

CMPT 165

INTRODUCTION TO THE INTERNET AND THE WORLD WIDE WEB

By Hassan S. Shavarani

UNIT 9: DYNAMIC HTML PROGRAMMING

TOPICS

1. Bits and Transmitting Information
2. Server-side Programming

BITS AND TRANSMITTING INFORMATION

WHAT ARE BITS?

for any information to be transmitted over the network
(or stored in your computer's memory or on its disks,) it must be encoded as a sequence of **bits**

WHAT ARE BITS?

- Each bit can have one of two values, usually written as 0 and 1
- it will be something like
"an electrical pulse for 1 but none for 0"

WHAT ARE BITS?

All of the information we can be working with has been turned into bits by our software

e.g. PNG and JPEG formats specify (very different) ways to turn image data into a sequence of bits

BITS AND BYTES

- a **bit** is a single 0 or 1
- A **byte** is a sequence of eight bits

Byte is often a more convenient unit to start with when talking about storing or transmitting information

WHAT ARE BYTES?

there are $2^8 = 256$ possible bytes, usually taken to represent numbers 0–255

00000000	= 0
00000001	= 1
00000010	= 2
00000011	= 3
00000100	= 4
⋮	
11111111	= 255

BITS AND BYTES

2^n is the number of unique values that can be represented by n bits

A kilobit and kilobyte(kb and kB) is $2^{10} = 1,024$ bits and bytes

A megabit and megabyte (Mb and MB) are $2^{20} = 1,048,576$ bits and bytes

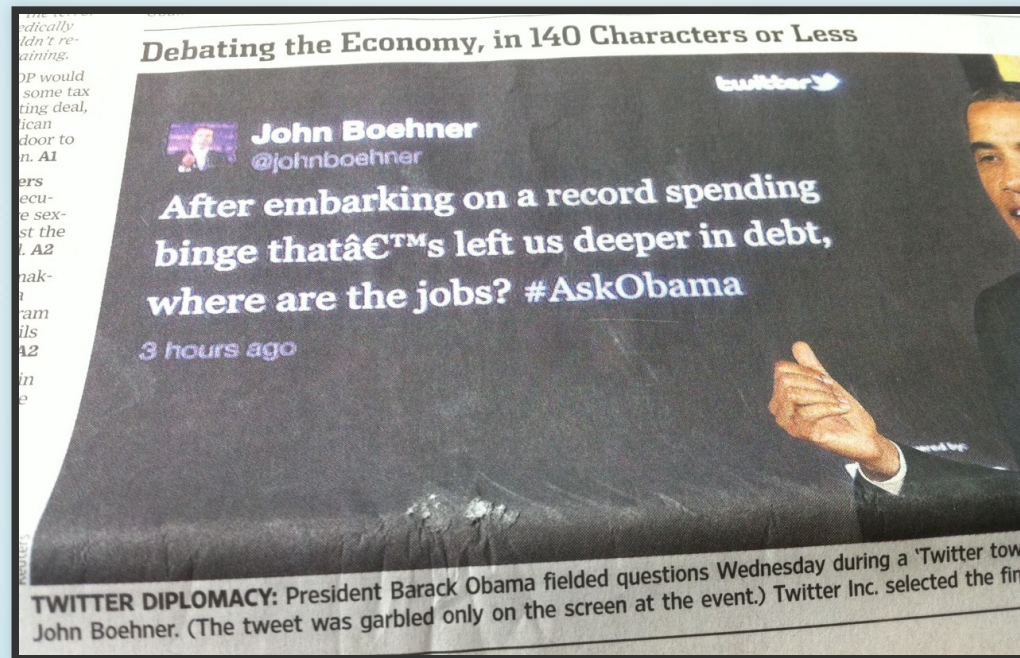
A gigabit and gigabyte (Gb and GB) are $2^{30} = 1,073,741,824$ bits and bytes

we have stored and transmitted characters frequently
HTML, CSS, and JavaScript are all just
characters stored in a text file

these have been sent from server to client for all of our
pages

ENCODING CHARACTERS

- characters must be encoded as bits to be able to be sent through internet
- client and server agree on this encoding



* Image from <http://www.hanselman.com/blog/content/binary/Windows-Live-Writer>

CHARACTER ENCODING

a way to encode the character numbers into bits
there are several character encodings for Unicode
character set (about 120,000 characters)

for example, in Unicode “A” is character number 65, and
“€” is 8364 (so you can produce € in HTML with €)

the choice for web pages is easy: use **UTF-8**

```
<meta charset="UTF-8" />
```

SERVER-SIDE PROGRAMMING

we have learned how to construct web pages,
make them look the way we want,
and give them behaviour

JOB OF WEB-SERVER

the job of the web server is very simple

- find the `.html` file which is put on the server
- send the file out as-is to web browsers

WEBPAGE GENERATION/UPDATE PROBLEM

can we modify the HTML code for adding every single piece of news on the news feed?

can we change the HTML code after each comment a user adds under a post?

can we have a prepared HTML file for every possible search keyword that could come to your mind to ask from Google?

HOW TO MODIFY THE `.html` FILES

we have been able to modify the HTML page with code we have written, but that was all JavaScript code that runs in the web browser

there was never any change in the `.html` file sent from the server to the browser

STATIC VS. DYNAMIC WEBPAGES

static web pages

webpages that come from `.html` files on the server

dynamic web pages

webpages that are created as they are requested

CLIENT-SIDE VS. SERVER-SIDE PROGRAMMING

client-side programming

the programming we have been doing in
JavaScript

(the programs run in the user's web browser)

server-side programming

writing programs to create dynamic pages
(the programs run on the web server)

SERVER-SIDE PROGRAMMING

server-side programs are often written in
Python, Ruby, PHP, Java, or C#

JavaScript is also a common choice
since most web developers already know it

SERVER-SIDE JAVASCRIPT

there exists a long list of different libraries
to be used for server-side JavaScripting
[[you can check the list here](#)]

[Node.js](#) is one of the most common server-side
JavaScripting libraries

SERVER-SIDE JAVASCRIPT: NODE.JS EXAMPLE

```
var http = require('http');
var server = http.createServer(function(req, res) {
  res.setHeader('Content-type', 'text/html');
  res.writeHead(200);
  res.write('<!DOCTYPE html>\n');
  res.write('<html><head><meta charset="UTF-8" />\n');
  res.write('<title>Node.js app</title></head><body>\n');
  res.write('<p>This is a page generated by Node.js.</p>\n');
  for ( i=1; i<=10; i+=1 ) {
    res.write('<p>Another paragraph. ' + i + '</p>\n');
  }
  res.write('</body></html>\n');
  res.end();
});
server.listen(8080);
```

you can try this example [here](#)!

You can start learning about Node.js library in [here](#)

Any Questions?