Bit Representation

Binary Representation: Review

- Binary representation of unsigned integers:
  $1011_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$
  $= 11_{10}$

Shifting bits

```
int a = 11; //1011
a >> 2; // shift a by 2 bits to the right
        // The digits pushed to the right disappear
        //  1011 -> 101 -> 10
```

- Shifting by 1 bit to the right is equivalent to dividing the number by 2 (using integer division)
- $\frac{1101_2}{10_2} = 110_2$ with remainder 1 $\frac{11_{10}}{2_{10}} = 5$ with remainder 1

```
a << 2 // shift a by 2 bits to the left
       // add 0's on the right
       // 1011-> 10110 -> 101100
```

- Shifting by 1 bit to the left is equivalent to multiplying the number by two
- $1101_2 \times 10_2 = 11010_2$ $11_{10} \times 2_{10} = 22_{10}$

Storing integers

- The smallest addressable unit of memory is usually a byte (8 bits)
- An integer can span multiple bytes (4 bytes) on most systems
- Two ways of storing the integer
  [byte1 byte2 byte3 byte4]

1024	byte1
1025	byte2
1026	byte3
1028	byte4

Big-endian

1024	byte4
1025	Byte3
1026	Byte2
1028	byte1

Little-endian

Big-endian vs Little-endian

- Mostly an arbitrary choice
- Name origin: a dispute over which side of the egg to break when eating it in *Gulliver's Travels*