Crate | regex

regex

- ***regex*** is a widely-used regular expression crate
- Supports Perl-style regular expressions
- Runs in linear time
  - No backreferences or lookarounds
- Can match on:
  - ASCII characters
  - Unicode
  - Bytes

Regular Expressions

```
^[a-z]{4}.*f\d[[:upper:]]$
```

Regular Expressions

- abcd 123 f5E

```
^[a-z]{4}.*f\d[[:upper:]]$
```

Creating a Regular Expression

```rust
use cached::proc_macro::cached;
use regex::Regex;

#[cached]
fn date_regex() -> Regex {
    // Matches ISO 8601 dates: 2021-02-19
    const re: &'static str = r"\d{4}-\d{2}-\d{2}";
    Regex::new(re).expect("compilation failure")
}
```

Example

```rust
let test_str = r#"
    today is 2021-02-17
    tomorrow is 2021-02-18
    yesterday was 2021-02-16
"#;
if date_regex().is_match(test_str) {⋯
}
if let Some(date) = date_regex().find(test_str) {⋯
}
for date in date_regex().find_iter(test_str) {⋯
}
```

Recap

- ***regex*** crate enables the usage of regular expressions
- Compiled regular expressions can be cached using the ***cached*** crate
  - Use the `#[cached]` macro to cache the expression
- When working with regular expressions:
  - ***is_match()*** determines if there is a match
  - ***find()*** finds the first match
  - ***find_iter()*** iterates over all matches
- Always write a test case to ensure the regular expression compiles properly