- ASCII
    - character encoding scheme
    - based on ordering of english alphabet
    - represents text in computers
    - 7-bit: 0-127 (printable: 32-126)
    - A=65, Z=90, a=97, z=122, 0=48, 9=57
    - later, 8 bits for more characters (extended ASCII)
- base-64
    - encodes binary data by translating it into base 64 representation
    - used for transmission media that can only handle text-based data
    - choose a 64 character set that is common to many systems
        - e.g. A-Za-z0-9 (62 values); and add + and / (A=0, /=63)
    - “Wit”
        - W=87, i=105, t=116
        - 01010111, 01101001, 01110100
        - 010101110110100101110100
        - divided into 6 bits (64 different binary values)
        - so 4 characters in base-64 to represent 3 in ASCII

<table>
<tr><td>Type</td><td colspan="8">W</td><td colspan="8">i</td><td colspan="8">t</td></tr>
<tr><td>ASCII</td><td colspan="8">87</td><td colspan="8">105</td><td colspan="8">116</td></tr>
<tr><td>Binary</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>Index</td><td colspan="6">21</td><td colspan="6">54</td><td colspan="6">37</td><td colspan="6">52</td></tr>
<tr><td>Base-64</td><td colspan="6">V</td><td colspan="6">2</td><td colspan="6">L</td><td colspan="6">0</td></tr>
</table>

- ciphers
    - caesar
        - shift cypher
        - ABCDEFGHIJKLMNOPQRSTUVWXYZ
        - DEFGHIJKLMNOPQRSTUVWXYZABC
        - WIT → ZLW
        - how could we break this cypher?
    - coke can (or wasabi and soy almond) crypto
- hashing
    - simply put: converts large (maybe variable sized) data into small (fixed size) data
    - often the data serves as index into an array (hash table)
    - we need to be aware of collisions
    - we need to be aware of reversibility
    - perfect hashing: no collisions
    - MD5
        - message digest algorithm 5
        - cryptographic hash function (128-bits)
        - also used to check integrity of files
        - not collision resistant
        - usually expressed as 32-bit hex number
        - input message broken up into chunks of 512-bits
        - padded so length is divisible by 512
            - single bit 1
            - zeros (bring message to 64 bits less than a multiple of 512)
            - 64-bit integer representing the length of the original data in bits
        - 128 bit state divided into 4 32-bit chunks (a, b, c, d) initialized to fixed constants

    - operate on each 512-bit chunk of data and modify the state
    - 4 total stages (rounds) of 16 operations:
        - $F(x,y,z)=(x\wedge y)\vee(\neg x\wedge z)$
        - $G(x,y,z)=(x\wedge z)\vee(y\wedge\neg z)$
        - $H(x,y,z)=x\oplus y\oplus z$
        - $I(x,y,z)=y\oplus(x\vee\neg z)$
    - “” → d41d8cd98f00b204e9800998ecf8427e

- digital signatures
    - to determine authenticity of message
    - a type of asymmetric cryptography
    - used in authentication and integrity
        - is this message from a trusted source?
        - has the message been changed in transit?
- symmetric
    - single secret key exchanged
        - used to encrypt and decrypt
    - e.g. AES, serpent, twofish, blowfish
- asymmetric
    - public-key cryptography
    - 2 keys
        - public key used to encrypt
        - private key used to decrypt
        - related mathematically
    - e.g. Diffie-Hellman, RSA