**Endian**

Purpose:  
1. Networking , any.  
2. Doing some hacky trick to encrypt your data

Data is stored 8 bits/1 byte at a time. 2 ways of storing, Little Endian or Big Endian.

Example of an int = 0xABCDEF12

| **Most significant bit (MSB)** |  |  | **Least significant bit (LSB)** |
| --- | --- | --- | --- |
| AB | CD | EF | 12 |

32-bit Memory Layout: storing an int = 0xABCDEF12 , a short: 0x3456 , a char = 0x13, a char = 0x77

| **Big Endian** |  | **Little Endian** |  |
| --- | --- | --- | --- |
| 0x00000000 (Low Address) | AB | 0x00000000 (Low Address) | 12 |
| 0x00000001 | CD | 0x00000001 | EF |
| 0x00000002 | EF | 0x00000002 | CD |
| 0x00000003 | 12 | 0x00000003 | AB |
| 0x00000004 | 34 | 0x00000004 | 56 |
| 0x00000005 | 56 | 0x00000005 | 34 |
| 0x00000006 | 13 | 0x00000006 | 13 |
| 0x00000007 | 77 | 0x00000007 | 77 |
| 0x00000008 (High Address) |  | 0x00000008 (High Address) |  |

How to remember?  
1. Lets set an anchor point where memory layout always stars at 0, normal way of remembering (e.g. 0x00000000)  
2. Big Endian stores it’s values as it is  
3. Little Endian stores it’s values flip

Byte by Byte

Reference:  
<https://en.wikipedia.org/wiki/Endianness>