Datetimes with chrono Crate

Learn to Code with Rust / Section Review

The **chrono** Crate

- The **chrono** library crate is the community standard for working with datetimes.
- The word **chrono** means "relating to time".
- The **chrono-tz** crate adds various named timezones for easy timezone conversion.
- Rust will download and compile the crate alongside your executable.

The **Naive** Structs

- The **NaiveDate** struct models a calendar day (1999-09-23).
- The **NaiveTime** struct models a time of day (11:43:56).
- The **NaiveDateTime** struct models a specific moment in time (1999-09-23 11:43:56).

The **TimeDelta** Struct

- The **TimeDelta** models a duration, a span of time.
- The **num_days**, **num_hours**, and **num_minutes** methods return the count of each respective unit within the duration.
- TimeDeltas can be added or subtracted from datetimes, as well as from other TimeDeltas.
- Subtracting two datetimes from each other returns a **TimeDelta**.

The **DateTime** Struct

- The **DateTime** structs models a day + time with an associated timezone.
- The **Local** struct represents the current system datetime.
- The **Utc** struct represents the UTC datetime. UTC is a global standard for timezone comparisons.
- The **offset** method returns the timezone.

Changing Timezones

- Invoke the **with_timezone** method on a **DateTime** to model the same moment in a different timezone.
- Pass in a reference to a timezone (**&Utc**, **&Local**, or any value from the **chrono-tz** crate).

Converting between DateTimes and Strings

- The **parse_from_str** function converts a time string into a datetime.
- The **format** method converts a datetime into a time string.
- Both accept a format string which consists of symbols that designate various components of time (year, month, day, hour, timezone, etc).