Randomness with rand crate

Learn to Code with Rust / Section Review

The **rand** Crate

- The **rand** library crate generates random values.
- Add the **rand** dependency to the project's **Cargo.toml** file with a version number.
- Rust will download and compile the crate alongside your executable.

The **random** Function

- The **random** function returns a random value.
- Provide the data type using either a variable annotation or the turbofish operator.
- The data type alters the constraints of the generated value.

The **rng** Function

- The **rng** function returns a **ThreadRng** struct which is a random number generator.
- An RNG is ideal when the program will be generating multiple random values.
- The RNG must be declared as mutable.
- The **random** function creates an RNG behind the scenes.
- Many of the RNG's methods require the **Rng** trait to be in the module's scope.

Methods on the ThreadRng Struct

- The **random** method returns a random value.
- The **random_range** method returns a value that falls within the specified range.
- The **random_bool** method returns a random Boolean. It accepts a float argument that represents the percentage chance of getting a **true**.
- The **random_ratio** method also returns a random Boolean. Its two arguments represent the proportional chance of getting a **true**.

Randomizing Vector Elements

- The **shuffle** method randomizes the order of elements in a collection type like a vector.
- Pass the method a mutable reference to a random number generator (**ThreadRng**).
- The **rand::seq::SliceRandom** trait must be brought into scope.