ISEM 501-90 Information and Communications Technologies
Wednesday June 28, 2017

AGENDA:

1. Still grading HW/SW Essays; look for comments by tomorrow

2. Database Essay due next week

3. Web Project due in 2 weeks

4. Tonight:
   - Web Theory (just a bit)
   - Web Project Hands-on

2017-06-21	Online Session		Hardware/Software Essays (Chapters 3 and 4)
2017-06-28	Online Session		
2017-07-05	Online Session		Database Essay (Chapter 5)
2017-07-12	Online Session		Web Project 1

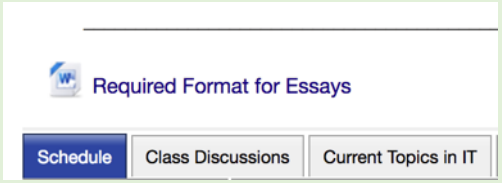
Download class notes

Database Essay – Due July 5, 2017

Assignment:

"Compare and contrast one SQL database system and one NoSQL database system."

- At least 800 words
- Follow standard essay format



Readings:

- Text, Chapter 5
- Codd : Relational Data Model
- SQL Versus NoSQL Movement with Big Data Analytics, Venkatraman et al

Sample References

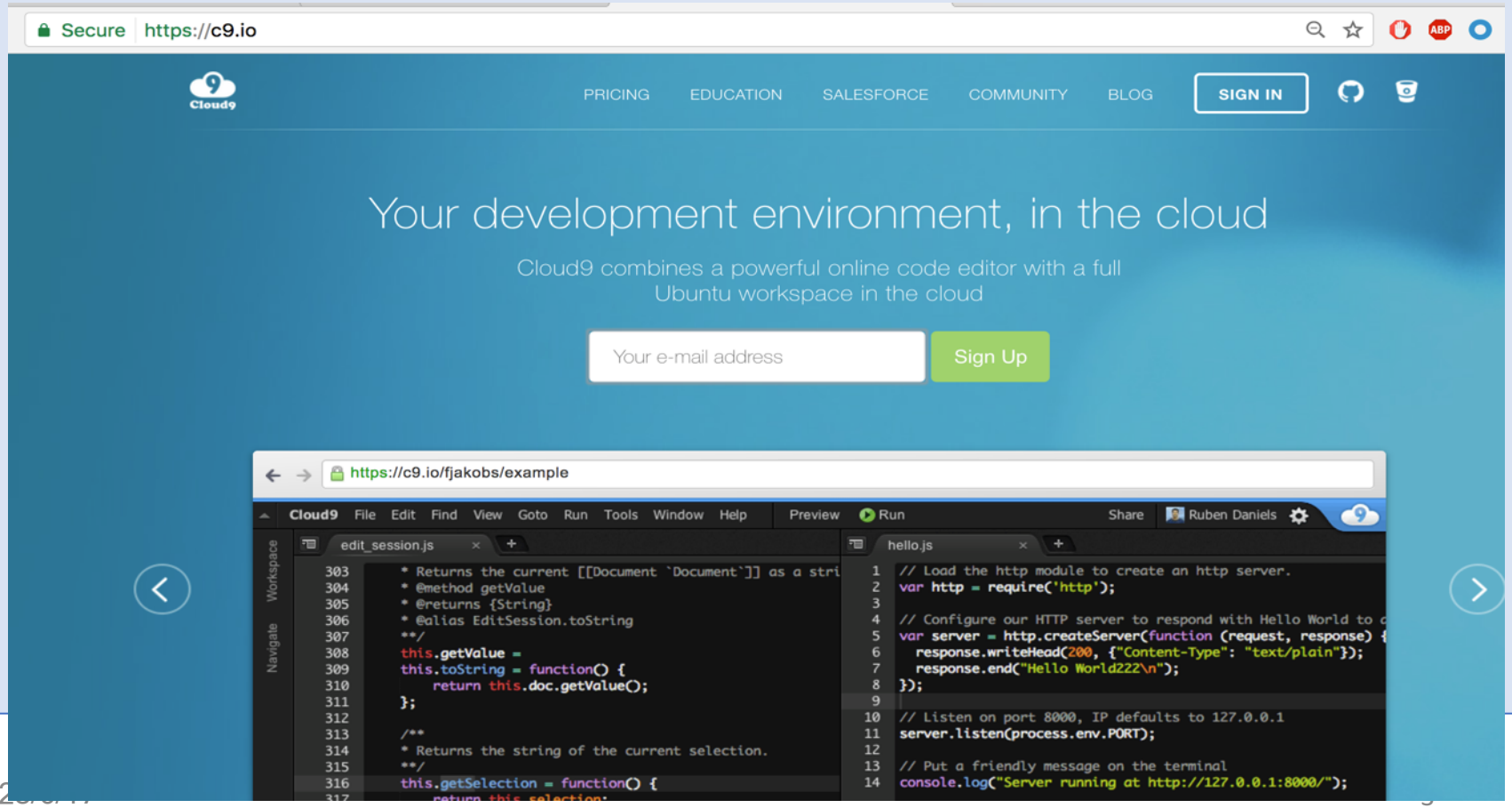
- https://en.wikipedia.org/wiki/NoSQL
- https://en.wikipedia.org/wiki/Comparison_of_relational_database_management_systems

Due Wednesday: July 5, 2017

Cloud 9 (c9.io) : Web-based programming

1. Do you have c9.io access set up?

2. Have you created a project ?



Module 5 - Web site Project


Web Project Requirements


Web site Project 1: "Static" - Due Wed July 12, 2017


Web site Project 2: "Dynamic, no database" - Due Wed July 26, 2017


Web site - Project 3: "Dynamic, with database" - Due Wed Aug 16, 2017


Web Site Project 1 -"Static" Sample

Web Projects

1. I supply the sample code.

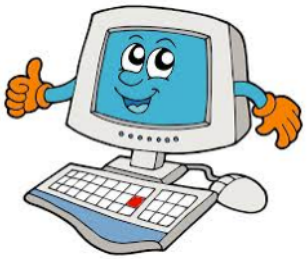
2. You modify the sample code.

3. Grading:

Effort	Grade (Max 30)
No project submitted	Zero
Change a few lines of code (Meet requirements)	20
Add extra features: - HTML - CSS - Javascript - PHP - SQL	21 - 30

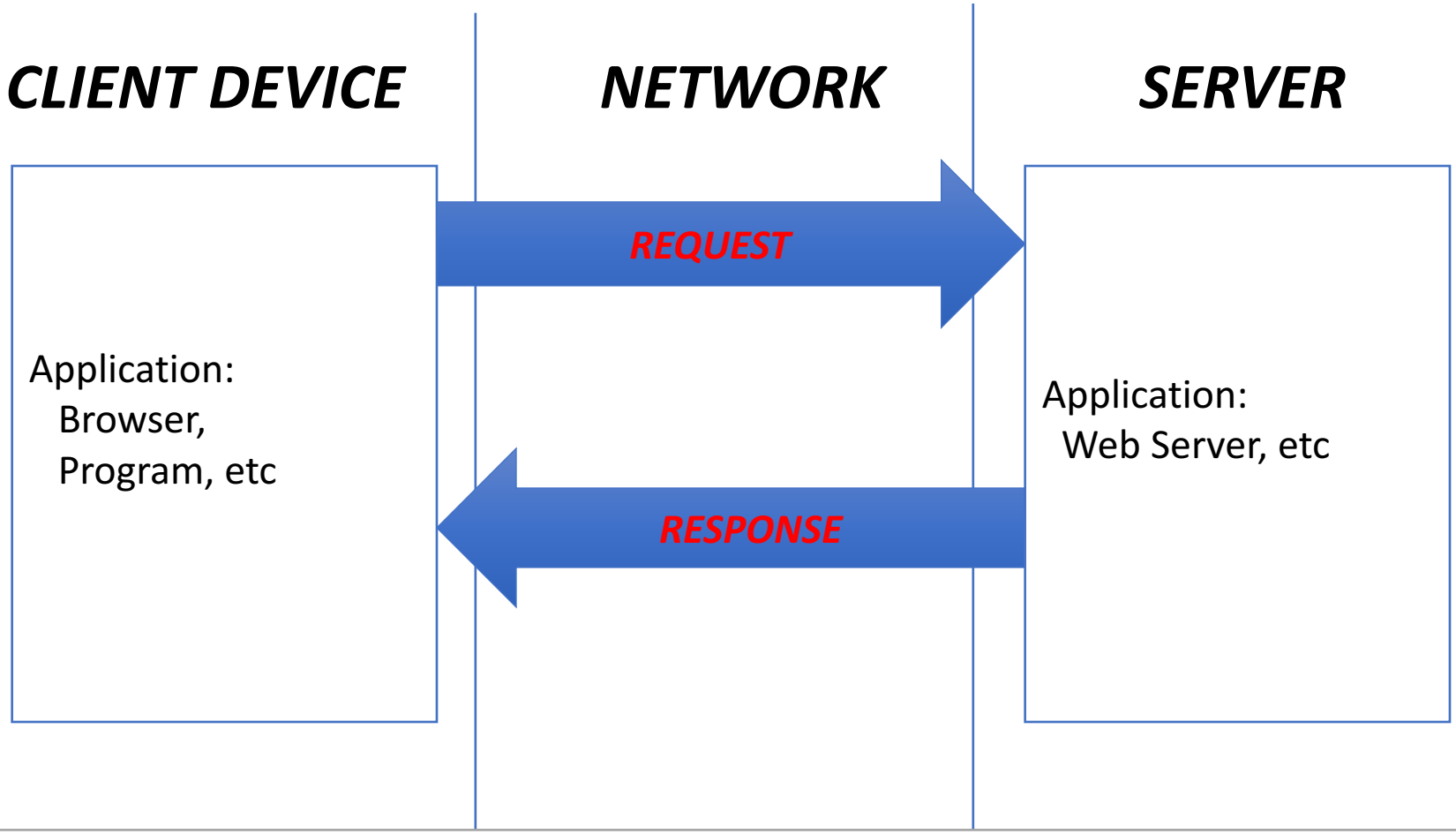
Just enough web theory.....

KISS – Keep It Simple, Sam !



Client…….Internet….Server

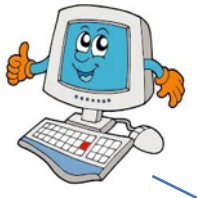
Logical Model



Client.....Internet....Server

Logical Model

1. **Client REQUESTS a resource:**



http://www.google.com

2. **Internet connects client to server**

3. **Server sends RESPONSE back**

Physical Format

1. **Client REQUESTS a resource:**

http://www.google.com

```
Dons-MBP:~ donohara$ wget -v http://www.google.com
--2015-08-07 19:59:51--  http://www.google.com/
Resolving www.google.com... 74.125.226.180, 74.125.226.176, 74.125.226.178, ...
Connecting to www.google.com|74.125.226.180|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'index.html.11'

index.html.11                                           [ <=>

2015-08-07 19:59:51 (1.14 MB/s) - 'index.html.11' saved [18539]

Dons-MBP:~ donohara$
```

Physical Format

1. **Server sends a RESPONSE**

index.html.11

```
<!doctype html>
<html itemscope="" itemtype="http://schema.org/WebPage" lang="en">
  <head>
    <meta content="Search the world's information, including webpages,
    images, videos and more. Google has many special features to help you find ex
    name="description">
    <meta content="/images/google_favicon_128.png" itemprop="image">
    <title>Google</title>
  </head>
  <body>
    <div id="mngb">
      <div id=gbar><nobr>
        <b class=gb1>Search</b>
        <a class=gb1 href="http://www.google.com/imghp?hl=en&tab=wi">Images</a>
        <a class=gb1 href="http://maps.google.com/maps?hl=en&tab=wl">Maps</a>
      </div>
    </div>
  </body>
</html>
```

Google

Search Google or type URL

HTTP: Request/Response commands

- HyperText Transfer Protocol
  - GET
  - PUT
  - POST
  - DLET

HTML: Web Page "Language"

- HyperText Markup Language
  - <HTML>
  - <HEAD>
  - <META>
  - <H1>
  - <A>
  - <TABLE>
  - etc

CSS: Web Page Formatting

- Cascading Style sheets

```
p {
    color: red;
    text-align: center;
}
```

Javascript: Bring web pages to life

```
<img onmouseover="bigImg(this)"
     onmouseout="normalImg(this)”
      src="smiley.gif” width="32" height="32">

<script>
function bigImg(x) {
    x.style.height = "64px";
    x.style.width = "64px";
}
function normalImg(x) {
    x.style.height = "32px";
    x.style.width = "32px";
}
</script>
```

That's all there is to it!

- HTTP:
  - Client REQUESTS something from a server
  - Internet transports request "*data package*" to server
  - Server sends RESPONSE "*data package*" back to client
  - Internet transports response *"data package*" to client

Well, lots of details….

- HTML
  - The language that describes the “data package”
  - One of several “data package” format languages
    - XML
    - JSON
    - Etc
- CSS
  - A way to “style” the data to make it easy to change
- Javascript
  - Language to interact with user on client side

To Keep Things Simple

- We won't be doing much styling
- We won't be doing much javascript
- We may not get very far

The important thing is to get familiar with the tools and the procedures

Resources

http://www.w3schools.com/html/

This is the best place to start.

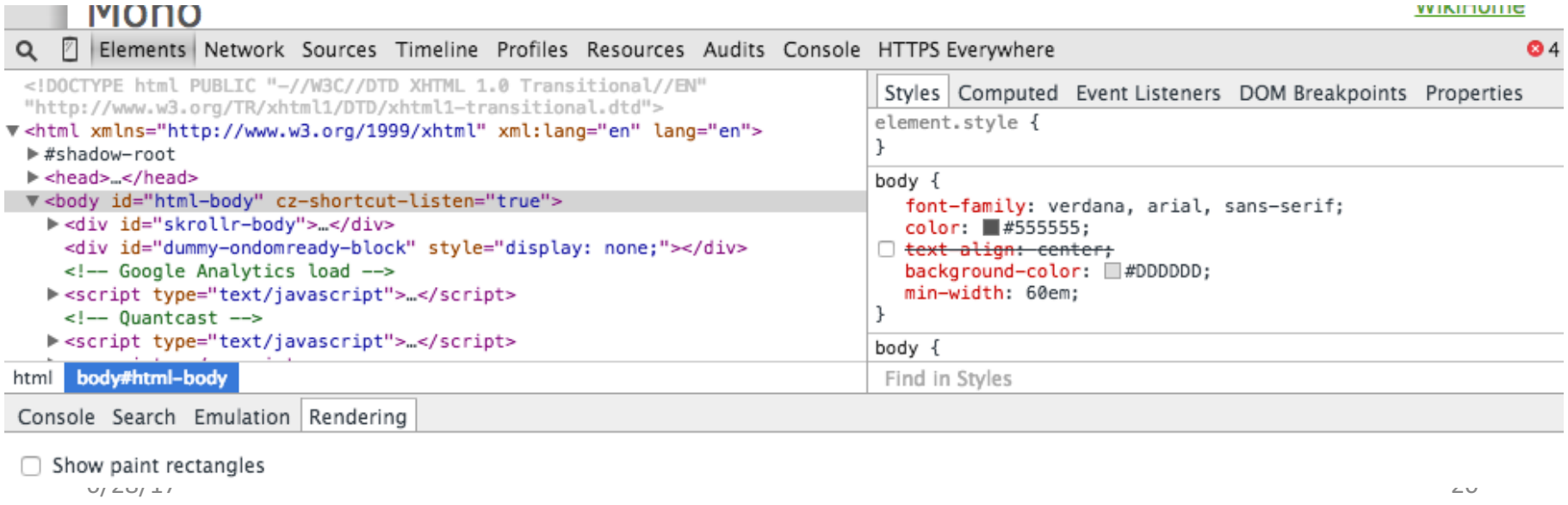
As you progress you will find more, but this *still* is my first stop when looking up syntax, etc.

Resources

Browser Tools:

- View Source
- Developer Tools

Each browser has tools to look at what's happening behind the scenes.

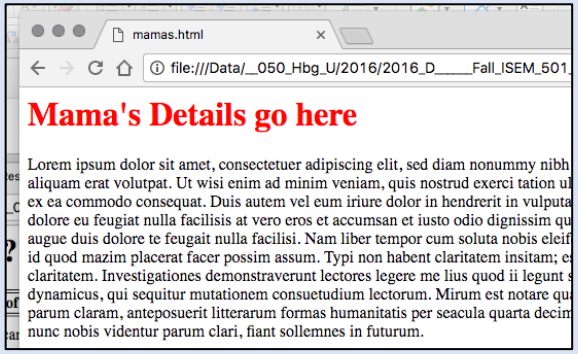


Web Project 1- Sample

Requirements: Present an index page, and link to multiple sub pages

Index Page

Sub Pages



Your Job

Procedure

1. ***Pick a topic that interests you***
   - sports, movies, books, food, travel destinations, cities, etc
   - anything that has a collection of items

2. ***Upload the "Project 1 – Static Sample" to your C9.io account***

3. ***Modify the index.html file***
   - add your own topic details
   - experiment with HTML, CSS, images, JavaScript, etc
     https://www.w3schools.com/html/

4. ***Learn how to use C9.io IDE***
   - tutorials, youtube videos