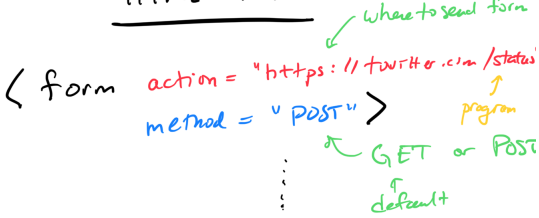
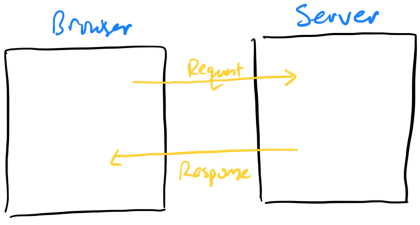
HTML Forms



→ allows user to submit information.

So, at the beginning of the course we talked about how HTTP communication happens...

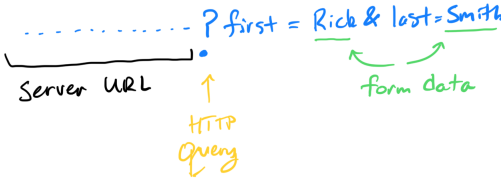


The basic idea of forms is that we will use the information provided from the user to "construct" a new HTTP request, which then will be sent to the server. Then, we will either display or update the webpage being displayed with the response.

Steps:

1. Collect all data from the form fields into a query string

e.g.



2. After the URL is constructed, send the http request, using the GET method.

3. display / update based on the response

What does this look like in HTML ??

```
<body>
    <form action = "https://example.com/apply">
    <p>
        <label for = "first">FirstName:</label>
        <input type = "text" name = "first" id = "first">
    </p>
    <p>
        <label for = "last">Last Name</label>
        <input type = "text" name = "last" id = "last">
    </p>
    <input type = "submit">
    </form>
</body>
```

associate with id =

for the server

for the browser

How do we test this? ... well, we can't !!

The issue is that the program that does all the work → https://example.com/apply isn't real ... it is a SERVER SIDE APPLICATION !!! We have no ability to create/modify server-side applications !!! It is hard work, beyond the scope of this course, and even if we wanted to, it is fraught with peril ... it is a HUGE security risk to allow others to run programs on our servers !!!!

Useful Script for Processing Forms:

https://wp.zybooks.com/form-viewer.php

↑ Zybooks provides this for testing for us !!!

Pulsar on Windows 10

1. Go to pulsar-edit.dev
   - → Releases
     - ↳ Windows
       - ↳ Package
         - Setup - Latest | Beta

2. Run the Windows.Pulsar. ....exe file that is downloaded.
   - → Choose 'Only For Me'
   - → Choose default installation location
   - → Click on 'Finish'
   - → Pulsar should start up automatically !!

3. Packages → Open Package Manager
   - → + Install
   - → Search for atom-live-server
   - → Install atom-live-server
     atom-live-server-plus

4. Go to GitHub.com
   - → If necessary, create a new account, using your CNU Email.
   - → fork brash99/cpsc216.git
   - → Generate Personal Access Token
     - (i) (gear) → Settings →
     - (ii) Developer settings
     - (iii) Personal Access Tokens
       - ↳ Tokens (Classic)
     - (iv) Generate New Token
       - ↳ Generate Classic
       - "cpsc 216"
       - Expiration → never
       - choose all scopes
     - (v) Write Down token, and copy to clipboard !!!!

5. Download and Install GitHub Desktop
   - → sign into GitHub
   - → Clone forked Repo locally
   - → Open in Pulsar

→ commit and push in GitHub Desktop

3.4 Additional Form Widgets

- date picker `<input type="date">`
- color picker `<input type="color">`
- numbers `<input type="number">`
- range/slider `<input type="range">`

See Section 3.4 in the text for more examples !!!

3.5 Audio and Video

a) Audio

```
<audio controls>
    <source src="....">
</audio>
```

Audio Types

- mp3 — The sound portion of an MPEG
- aac — Successor of mp3 → better quality. (Advanced Audio Coding)
- ogg — Ogg Vorbis ← spotify !!
- wav — Wave (Microsoft & IBM) → uncompressed audio on Windows

---

Aside: Ogg Vorbis is developed by Xiph.org ← non-profit OPENSOURCE (in contrast to the others !!)

See History 101 on Netflix

---

b) Video

```
<video controls width="500">
    <source src="____">
</video>
```

Video Formats

- .mp4 } MPEG-4 — Moving Pictures Expert Group
- .m4a — audio only!

- .ogg
- .ogv
- .ogm
} Ogg Theora → Xiph.org

- .webm → WebM → Mozilla / Opera / Adobe / Google for the web.

YouTube Videos

```
<iframe width="460" height="250"
src="https://www.youtube.com/embed/7g7kP_TrpQg
?rel=0"></iframe>
```