Rust variables & Mutability:

⇒ By default variables are immutable. ← once value is bound to a name, you cannot change that value

⇒ But we have the option to make the variables mutable

⇒ make variables mutable by adding `mut` infront of variable name.

Constants

⇒ Constants are also bound to a name & are not allowed to change.

⇒ Constants are always immutable by default and must be type annotated.

```
const HOURS_IN_SECONDS = 60 * 60 * 1
```

⇒ Naming convention for constants: Uppercase with underscore bet^n words.

Shadowing

⇒ We can declare a new variable with the same name as previous variable.

```
fn main() {
    let x = 5
    let x = x+3
}
```

This x shadows this variable.

- the first x binds to a value of 5
- it then creates a new variable adding 3, so the value of x becomes 8.

Data Types

⇒ Since Rust is statically typed language, it must know the types of all the variables at compile time.

```
let guess: u32 = "42".parse().expect("Not a no.")
```

type annotation

Scalar Types:

① Integer Types:→ without fractional component.

length	Signed	Unsigned
8-bit	i8	u8
16-bit	i16	u16
32-bit	i32	u32
64-bit	i64	u64
128-bit	i128	u128
arch	isize	usize

signed & unsigned refer to whether its possible for number to be negative

- signed (+|-)
- unsigned (only +)

Signed variant can store numbers from $-(2^{n-1})$ to $2^{n-1}-1$

So, i8 can store no. from $-(2^{8-1})$ to $2^{8-1}-1$

⇒ -128 to 127

Unsigned variants can store no. from 0 to $2^n - 1$

② Floating point types:

→ floating point numbers (f32 & f64) ← 32 bit ← 64 bit ↑ default

③ Boolean Type: (true/false)

④ Character Type: `let c = 'z'` alphabetic type

- char literals are specified with single quotes. String literals are specified with double quotes.
- char is 4-byte in size.

⑤ Compound Types:

- Tuples:
  - Grouping together a number of values with a variety of types into one compound type
  - They have fixed length: once declared, they cannot grow or shrink in size.

```
fn main() {
    let tup: (i32, f64, u8) = (500, 6.4, 1);
```

- Array Type:
  - collections of multiple values within the array.
  - every element in the array must have same type.