CS 144

Web Applications

Discussion 1

Amogh Param

Slide Credit - Luis Angel Larios Cardenas

Logistics

Office Hours

Tuesday, Thursday | 10:30AM - 11:30AM
BH 2432

General questions?

Use the Piazza forum

Individual/private questions?

- Email
  - amoghparam@gmail.com
  - aparam@cs.ucla.edu

Other Discussion Sections

You may attend any of the different discussion sections.

Discussion Sections

1A | 1B

About Me

Machine Learning Researcher | Game Developer | Web Developer | Guitarist | Music Producer

Expectations

What do you guys expect from the discussions?

Piazza

https://piazza.com/ucla/winter2016/cs144

- Post general questions

- Visible to other classmates.
- Option to post anonymously (anonymous to classmates, NOT anonymous to the instructors)

- Private messages to “instructors”:

- Clarify Grades
- Questions

- Email / Office hours

- Personal Issues
- Questions

Piazza

- **Make clear questions**
  - Please provide a lot of detail about your problem/question.
  - Include screen shots, error messages and whatever else you think is necessary to give context
  - Be as specific as possible

- **Do not post any of your code on the forums**
  - It is against the Academic Integrity Rules, since you would be sharing your solutions with other students.

- **Try not sending code to the TAs or to the professor.**
  - Debugging everyone's code is NOT SCALABLE.
  - If you're really stuck with your code, show up for office hours.

Projects and Exam

- One Final **exam**, that is worth **40%**.
- **Projects** worth **60%** in total
    - Projects will be (very likely) graded by someone else.
    - Please follow the directions on the specs.
    - Problem with your grade or want a detailed explanation, contact grader.
    - If the problem is not solved, contact one of the TAs.

Project Submissions

- Rule 1
    - Projects
        - Test your submission in the Virtual Machine (VM).
        - If your *.zip project archive and scripts do not run,you get a zero points.

- Rule 2
    - Use your 4-day grace period wisely (no more than 2 days per project).
    - If you need more days, contact Prof .Cho or your TA
    - Any additional day costs you 20% off your grade.

- Rule 3
    - You may work in a 2-person team to reduce the programming load.
    - If your team dissolves, you cannot team up again.
    - If you submit your work individually for project X, you cannot submit a team-work until project X+1.
    - If you and your partner submit independently, the final grade will be the minimum of both (minus another penalty of 10%)

Project Submissions

- Rule 4
  - You can submit more than once. CCLE will keep only the last archive.
- Rule 5
  - Include a README.txt file whenever you think appropriate,
  - or if you are required to do so (in the specs).
  - Your README.txt may contain
    - Anything you want the reader to take into account.
    - Acknowledge/Cite included work written by other authors.
    - Answers to any questions given in the specs.
- Rule 6
  - Submit your work through CCLE --- No emails with your code attached to them will be accepted.
- Other - Evernote

Review

- How does a browser show a webpage?

- HTTP request (actual bytes sent using TCP/UDP)
- DNS IP lookup
- HTTP response (actual bytes sent using TCP/UDP)

- HTTP

- HyperText Transfer Protocol
- Between Client and Server
- Stateless protocol (all information is in the request)
- Versions : HTTP/1.0 | HTTP/1.1 | HTTP/2

HTTP

- **HTTP Request**
  - **Client sends to Server**
  - **Example**

The actual request

/* **header**: additional information for the request */

Host: the name of the web server
User-Agent: information on the client software
Referrer: The page linking to the requested page.
Accept: Types of media/content acceptable by client
Keep-Alive, Connection: multiple requests one connection

HTTP

- **HTTP Response**
  - **Server sends to Client**
  - **Example**

- Status line:

- **2xx: Success** - The action was successfully received, understood, and accepted
- **3xx: Redirection** - Further action must be taken in order to complete the request
- **4xx: Client Error** - The request contains bad syntax or cannot be fulfilled
- **5xx: Server Error** - The server failed to fulfill an apparently valid request

- ETag: a unique tag that is the same only if the body is the same

- **Content-Length**: length of the body

- **Content-Type**: the type of the content html, flash, pdf, etc.

HTTP

HTTP Verbs/Methods

- **GET**: "retrieve" a resource (no side effect)
  - Should not have any significant side effect at server
  - input values are encoded within URL
- **POST**: "post" data through the specified URL
  - input values are encoded in the body of the request
- **PUT**: "(re)place" the data at location
- **DELETE**: "delete" data at location
- **HEAD**: "retrieve" header only
- **TRACE**: echo the contents of an HTTP Request back to the requester which can be used for debugging purpose at the time of development.

HTML

- **Versions**
  - **HTML 5 -** Current Version
  - **HTML 4.01 -** Most Popular

- **Basic HTML**
  - Tags + Text
    - Tags
      - enclosed in <>
    - Text
      - enclosed within tags

HTML

- **HTML Tags**
    - Represent structure not style
    - Open Tags have matching close tags
        - <p> Lorem Ipsum… </p>
    - Some tags do not need matching tags
        - <img src="http://radwebsi.te/sweet_img.png" />
        -
    - Attributes
        - img
            - src | class | id .....
            -
    - Comments
        - <!-- This is a comment -->

HTML

- **Example HTML**

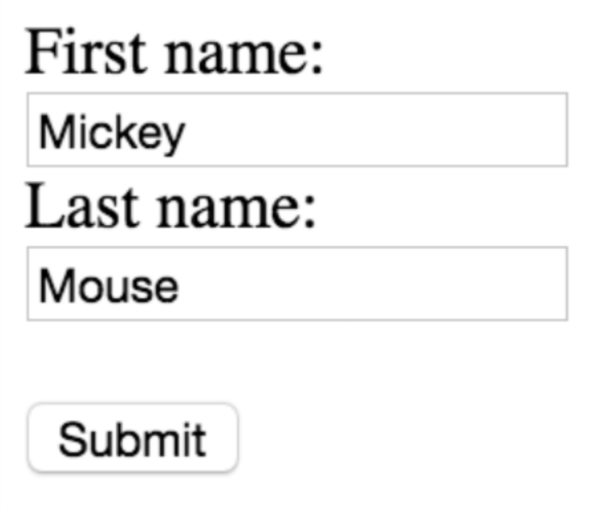
```
<!DOCTYPE html>
<html>
      <head>...</head>
      <body>...</body>
 </html>
```

- **HTML 5**
  - <audio ...>
  - <video ...>
  - drag and drop support
  - offline storage API
  - File system API

HTML

- **Forms**

```
<form action="submit_data_handler.php">
      First name:
     <input type="text" name="firstname" value="Mickey">
     Last name:
     <input type="text" name="lastname" value="Mouse">
     <input type="submit" value="Submit">
</form>
```



First name:

Mickey

Last name:

Mouse

- Types

- text, textarea, checkbox, radio, password, file, hidden, submit.

-**Action**

- the destination of data (or the location of the server process)

CSS

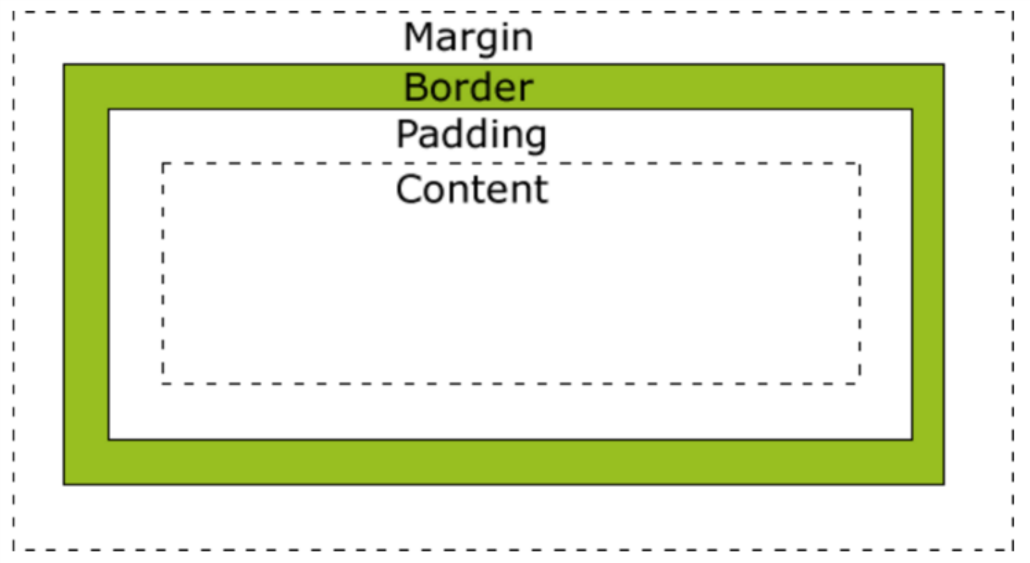
Cascading Style Sheets

- Set of rules for **presentation** and **formatting**
- Levels of Specification
  - External - Document Level
  - Internal - Head Level
  - Inline - Tag Level

- **Why "Cascading"?**
  - Most specific rule wins

CSS

- **CSS layout**
  - **display: block**
    - div, p, ul
  - **display: inline**
    - span, a
    -



- CSS Box Model
  - HTML elements can be considered as boxes.
  - Box that wraps around every HTML element. Consists of:
    - margins, borders, padding, and the actual content.

CSS

- **Positioning**
    - **static (default)**
        - positioned according to normal flow
    - **relative**
        - positioned relative to is normal position
    - **absolute**
        - positioned relative to its nearest positioned ancestor
    - **fixed**
        - positioned relative to the window viewport

- actual position specified through “top”, “bottom”, “left”, “right” properties (except for 'static')

CSS

- **Float property:** “wrap” text around the box
- **Overflow property:** dealing with overflow text
- **Overlapping elements and z-index**

HTML CSS Example