and “repository” for this class

Instructions

- Go to github.com and sign up
- Create a repository and name it the class code **CISC_2350_R01**
  - be sure to check the box that initialises with a README.md
- Make a folder for week3 -> Create new file -> Type “Week3” and add “/“ and then name a README.md file again
- Within that folder upload last week’s assignment. You can drag your files into the window and they will upload.
- Hit “Commit”
- You’ll be using your GitHub to upload homework for the rest of the class

Github pages

Easy way to host your website on Github without any external web hosting services

First, we need to create a new “branch” that our folder will be on

rutaitp / CISC-2350-R01-2017

Unwatch 1 | Star 1 | Fork 1

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Settings | Insights

Branch: master ▾ CISC-2350-R01-2017 / test /

Create new file | Upload files | Find file | History

rutaitp committed on GitHub Add files via upload — Latest commit 9c15bfa an hour ago

..		
pages	Add files via upload	an hour ago
README.md	Create README.md	an hour ago
index.html	Add files via upload	an hour ago

README.md

rutaitp / CISC-2350-R01-2017

Unwatch 1 | Star 1 | Fork 1

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Settings | Insights

Branch: master ▾ CISC-2350-R01-2017 / test /

Create new file | Upload files | Find file | History

Switch branches/tags

gh-pages

Branches | Tags

gh-pages

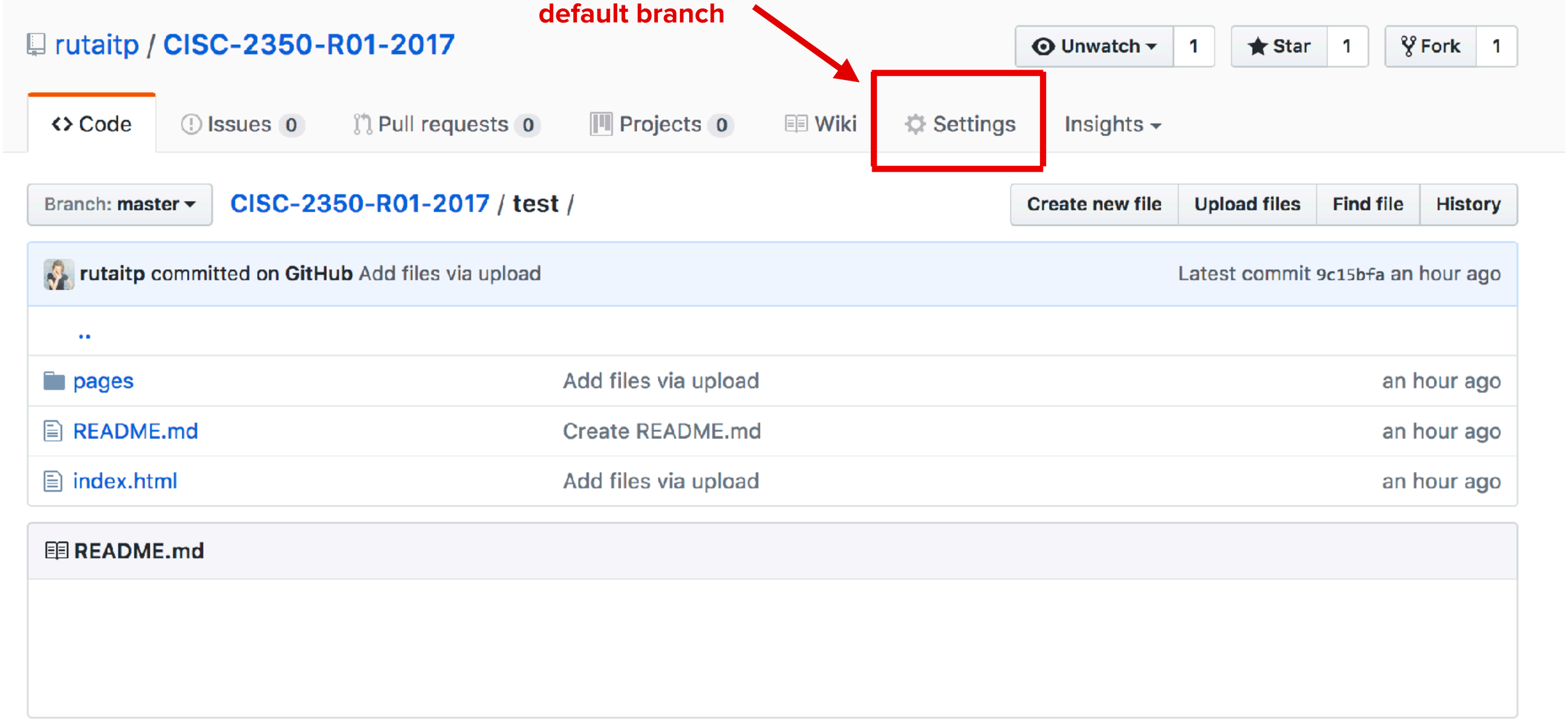
...load — Latest commit 9c15bfa an hour ago

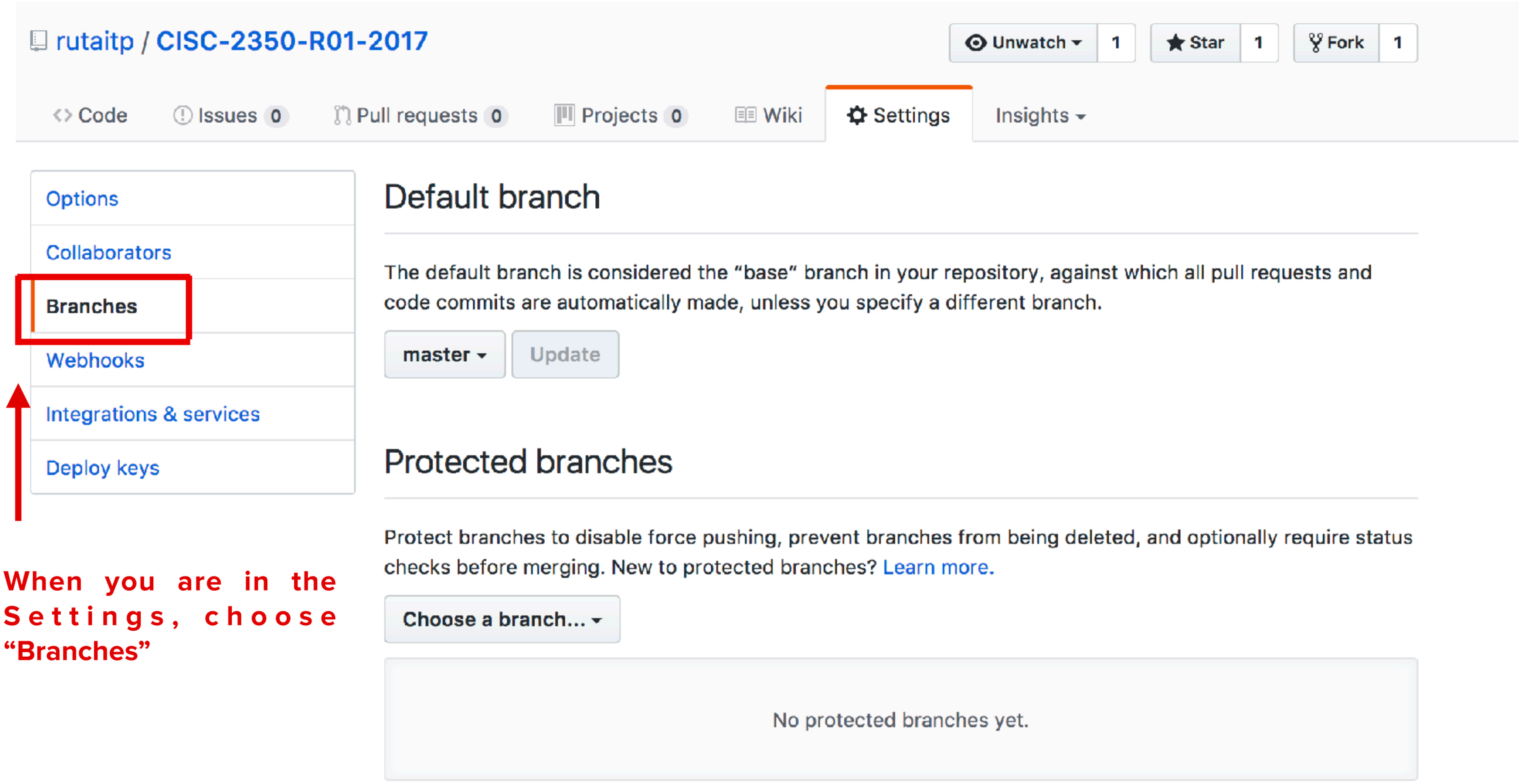
	Add files via upload	an hour ago
	Create README.md	an hour ago
index.html	Add files via upload	an hour ago

README.md

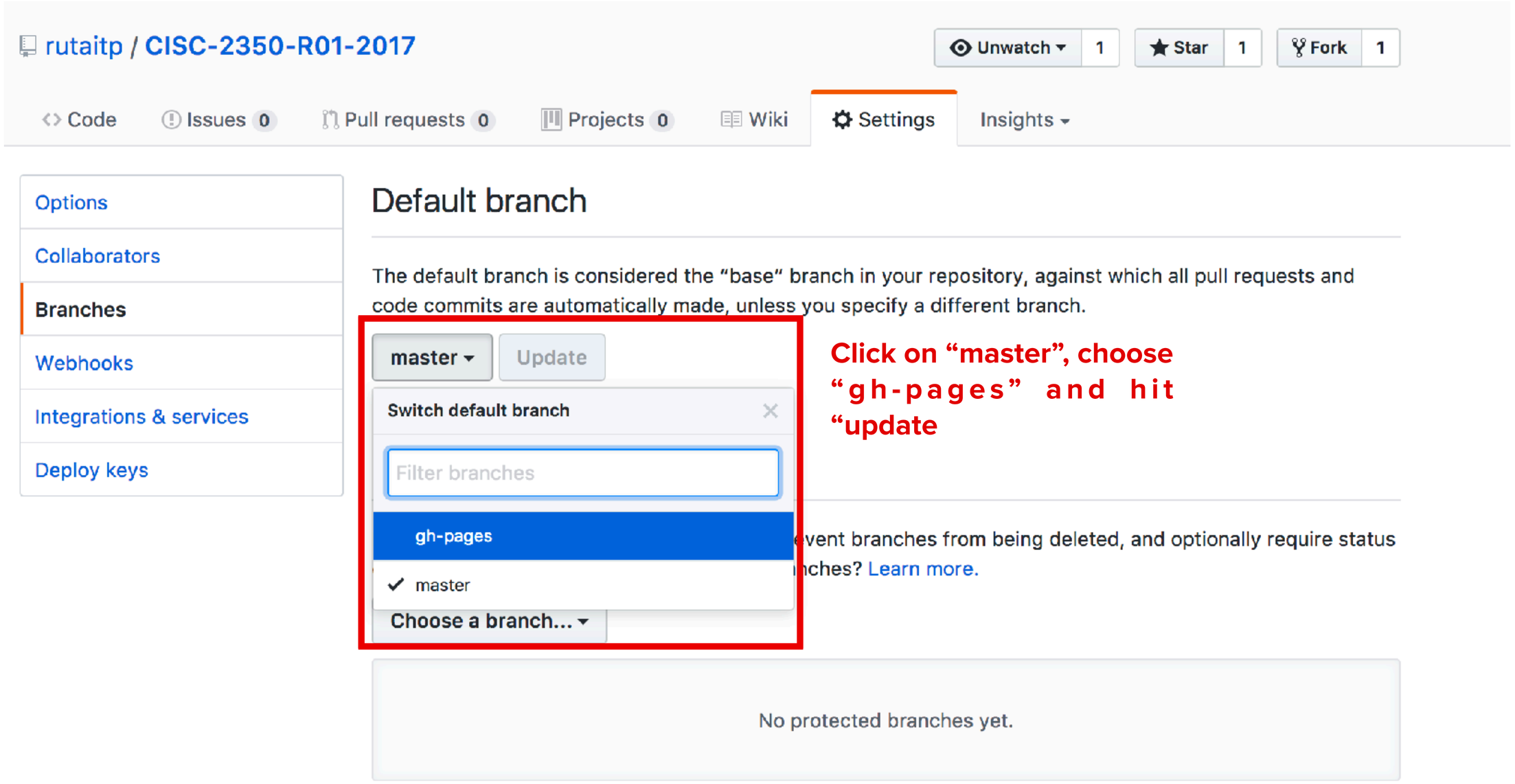
We do so by clicking on “master” and adding new branch named “gh-pages”. That's the default branch name on which Github pages are working. Write “gh-pages” and hit “enter”.

Then we have to go to "settings" to change our default branch





When you are in the Settings, choose "Branches"


rutaitp / CISC-2350-R01-2017
Unwatch 1
Star 1
Fork 1
Code
Issues 0
Pull requests 0
Projects 0
Wiki
Settings
Insights
Options
Collaborators
Branches
Webhooks
Integrations & services
Deploy keys
Default branch
The default branch is considered the "base" branch in your repository, against which all pull requests and code commits are automatically made, unless you specify a different branch.
master
Update
Switch default branch
Filter branches
gh-pages
master
Choose a branch...
Click on "master", choose "gh-pages" and hit "update
vent branches from being deleted, and optionally require status
nches? Learn more.
No protected branches yet.

Default branch changed to gh-pages

rutaitp / CISC-2350-R01-2017

Unwatch 1 | Star 1 | Fork 1

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Settings | Insights

- Options
- Collaborators
- **Branches**
- Webhooks
- Integrations & services
- Deploy keys

Default branch

The default branch is considered the "base" branch in your repository, against which all pull requests and code commits are automatically made, unless you specify a different branch.

gh-pages | Update

It will look like this.
You're almost done!

Protected branches

Protect branches to disable force pushing, prevent branches from being deleted, and optionally require status checks before merging. New to protected branches? Learn more.

Choose a branch...

Now go to

username.github.io/mainfolder/thisweekfolder/

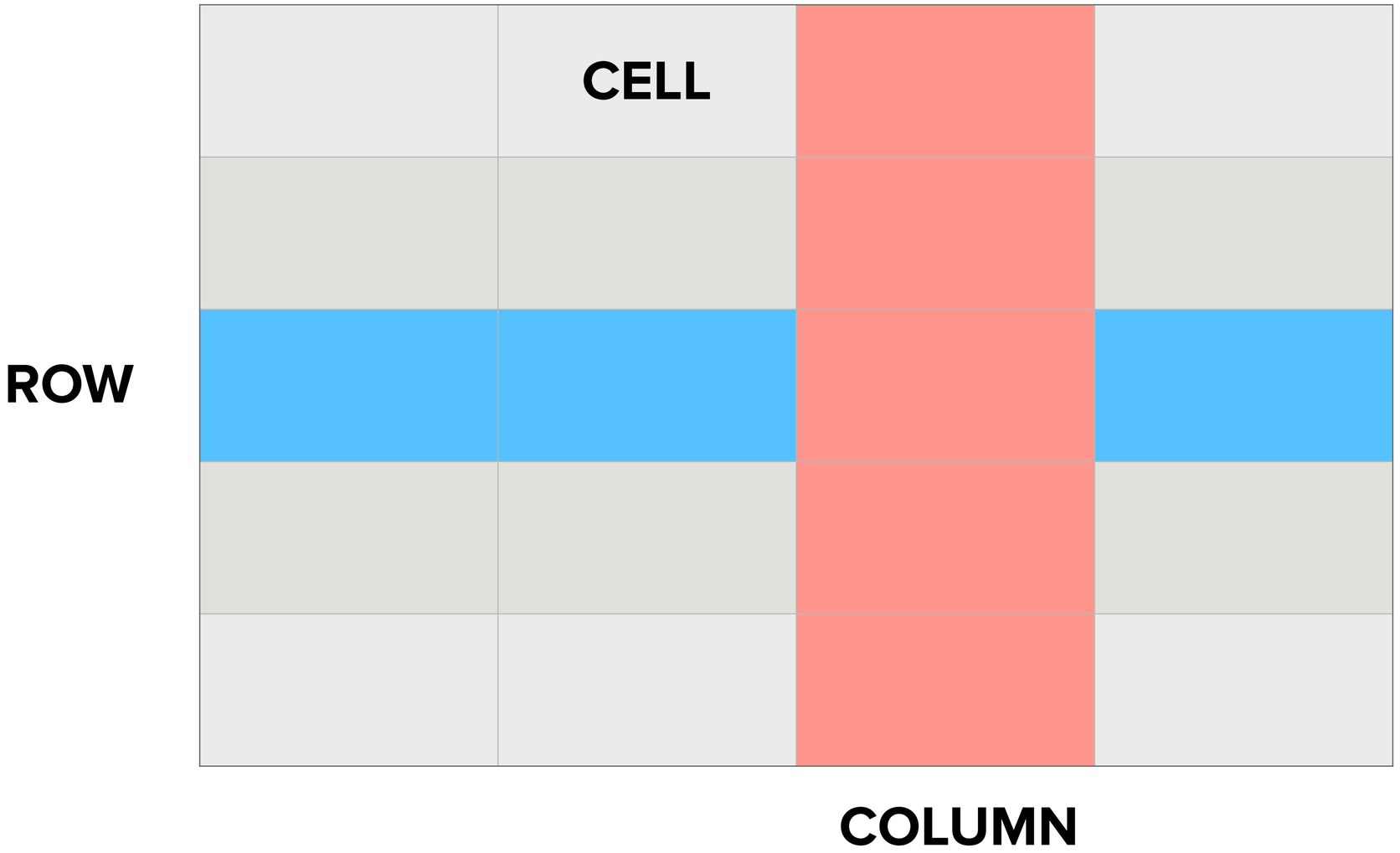
Web review presentations

HTML Tables

Tables

- Tables are good to organise information
- A table represents information in a grid format
- Each block in the grid is referred to as a **table cell**
- In HTML a table is written out row by row

Table structure



Basic table structure

- **<table>** element is used to create a table (written out row by row)
- **<tr>** indicates each row
- **<td>** indicates each cell of a table

index.html

```
<!doctype html>
<html>
    <head>
        <title>Tables</title>
    </head>
    <body>
        <!-- basic table structure -->
        <table>
            <tr>
                <td>1</td>
                <td>2</td>
                <td>10</td>
            </tr>
            <tr>
                <td>3</td>
                <td>4</td>
                <td>11</td>
            </tr>
            <tr>
                <td>5</td>
                <td>6</td>
                <td>12</td>
            </tr>
        </table>

    </body>
</html>
```

Adding table headings

- **<th>** is used to represent the heading for either a column or a row
- Even though there is no content, you should still use it to represent an empty cell
- Add **<scope>** to indicate if it's a heading for row or column

```
<!-- table with headings -->
<table>
    <tr>
        <th scope="col">Day of a week</th>
        <th scope="col">Sports activity</th>
        <th scope="col">Km</th>
    </tr>
    <tr>
        <th scope="row">Monday</th>
        <td>Run</td>
        <td>5</td>
    </tr>
    <tr>
        <th scope="row">Tuesday</th>
        <td>Run</td>
        <td>10</td>
    </tr>
    <tr>
        <th scope="row">Wednesday</th>
        <td>Run</td>
        <td>3</td>
    </tr>
</table>
```

Spanning columns

- Sometimes you may need the entries in a table to stretch across more than one column
- You can add ***colspan*** attribute on <th> or <td> to indicate how many columns that cell should run across

	Morning	Lunch	Afternoon	Evening
Monday	Run	Meeting	Work	Meeting friends
Tuesday	Workout and breakfast		Work	Relax
Wednesday	Day off			

	Morning	Lunch	Afternoon	Evening
Monday	Run	Meeting	Work	Meeting friends
Tuesday	Workout and breakfast		Work	Relax
Wednesday	Day off			

```
<!-- spanning columns -->
<table>
    <tr>
        <th></th>
        <th>Morning</th>
        <th>Lunch</th>
        <th>Afternoon</th>
        <th>Evening</th>
    </tr>
    <tr>
        <th scope="row">Monday</th>
        <td>Run</td>
        <td>Meeting</td>
        <td>Work</td>
        <td>Meeting friends</td>
    </tr>
    <tr>
        <th scope="row">Tuesday</th>
        <td colspan="2">Workout and breakfast</td>
        <td>Work</td>
        <td>Relax</td>
    </tr>
    <tr>
        <th scope="row">Wednesday</th>
        <td colspan="4">Day off</td>
    </tr>
</table>
```

Spanning rows

- Add ***rowspan*** attribute on <th> or <td> to indicate how many columns that cell should run across

	Morning	Lunch	Afternoon	Evening
Monday		Work	Work	Drinks
Tuesday	Run	Work	Relax	Dinner
Wednesday		Time off		Read

	Morning	Lunch	Afternoon	Evening
Monday		Work	Work	Drinks
Tuesday	Run	Work	Relax	Dinner
Wednesday		Time off		Read

```html
<!-- spanning rows -->
<table>
    <tr>
        <th></th>
        <th>Morning</th>
        <th>Lunch</th>
        <th>Afternoon</th>
        <th>Evening</th>
    </tr>
    <tr>
        <th scope="row">Monday</th>
        <td rowspan="3">Run</td>
        <td>Work</td>
        <td>Work</td>
        <td>Drinks</td>
    </tr>
    <tr>
        <th scope="row">Tuesday</th>
        <td>Work</td>
        <td rowspan="2">Relax</td>
        <td>Dinner</td>
    </tr>
    <tr>
        <th scope="row">Wednesday</th>
        <td>Time off</td>
        <td>Read</td>
    </tr>
</table>
```

Long tables

- Sometimes tables contain a lot of rows and columns
- <thead>, <tbody> and <tfoot> help distinguish between the main content of the table and the first and last rows
- **<thead>**: the headings of the table should sit here
- **<tbody>**: the body of the table should sit here
- **<tfoot>**: the footer of the table should sit here
- Browsers don't really treat these elements any different, but it’s helpful for designers working with CSS

```
<!-- long tables -->
<table>
    <!-- headings -->
    <thead>
        <tr>
            <th></th>
            <th></th>
        </tr>
    </thead>

    <!-- body -->
    <tbody>
        <tr>
            <td></td>
            <td></td>
        </tr>
    </tbody>

    <!-- footer -->
    <tfoot>
        <tr>
            <td></td>
            <td></td>
        </tr>
    </tfoot>
</table>
```

HTML Tables Summary

FROM JOHN DUCKET “HTML AND CSS”

- Start with a <table> element
- For each row, use a <tr> (stands for table row) element
- Inside each row, there are number of cells. Use <td> for each cell item, or <th> for a cell item that’s a table header
- There are other ways to customize tables using attributes such as *colspan* or *rowspan*

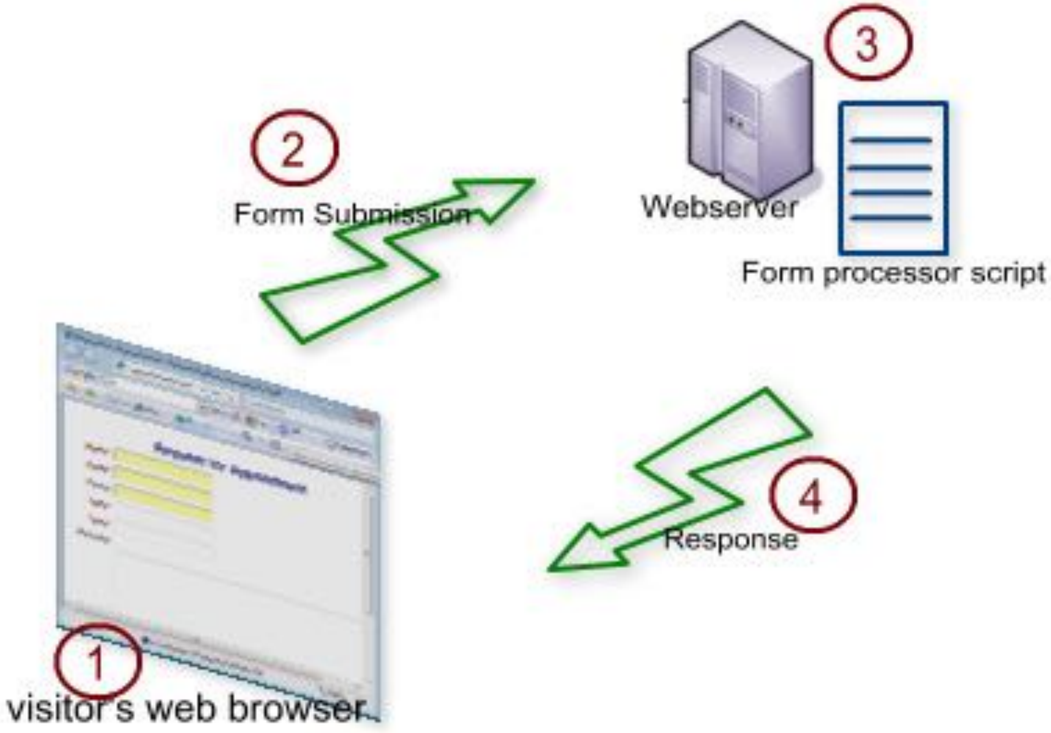
HTML Forms and Inputs

Forms

- Allows to collect information from visitors to your site
- Can include text input, selectors, buttons, upload files, drop-down boxes, and more
- Form examples:
  - search boxes
  - shopping online
  - registering for newsletters, etc.
- These end up being heavily integrated with JavaScript and server-side programming... but not in this class.

How forms work

- User fills in the form and submits information
- The name of the form and value submitted is being sent to the server
- Server processes information
- Server sends back information by creating a new page in the browser



For now we will make the forms on the client side and assume the other side will get taken care of.

ADDING TEXT:

Text input (single-line)
Used for a single line of text such as email addresses and names.

Password input
Like a single line text box but it masks the characters entered.

Text area (multi-line)
For longer areas of text, such as messages and comments.

MAKING CHOICES:

Radio buttons
For use when a user must select one of a number of options.

Checkboxes
When a user can select and unselect one or more options.

Drop-down boxes
When a user must pick one of a number of options from a list.

SUBMITTING FORMS:

Submit buttons
To submit data from your form to another web page.

Image buttons
Similar to submit buttons but they allow you to use an image.

UPLOADING FILES:

File upload
Allows users to upload files (e.g. images) to a website.

FROM JOHN DUCKET “HTML AND CSS”

Form tag

- Usually there would be a **<form> </form>** tag around your forms
- That tag would specify where the information is going (so a server somewhere) and the method for that
- There’s also a value which is the answer that’s filled into forms which is sent to the server as well
- We are still going to use the form tag, but without the other information, it’s a good practice for future life as a web developer

Text input

- Single line text input
- The type is text because it’s text input
- It has a name which would be accessible later, but just be clear with it
- Has a max length option for maximum character

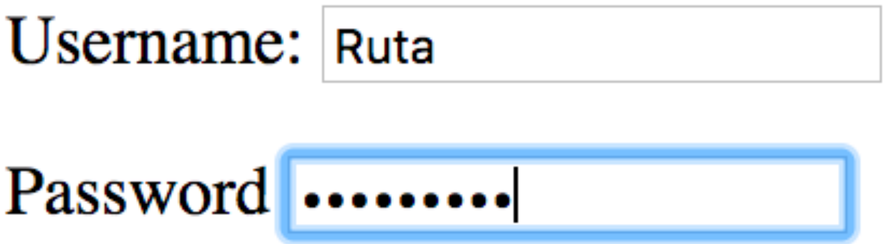
```
<!-- text input -->
<form>
    <!-- here write what information you're asking from the user -->
    <p>Username:
        <input type="text" name="username" maxlength="30" />
    </p>
</form>
```

Password input

- Like the text input but will block out the password

```
<!-- text input -->
<form>
    <!-- here write what information you're asking from the user -->
    <p>Username:
        <input type="text" name="username" maxlength="30" />
    </p>

    <p>Password
        <input type="password" name="password" maxlength="20" />
    </p>
</form>
```

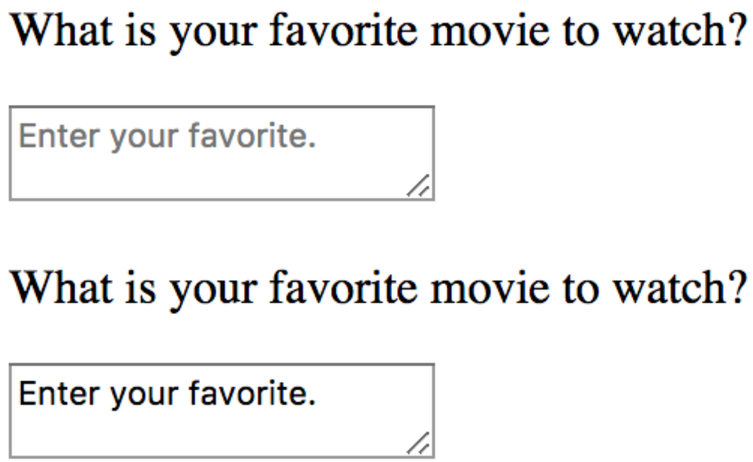


Text area

- For multi-line text input
- Has opening / closing tags <textarea></textarea>
- Text in between tags appears as a placeholder when the page loads
- Text under placeholder attribute disappears when clicked on

```
<!-- with placeholder that will disappear -->
<form>
    <p>What is your favorite movie to watch?</p>
    <textarea name="comments" placeholder="Enter your favorite."></textarea>
</form>

<!-- without placeholder -->
<form>
    <p>What is your favorite movie to watch?</p>
    <textarea name="comments">Enter your favorite.</textarea>
</form>
```



Radio buttons

- The little circle buttons we see everywhere are referred to as radio buttons
- They'll have a value attribute because the possible answers are already written
- Can have a default checked button
- Allow users to pick only one option

```
<!-- radio button -->
<form>
    <p>Select your favorite input type: <br>
        <input type="radio" name="favButton" value="radio" checked="checked"/> Radio
        <input type="radio" name="favButton" value="checkbox" /> Checkbox
        <input type="radio" name="favButton" value="text" /> Text
    </p>
</form>
```

Select your favorite input type:
Radio Checkbox Text

Checkbox

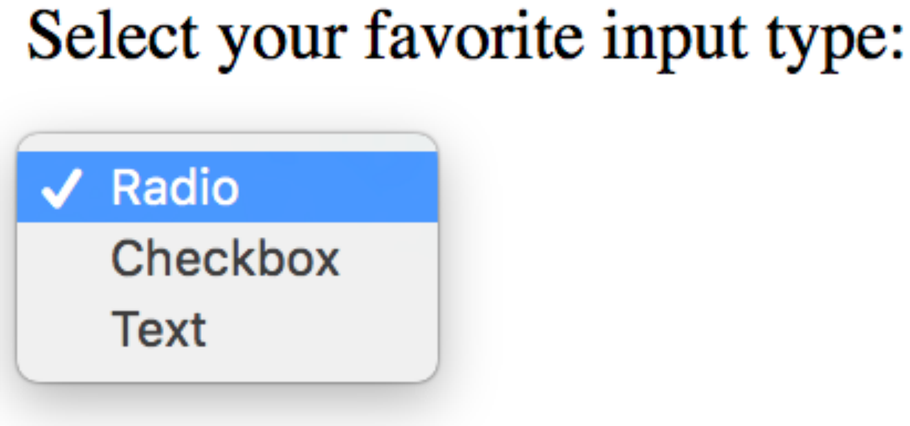
- Check one or more options

```
<!-- checkbox -->
<form>
    <p>Select your favorite input type: <br>
        <input type="checkbox" name="favButton" value="radio" checked="checked"/> Radio
        <input type="checkbox" name="favButton" value="checkbox" /> Checkbox
        <input type="checkbox" name="favButton" value="text" /> Text
    </p>
</form>
```

Drop down list

- Also known as select box
- Allows users to select one option from drop down list
- <select> element is used to create a dropdown list box
- It contains two or more <option> elements

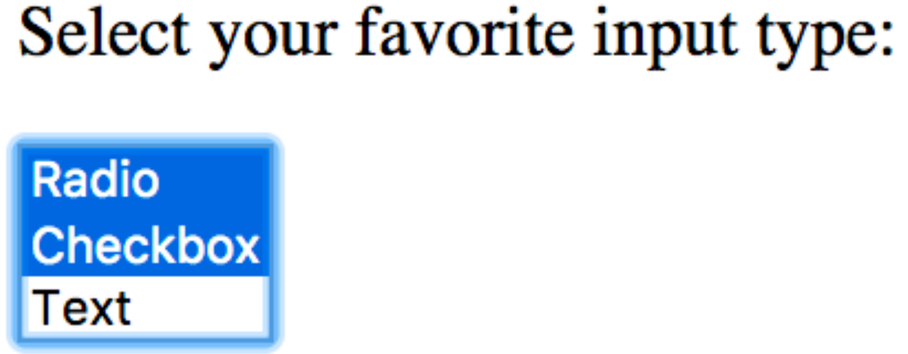
```
<!-- dropdown -->
<form>
    <p>Select your favorite input type:</p>
    <select>
        <option value="radio" selected="selected">Radio</option>
        <option value="checkbox">Checkbox</option>
        <option value="text">Text</option>
    </select>
</form>
```



Multiple select box

- size attribute turns dropdown into a box that shows more than one option
- multiple attribute allows users to select more than one option

```
<!-- multiple from the dropdown -->
<form>
    <p>Select your favorite input type:</p>
    <select multiple="multiple" size="3">
        <option value="radio" selected="selected">Radio</option>
        <option value="checkbox">Checkbox</option>
        <option value="text">Text</option>
    </select>
</form>
```



Submit button

- Probably most common button type to submit user input

```
<!-- submit button -->
<form>
    <p>Are you ready to make that selection?</p>
    <input type="submit" name="submit" value="SUBMIT" />
</form>
```

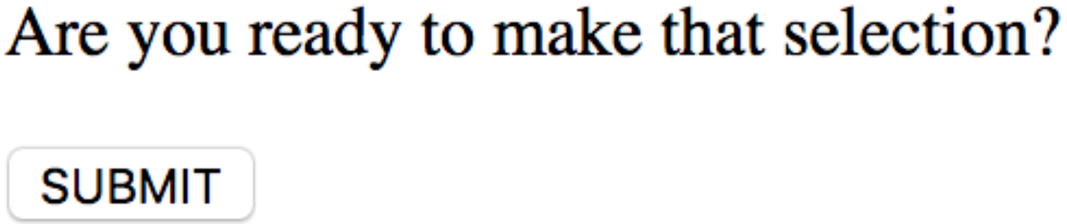
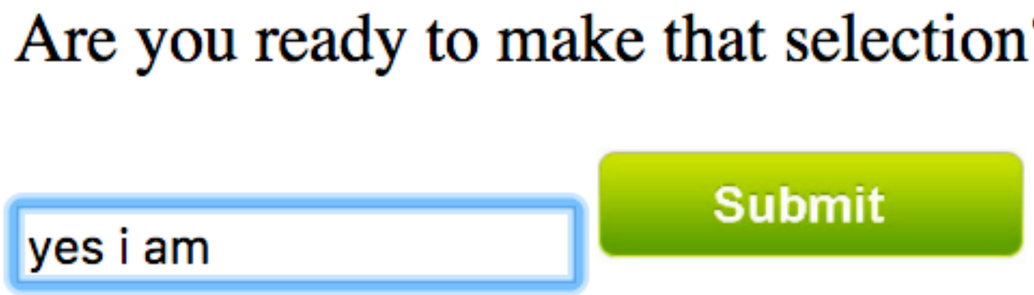


Image button

- Allows to use an image for the submit button
- Use src, width, height attributes to control the size of it

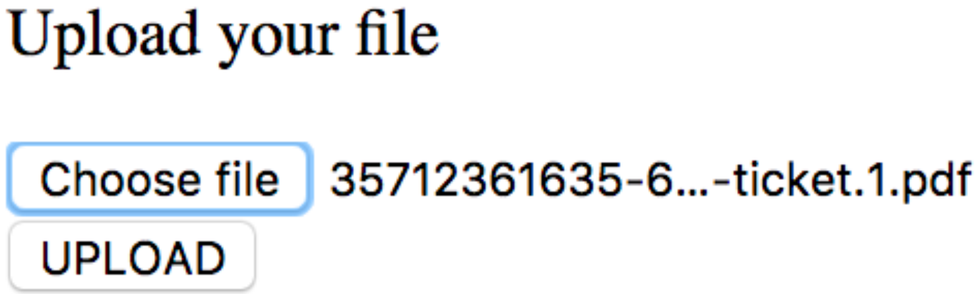
```
<!-- image button -->
<form>
    <p>Are you ready to make that selection?</p>
    <input type="text" />
    <input type="image" src="images/submit.jpg" width="100" />
</form>
```



File input box

- Allows users to upload a file

```
<!-- file input -->
<form>
    <p>Upload your file</p>
    <input type="file" name="user-file" /><br>
    <input type="submit" name="submit" value="UPLOAD" />
</form>
```



<fieldset>

- Groups related form controls together inside one <fieldset> element
- Helpful for longer forms
- Most browsers will show the fields with a line around the edge to show that they are related

```
<fieldset width="100">
    <!-- checkbox -->
    <form>
        <p>Select your favorite input type: <br>
            <input type="checkbox" name="favButton" value="radio" checked="checked"/> Radio
            <input type="checkbox" name="favButton" value="checkbox" /> Checkbox
            <input type="checkbox" name="favButton" value="text" /> Text
        </p>
    </form>

    <!-- submit button -->
    <form>
        <p>Are you ready to make that selection?</p>
        <input type="submit" name="submit" value="SUBMIT" />
    </form>

    <!-- image button -->
    <form>
        <p>Are you ready to make that selection?</p>
        <input type="text" />
        <input type="image" src="images/submit.jpg" width="100" />
    </form>
</fieldset>
```

Select your favorite input type:
Radio Checkbox Text

Are you ready to make that selection?

SUBMIT

Are you ready to make that selection?

Submit

Other useful form types

- Date input (select the date and submit)
- Email & URL input
- Search input

More detailed documentation can be found **here**.

HTML Forms Summary

FROM JOHN DUCKET “HTML AND CSS”

- Whenever you want to collect information from visitors you will need a form, which lives inside a <form> element
- Information from a form is sent in name/value pairs
- Each form control is given a name, and the text the user types in or the values of the options they select are sent to the server
- HTML5 introduces new form elements which make it easier for visitors to fill in forms

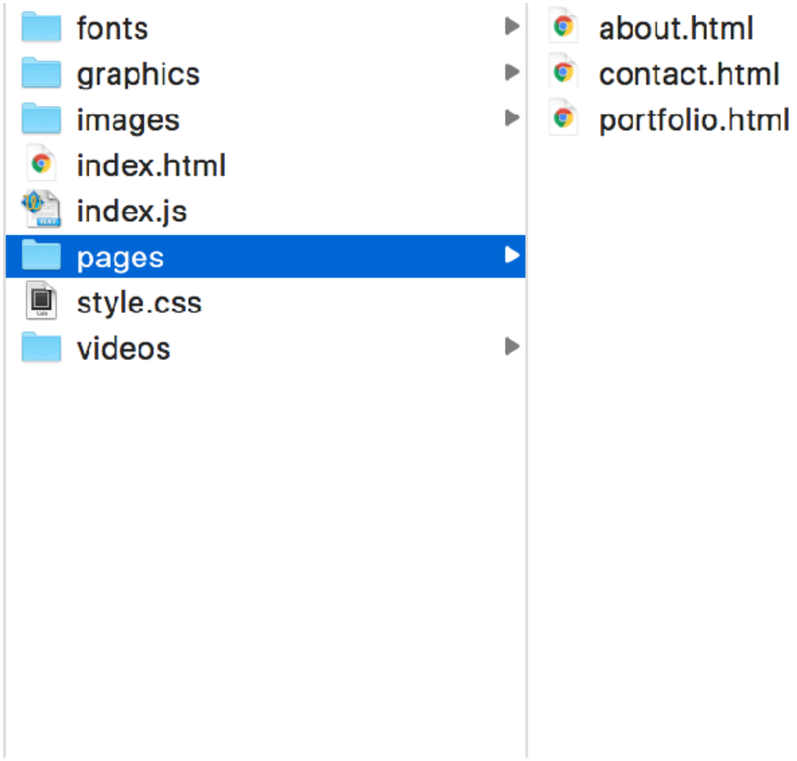
Linking to different pages

We need to link to all the other pages, just like any other links:

About
Contact

To return to our main page from About or Contact pages, we’d write:

Home



Homework assignment

Homework

Due Wednesday by 6pm

1. **Read:**
   - HTML Forms documentation on: https://www.w3schools.com/html/html_form_input_types.asp
   - Contribute a question to #general channel on Slack
2. **Building on the single page from last week, add the following:**
   1. A second HTML page. For the second page you can add additional content and link it in your file.
   2. An additional iframe
   3. An audio element:
      - You can find something online or record your own audio with your phone
   4. A video that relates to the content of your site
      - A video must be locally on your page and linked appropriately. The video should relate to the content of your site. Even better if it's shot by you!
   5. <nav> element or your page (which you can use to link your second page if you choose)
   6. **A table**
   7. **A form - use any 2 elements that we just learned in class**
   8. Be sure to include additional comments
3. **Upload your page to Github by Wednesday 6pm.**
4. **Create a Github page and post link to it on #general channel on Slack by Wednesday 6pm.**
5. **Office hours:** today 6:45-7:45pm. Sign up **here**.