Text and Graphics

September 26, Unit 3

How Computers Store Data

- Computers store everything in *bits*
  - In memory (RAM)
  - Hard drive
- Data is a string of 1's and 0's
  - 00101010111….
- A *bit* is a 0 or 1
- All data is stored as a string of bits
  - Numbers
  - Text
  - Images
  - Everything!

Numbers

- Numbers are the simplest things to store in bits
- Once we can store numbers, we can convert other types of data to numbers to store them
- Storing numbers requires us to convert them from base 10 to base 2 (binary)
  - Just like the very brief example for converting RGB to hex values for color

Converting to Binary

- Binary has two digits: 0 and 1
- Lets look at the number 72 in decimal (the normal way we count)

- $72 = (7 * 10^1) + (2 * 10^0)$
- Just like elementary math: 1's, 10's, 100's,1000's, etc. for each column
- Or, 10^0, 10^1, 10^2, 10^3,

Bases

- All bases work exactly the same way
- The available digits are going to be 1 less than the base
  - Base 10 : 0-9
  - Base 2 (binary): 0,1
  - Base 5: 0-4
  - Base 16 (hexadecimal, or just hex): 0-F (0-15)

Bases Continued

- Take the number 12931_{10}
  - Subscript indicates base
- That is:

Digit	1	2	9	3	1
Value for that column	10^4	10^3	10^2	10^1	10^0

An Arbitrary Base

- Each value of each column is that base, say *k*, raised to the appropriate power
- This value gets multiplied by the digit in that column

k^4	k^3	k^2	k^1	k^0
10^4	10^3	10^2	10^1	10^0

Binary (base 2)

- Easiest (besides base 10)
- Easy because its determining the value for the column is simple (multiply the last column by 2)

128	64	32	16	8	4	2	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

Converting to Binary (or any base), and Back

- In class example
- For binary and any base

Bytes

- A bit is a single digit 0 or 1
- Bits usually get arranged into bytes
- 8 bits = 1 byte
- 1 byte can represent 256 numbers
  - How?
  - Number 11111111_2 = (128+64+32+16+4+2+1) = 255
  - 1 byte can represent from 0-255, or 256 numbers

1	1	1	1	1	1	1	1
128	64	32	16	8	4	2	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

Max Value of a Group of Bits

- For any group of n bits, you can have 2^n values
- These values will always range from 0 to $2^n - 1$
- Some standard groups are:
  - 8: a byte, 0 to 255
  - 16: 0 to 65535
  - 32: 0 to 4294967295

Storage

- Pretty awkward if we had to say how many bits of storage our computers had
- We use prefixes to simplify this
- See figure 3.1 in your course pack
- We commonly refer to bytes, not bits.
  - Bytes are too small
- Most common:
  - Kilo-: $2^{10} \approx 10^3$
  - Mega-: $2^{20} \approx 10^6$
  - Giga-: $2^{30} \approx 10^9$
  - Tera-: $2^{40} \approx 10^{12}$

We still don't Measure Exactly

- 1 kB (1 kilobyte) is 1024 bytes
  - Kilo in metric though is 1000
  - 1kB is not 1000 bytes
  - But, its roughly 1000 bytes
- Same goes for measuring hard drive storage
  - We use GB or gigabytes mostly
  - But a 60GB hard drive is really 55.88 GB

Text

- We store text by converting it to a string of bits
- Takes two steps:
  - 1. Convert the text to numbers
  - 2. Convert numbers to bits
- Many different ways to translate between text and numbers
  - A translation is called a *character set*

ASCII

- ASCII is the most common translation from text to numbers
- American Standard Code for Information Interchange
- Each character has a numeric value
  - A: 65
  - a: 97
  - ; (semi-colon) : 59

ASCII, cont.

- ASCII defines 127 different characters
- Every PC uses ASCII
- There are even variants for different regions
- Also there are extensions for up to 255 characters
  - Allows for things like accents
- 255 is the maximum number of characters which can be defined if we are limited to 1 byte per character
  - $2^n - 1$

Unicode

- Unicode is like ASCII but supports a LOT more characters
  - 2^{32} characters to be precise
- Goal is to have a character set which can be used for any language
  - Support for Chinese, Japanese, Greek
  - Lots of special symbols
- Current Unicode standard defines 95221 characters

Unicode in XHTML

- You can add Unicode characters to your html
- For instance the x or times symbol is the Unicode character 215
- You add the symbol as an entity
  - &#”Unicode character”;
  - Or ×
- Any Unicode symbol can be added this way

Text Files

- Word processors and text editors are different
- You've been using a text editor to write your html (I hope)
- Text editors store the files as pure characters
  - There is no formatting

Text Editors

- When a text editor saves a file, it saves it one byte, or one character, at a time.
- When it reads a text file, it does the reverse
  - Reads one byte at a time and converts that to a character
- Text editors can open a lot of different types of files because its dealing with just the bytes for that file
- If you open a non-text file, you will get very odd characters or those empty boxes when it can't translate
- If you open a file from say MS Word, it won't be readable in notepad

Word Processors

- Word processors, like MS Word, are different than text editors in that they contain formatting information
- They have to store the formatting information along with the text
- Each word processor does this differently
  - Which is why its often difficult to open files created with another program
  - May sort of work, but be just a bit off

Benefit of Text Editors

- Because they only store characters, you can open, view, and edit them with any text editor
- *Text files* refers to files created with a text editor
  - Html
  - CSS
  - Python
  - .txt , *plain text file*

Fonts

- Each character is stored numerically
- Computer still needs to know how to display it on the screen
  - What the character “looks” like
- Each character can be drawn in many different ways
- “*Font* (or *Typeface*) is a collection of drawings, one for each character”

Glyphs

- The drawing for a character is called a *glyph*
- It is a picture of that character
- A font is then a collection of glyphs
- Think of all the fonts you have on your computer
  - Would you want to store the glyphs for all 95000 characters for every font?
- If the font you're using does not have a glyph for the Unicode character you want, your computer will try to substitute for it

Questions?