Rust basics.

For those of you who are unfamilar with Rust, let us start with a few basics.

Rust is very similar to C but a little more formal.

Basic rust types

```
u8,u16,u32,u64,u128 - Unsigned integers - may not be negative 0-2^(n-1)

i8,i16,i32,i64,i128 - Signed integers   - May be negative.

usize               - The same size as a pointer (memory address)
                        Used to index arrays.

f32,f64             - Floating point numbers - may have decimals.

str                 - A sequence of bytes with a length that uses UTF8
& <xxx>             - A read-only reference (pointer) to <xxx>
```

Examples

```
let x : u8 = 1;          // A single byte with the value 1

let s : &str = "hello";
```

Note that `&str` actually has two parts - a pointer to the first byte and a length. This makes it 16 bytes on most machines.

Rationale

Memory access speed is everything in high performance code. The more bytes you read, the slower your program will run.

By using the smallest possible type, we can go a lot faster.