Web Design

Forms

Outline

- Forms
  - Form
  - Inputs
  - Submission
  - States
  - Validation
- How the web works

Form Basics

Intro

- Forms allow us to take input from the user

Form Element

- Form content should be wrapped in a <form> element

Inputs

- Main input types:
  - text (from keyboard) (“subtypes”: number, email, tel, url)
    - Additional: range, datetime types, color
  - textarea
  - checkbox
  - radio
  - select

Submission

- How can the user submit
  - Pressing the “enter”/“return” key in a field
  - Clicking a submit button (input or button)

Submission

- Where and how should the form values be sent?
- Action property
  - URL to send form values to
- Method property
  - get, post, dialog

Try It

- Create a form that uses at least 3 different input types and a submit button

Input States and Properties

Input States

- Default
- Focused
- Disabled

Input Properties

- Important properties:
  - id
  - name
  - type
  - value
  - placeholder
  - disabled
  - validation properties

Form Labeling

Importance of Labeling

- Inputs should have labels
  - Lets users know what the field is for
  - Helps screen readers and other non-human users understand the form
- Helps with user interaction
  - Click label to select in radio and checkbox inputs
  - Click label to focus input with other input types
- Note: Don’t use placeholders as labels

Connecting Labels and Inputs

- Putting a label next to an input can work for humans (who aren’t relying on screen readers or similar technology), but it’s best to be explicit with the connection between labels and inputs
- Two methods
  - Wrap the input in the label element
  - Match the `for` property of the label with the `id` property of the input

Try It

- Add labels to the inputs in the form you created earlier (make sure they’re properly connected)

Form Validation

Validation Types

Client- vs Server-Side Validation

- Using HTML (and JavaScript), we can perform client-side validation
- Should be used in conjunction with server-side validation
  - Note that client-side validation alone is not enough to guarantee receipt of valid values
  - Client-side validation serves to let the user know how to correct fill out a form and indicate any errors to them

Input Validation

- Validation using the `type` property (e.g. “number”)
- Validation using other properties (e.g. required, minlength, maxlength, min, max, pattern)

Try It

- Add validation to the inputs in the form you created earlier, or
- Create some new inputs that have validation
  - Required text field
  - Number input with max and min constraints

Accessibility

Importance of Accessibility

- Ensure your website can be used by as many people as possible
- Sometimes is a legal requirement

How to Make Sites Accessible

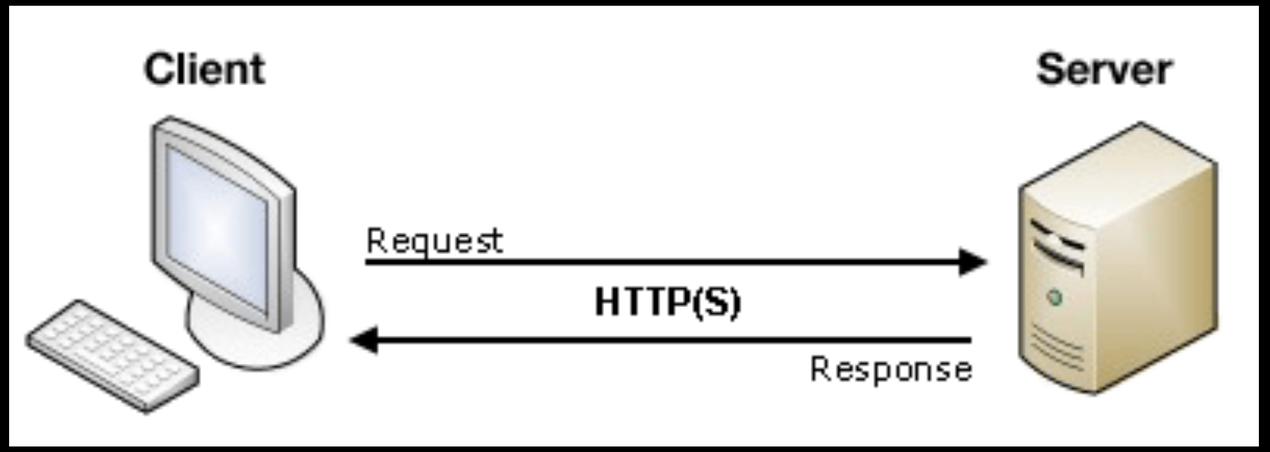
- Write semantic HTML
- Follow best practices
- Ensure elements are linked via code, not just visually
- Read more: https://developer.mozilla.org/en-US/docs/Web/Accessibility

How the Web Works

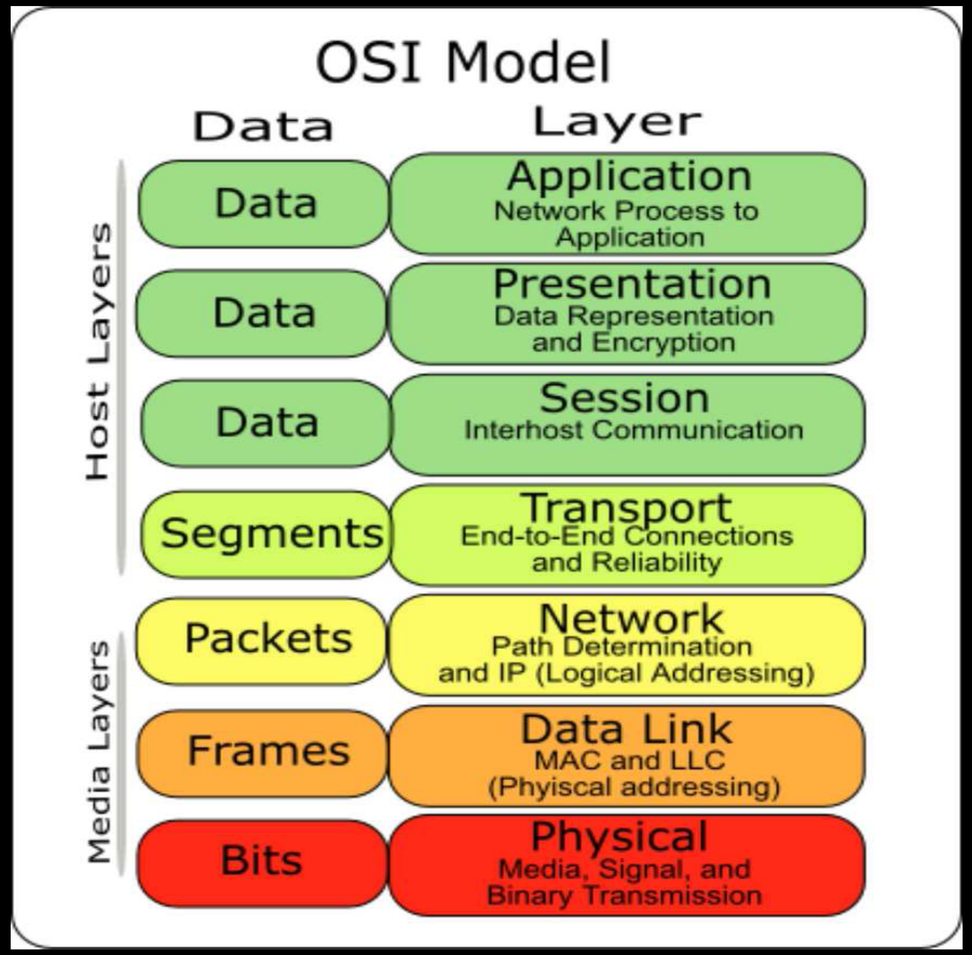
Client-Server Model

How The Web Works

- A client requests information/assets from a server
- A client and server need to share information using agreed-upon standards (protocols)

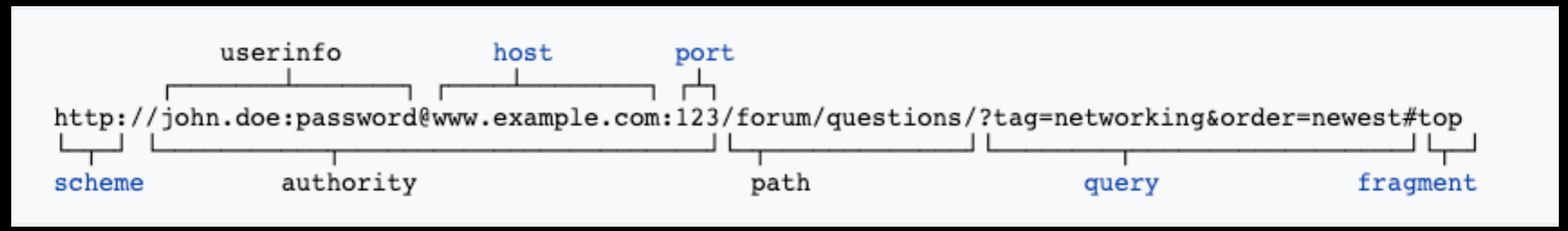


Layered Architecture



Hypertext Transfer Protocol (HTTP)

- Application layer protocol
- Request-response protocol (client-server)
- Is basically a message format
- Methods: GET, POST, PUT, DELETE
- Response Statuses: 200, 400, 500
- Addressing:



Hypertext Transfer Protocol (HTTP) (cont.)

- Request:

```
GET / HTTP/1.1
Host: www.example.com
```

- Response:

```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 155
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
ETag: "3f80f-1b6-3e1cb03b"
Accept-Ranges: bytes
Connection: close

<html>
  <head>
    <title>An Example Page</title>
  </head>
  <body>
    <p>Hello World, this is a very simple HTML document.</p>
  </body>
</html>
```

Transmission Control Protocol (TCP)

- Transport layer protocol
- TCP uses IP to send messages
- TCP breaks messages up into packets and handles packet ordering and failures

Internet Protocol (IP)

- IP controls delivering packets of information between machines
- IP addresses
  - 32-bit for IPv4 (172.250.71.231)
  - 128-bit for IPv6 (2a03:2880:f003:c07:face:b00c)

Domain Names & DNS

- Domains give stable, human readable addresses
- The Domain Name System (DNS) translates domain names to IP addresses

Physical Layer

- The actual wires that the data is sent over

Web Browsers

- Provide a way to make HTTP requests that’s easy to use
- Render HTML/CSS documents and run JavaScript code