Csci 4131 Internet Programming
Fall 2021
September 13th

Instructor: Dr. Dan Challou

Logistics – Csci 4131 Lecture 2, 9/13

- **Note** my office hours have been added to the calendar

  The course calendar has been updated to reflect the change - Monday's in person in 383 Shepherd (with a virtual connection via zoom as well) and Wednesday's from 7:15pm – 8:15pm at night virtually via zoom.
- The course calendar, which contains zoom links to all the office hours is publically accessible, so you can all get to it!!!
- HW Assignment 1 Out, Due THIS Friday 9/17 at 11:59pm
  - You can get the assignment write up in the assignment item:
    - **Homework Programming Assignment 1 Specification and Submission Link** in the week 1 module on the class Canvas
    - I will demo in class today.

This week's assignments, readings and tutorials (in addition to HW Programming Assignment 1)

Zybooks Assignments (see your zybook),
www.w3schools.com - CSS Tutorial, JavaScript Tutorials
https://www.w3schools.com/css/default.asp
http://www.w3schools.com/js/
Optional: Sebesta Chapters 3,4

Questions?

Agenda

- Last Time:
  - Course Logistics – Intro and Overview
  - History of the Internet
  - Started How Computers Represent text…
- Today
    - Get going with HTML

Questions?

Question: Internet = World Wide Web?

- **Raise your hand if you agree**
- **Stand up if you disagree**

Let’s start understanding the Elements that are used to make up the internet

- Recall, the internet was created to enable the exchange of information between people
- Information is exchanged in various ways
  - Text
  - Audio/ Voice / Music
  - Pictures / Videos
  - Touch (Haptic)
  - Smell (e.g. Pheremones)

- But, it all started with text (and some rudimentary sounds)

Recall: Evolution of How Computers Represent Text

- Before Digital Images, Digital Video, Digital Music, etc. :
  - Computers were used to manipulate numbers
  - Earliest machines were used in number crunching – computing artillery firing tables, decrypt codes used by opposing forces to transmit messages, etc

And then there was ASCII

- American Standard Code for Information Exchange (circa 1960)
- Numeric Scheme for encoding characters of the English Alphabet
- Created by Bell Telephone Laboratories (aka – "The Phone Company") to support their tele-printers

http://en.wikipedia.org/wiki/ASCII

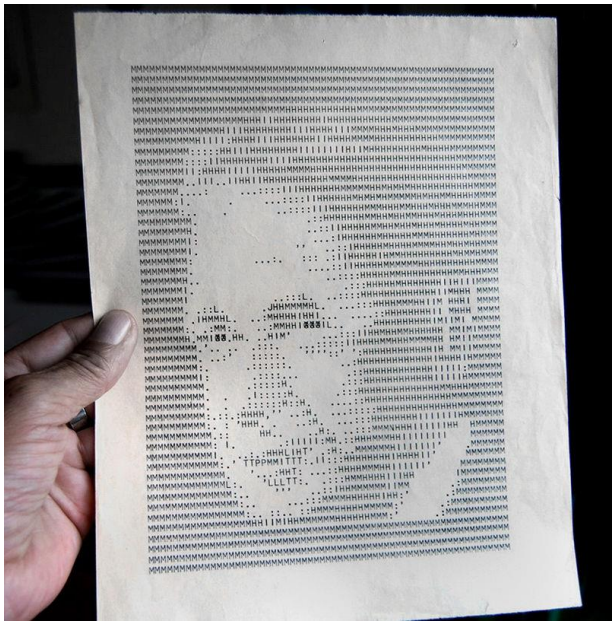
USASCII code chart

b_4	b_3	b_2	b_1	Row ↓ / Column →	0 (0 0 0)	1 (0 0 1)	2 (0 1 0)	3 (0 1 1)	4 (1 0 0)	5 (1 0 1)	6 (1 1 0)	7 (1 1 1)
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	|
1	1	0	1	13	CR	GS	-	=	M	]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	_	o	DEL

A much more usable conversion chart…

Decimal - Binary - Octal - Hex – ASCII Conversion Chart

Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII
0	00000000	000	00	NUL	32	00100000	040	20	SP	64	01000000	100	40	@	96	01100000	140	60	`
1	00000001	001	01	SOH	33	00100001	041	21	!	65	01000001	101	41	A	97	01100001	141	61	a
2	00000010	002	02	STX	34	00100010	042	22	"	66	01000010	102	42	B	98	01100010	142	62	b
3	00000011	003	03	ETX	35	00100011	043	23	#	67	01000011	103	43	C	99	01100011	143	63	c
4	00000100	004	04	EOT	36	00100100	044	24	$	68	01000100	104	44	D	100	01100100	144	64	d
5	00000101	005	05	ENQ	37	00100101	045	25	%	69	01000101	105	45	E	101	01100101	145	65	e
6	00000110	006	06	ACK	38	00100110	046	26	&	70	01000110	106	46	F	102	01100110	146	66	f
7	00000111	007	07	BEL	39	00100111	047	27	'	71	01000111	107	47	G	103	01100111	147	67	g
8	00001000	010	08	BS	40	00101000	050	28	(	72	01001000	110	48	H	104	01101000	150	68	h
9	00001001	011	09	HT	41	00101001	051	29	)	73	01001001	111	49	I	105	01101001	151	69	i
10	00001010	012	0A	LF	42	00101010	052	2A	*	74	01001010	112	4A	J	106	01101010	152	6A	j
11	00001011	013	0B	VT	43	00101011	053	2B	+	75	01001011	113	4B	K	107	01101011	153	6B	k
12	00001100	014	0C	FF	44	00101100	054	2C	,	76	01001100	114	4C	L	108	01101100	154	6C	l
13	00001101	015	0D	CR	45	00101101	055	2D	-	77	01001101	115	4D	M	109	01101101	155	6D	m
14	00001110	016	0E	SO	46	00101110	056	2E	.	78	01001110	116	4E	N	110	01101110	156	6E	n
15	00001111	017	0F	SI	47	00101111	057	2F	/	79	01001111	117	4F	O	111	01101111	157	6F	o
16	00010000	020	10	DLE	48	00110000	060	30	0	80	01010000	120	50	P	112	01110000	160	70	p
17	00010001	021	11	DC1	49	00110001	061	31	1	81	01010001	121	51	Q	113	01110001	161	71	q
18	00010010	022	12	DC2	50	00110010	062	32	2	82	01010010	122	52	R	114	01110010	162	72	r
19	00010011	023	13	DC3	51	00110011	063	33	3	83	01010011	123	53	S	115	01110011	163	73	s
20	00010100	024	14	DC4	52	00110100	064	34	4	84	01010100	124	54	T	116	01110100	164	74	t
21	00010101	025	15	NAK	53	00110101	065	35	5	85	01010101	125	55	U	117	01110101	165	75	u
22	00010110	026	16	SYN	54	00110110	066	36	6	86	01010110	126	56	V	118	01110110	166	76	v
23	00010111	027	17	ETB	55	00110111	067	37	7	87	01010111	127	57	W	119	01110111	167	77	w
24	00011000	030	18	CAN	56	00111000	070	38	8	88	01011000	130	58	X	120	01111000	170	78	x
25	00011001	031	19	EM	57	00111001	071	39	9	89	01011001	131	59	Y	121	01111001	171	79	y
26	00011010	032	1A	SUB	58	00111010	072	3A	:	90	01011010	132	5A	Z	122	01111010	172	7A	z
27	00011011	033	1B	ESC	59	00111011	073	3B	;	91	01011011	133	5B	[	123	01111011	173	7B	{
28	00011100	034	1C	FS	60	00111100	074	3C	<	92	01011100	134	5C	\	124	01111100	174	7C	|
29	00011101	035	1D	GS	61	00111101	075	3D	=	93	01011101	135	5D	]	125	01111101	175	7D	}
30	00011110	036	1E	RS	62	00111110	076	3E	>	94	01011110	136	5E	^	126	01111110	176	7E	~
31	00011111	037	1F	US	63	00111111	077	3F	?	95	01011111	137	5F	_	127	01111111	177	7F	DEL



https://www.youtube.com/watch?v=KTV4YPTv9GQ

- So – what was the answer to ASCII's inability to represent languages other than English?

  – Unicode Transformation Format (UTF)

Unicode 8 seems to becoming the Defacto Standard

- Unicode 8 – not big or little endian (does not depend on how computer hardware is implemented)
  - Big Endian - most significant byte in smallest (lowest) address
  - Little Endian – most significant byte in largest (highest) address
- Supports the legacy ASCII character set

Additional Supplemental Reading (you should read these…)

- http://unicode.org/standard/WhatIsUnicode.html
- https://en.wikipedia.org/wiki/Unicode
- http://doc.cat-v.org/bell_labs/utf-8_history

UTF Enables Web Browsers to Know How to Display HTML

- Hyper-text Markup Language is used to mark up text, which is represented in Unicode
- To correctly process HTML, a web browser must ascertain which Unicode characters are represented by the encoded form of an HTML document. In order to do this, the web browser must know what encoding was used
- http://en.wikipedia.org/wiki/Unicode_and_HTML

Questions?

HTML

- Hyper
- **TEXT**
- Markup
- Language

Basics (Code Along)

- Webpages consist of a collection of nested HTML elements
- The foundation of a page is the <html> element. All elements other than the version tag (element) are placed inside of it
- <html>
- [your webpage!]
- </html>
- Your actual page content goes inside the <body> element
- Segway to Try1.html (Try1.html)
- And check it with the validator (https://validator.w3.org)

Let’s build a simple, Correct web-page:

CODE ALONG

- Structure
  - Version tag (if html 5)
  - html tag
  - head tag
  - Title tag (for tab)
  - Meta tag, with information on the character set
  - Tag closing head
  - Tag opening body
    - Other HTML elements
  - Tag closing body
  - Tag closing html page

Try2.html

And, check it with with the validator:
https://validator.w3.org

Let’s look at an another HTML5 Web Page

www.w3schools.com/html/tryit.asp?filename=tryhtml_basic_link

www.w3schools.com –
Is an excellent resource for a number of the course topics

Elements

- Elements are the building blocks of HTML
- Elements give pages structure
- We use them to format text, add images, create tables, and much more
- They are defined by (usually) two tags, a start and end tag
- <tag> [element content] </tag>
- We'll cover some common elements later

In-class Exercise 1

- Use a text editor for example, NotePad , or Notepad++ (**Not** MS Word), or TextEdit (Mac) and create an HTML 5 Webpage that:

1. Displays your name on the TAB opened by the browser
2. Has a header that says: My Favorite Site
3. Has a link that says: click here, and when you do, it opens up your favorite site

For example: Exercise1.html (NOTE: This will not be included with the lecture 2 examples because it is an answer to the exercise – and we would like you to formulate your own answer.)

If you don't have a computer, write your answer on a piece of paper, transfer it to electronic format using one of the text editors listed above, and upload your answer to Canvas

Submit your file via the class Canvas Site using the Link named:
Lecture 2, Exercise 1 Submission Link (September 13th)

Questions?

Validating Your HTML

https://validator.w3.org

– for HTML5,

https://jigsaw.w3.org/css-validator/

- for CSS

- Provide URL of online web page
- Upload a file to the validator
- or, Paste code directly into a text area provided on the validator suite

Make sure to try the validator on the HTML file you wrote for your Exercise

Lists

-   - ordered (numeric lists)
-   - unordered (bullet lists)
- Each Element

  Example: ListEx.html

Exercise 2 (Just for Fun) – I'll code along, *you can go ahead of me*

- Make an ordered list of your 2 favorite food groups – choose from:
    1. Fats, Oils, Sweets
    2. Dairy
    3. Protein
    4. Vegetable
    5. Fruit
    6. Grain;
- Each food group should contain an unordered nested list of your two favorite foods from each food group.
- Example: FoodsEx2.html

Tables

- <table> </table>
- *All elements below can be used (or not) after start tag for table:*
- <thead> </thead> - table heading
  - Use <th> </th> for a bold fond heading
- <tfoot> </tfoot>
  - Same as head if you want bold font in footer
- <tr> </tr> - elements specify a row
- <td> </td> - data element in a row (column)
- <caption> </caption> - caption for table

Simple Table Example

- L2_table_ex1.html

Two Column Tables

- L2_table_ex2A.html
- L2_table_ex2_v2.html

So How Do We Make a Multi-Column Table with collapsed borders and alternating row colors?

- http://www.w3schools.com/html/html_tables.asp

Exercise 3

- Create a Two row table, that has two columns
  - The first column in each row should contain the name of one of your favorite movies,
  - The second column in each row should contain your favorite actress or actor in that movie.

If you don't have a computer, write your answer on a piece of paper, transfer it to electronic format using your favorite text editor, and upload your answer to Canvas

Submit your file via the class Canvas Site using the Link named:

Lecture 2, Exercise 3 Submission Link (September 13th)

Questions?

Some Other Useful HTML5 Tags – block and inline elements

- Div, Span, and Navigation tags
- Nav Useful for creating a multi-page Website
  (see homepage.html / Page1.html example)
- Can alternately use div if no browser support
- Span is an inline element – useful for inline styling!
- See - http://www.w3schools.com/html/html_blocks.asp

Questions?

Next Time:

- More on HTML
- CSS?

Make sure to do the readings and tutorials for Week 2 listed in the Week 2 Module on the Class Canvas site **(and in the course schedule in Resources Module) – and do your zybooks Lecture prep and homework**