Discussion 11

Regular Expressions 🥳

anto [at] berkeley [dot] edu

All slides can be found on

`teaching.rouxl.es`

Announcements

- Scheme Project Phase 1 due today
- Phases 2 and 3 due Friday
- Final project due on Tuesday
- Would recommend getting the whole project done by the weekend
- I'm ending today's discussion slightly earlier so I can go to Dwinelle!
  - come with me pls thx

Notes from last section

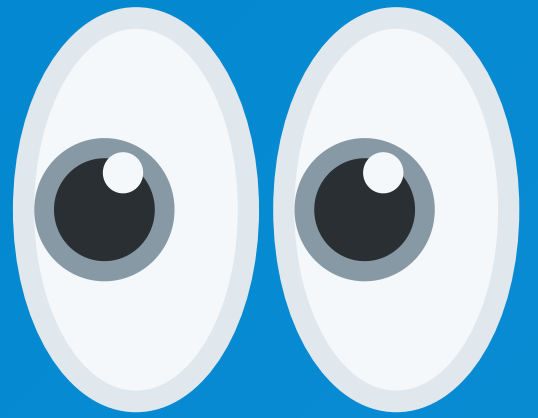
- Avid 3b1b watcher?
  - yep
  - got me through linalg/calc
- your explanation of tail calls/recursion was so good!! made me understand the lecture so much better
  - honestly a shocker because i thought i crapped the bed
- Python >> Scheme
  - i strongly agree

Notes from last section

- What notes app do you use on your ipad?
  - Goodnotes
  - Used to use OneNote/Microsoft Journal back when I used a surface/windows though
- In the game Pokemon, there is a tactic known as Fear. this is an acronym, standing for Focus sash, Endeavor, (quick) Attack, and Rattata. [...]
  - I personally liked the shell bell/aron/sandstorm thing a lot more just cause I thought it was funnier and could potentially last longer
  - There was also a FEAR reference in the Fall 2022 Final
  - fun question

Temperature Check 🌡️

- RegEx

RegEx 

Poll: How do we all pronounce it?

RegEx

- **Reg**ular **Ex**pressions are used to describe sets of strings that meet certain criteria
- Very useful for pattern matching (will see examples later)
  - Used in real life all the time!
  - Can be used for super simple patterns (`egg` in text)
  - Can also be very complicated (validating passwords)
- regex101.com
  - Super useful for learning/debugging your regular expressions, but don't rely on this!
  - (You won't have this resource for the exam! Don't use it as a GPS)

Character Classes

- If you want to search for a certain character in a set of characters, use character classes
- Can use pre-defined classes, or specify your own

Specified:

Pattern	Description
`[abc]`	Matches either a, b, or c
`[a-z]`	Matches anything from lowercase a to lowercase z
`[^a-z]`	Matches anything that isn't from lowercase a to lowercase z
`[0-9]`	Matches any (single) digit
`[a-zA-Z123]`	Matches any lowercase or uppercase character, or the digits 1, 2, or 3

Character Classes

Predefined character classes

Pattern	Equivalent
`\w`	`[A-Za-z0-9_]`
`\d`	`[0-9]`
`\s`	(Matches any 'whitespace' character)
`.`	Matches everything except for newlines.

Combining Multiple Patterns

- If you put a pattern after another pattern, you can have a regular expression that matches both of those characters in succession
  - `AB` - Matches A, then B (both are required)
  - `A[xt]e` Matches A, then either x or t, then E
    - Axe or Ate are the things that get matched
- `|` is the separator used to match 'or'
  - `A|B` - Matches either A or B

Quantifiers

Quantifier	Description
`?`	0 or 1 (kind of like a question - is it there or not?)
`*`	0 or more
`+`	1 or more
`{a}`	Match exactly `a` times (number)
`{a,}`	Match from `a` to infinity number of times
`{0,b}`	Match from `0` to `b` number of times
`{a,b}`	Match between `a` and `b` number of times

Groups

- Parentheses are used to group certain patterns together
  - Useful with quantifiers
  - `(Ha)+` and `Ha+` do different things!

Anchors

Anchor	Description
`^`	Matches the *beginning* of the string
`$`	Matches the *end* of a string
`\b`	Matches a *word boundary*

Special Characters

Some characters cannot be written directly because they have special roles in RegEx

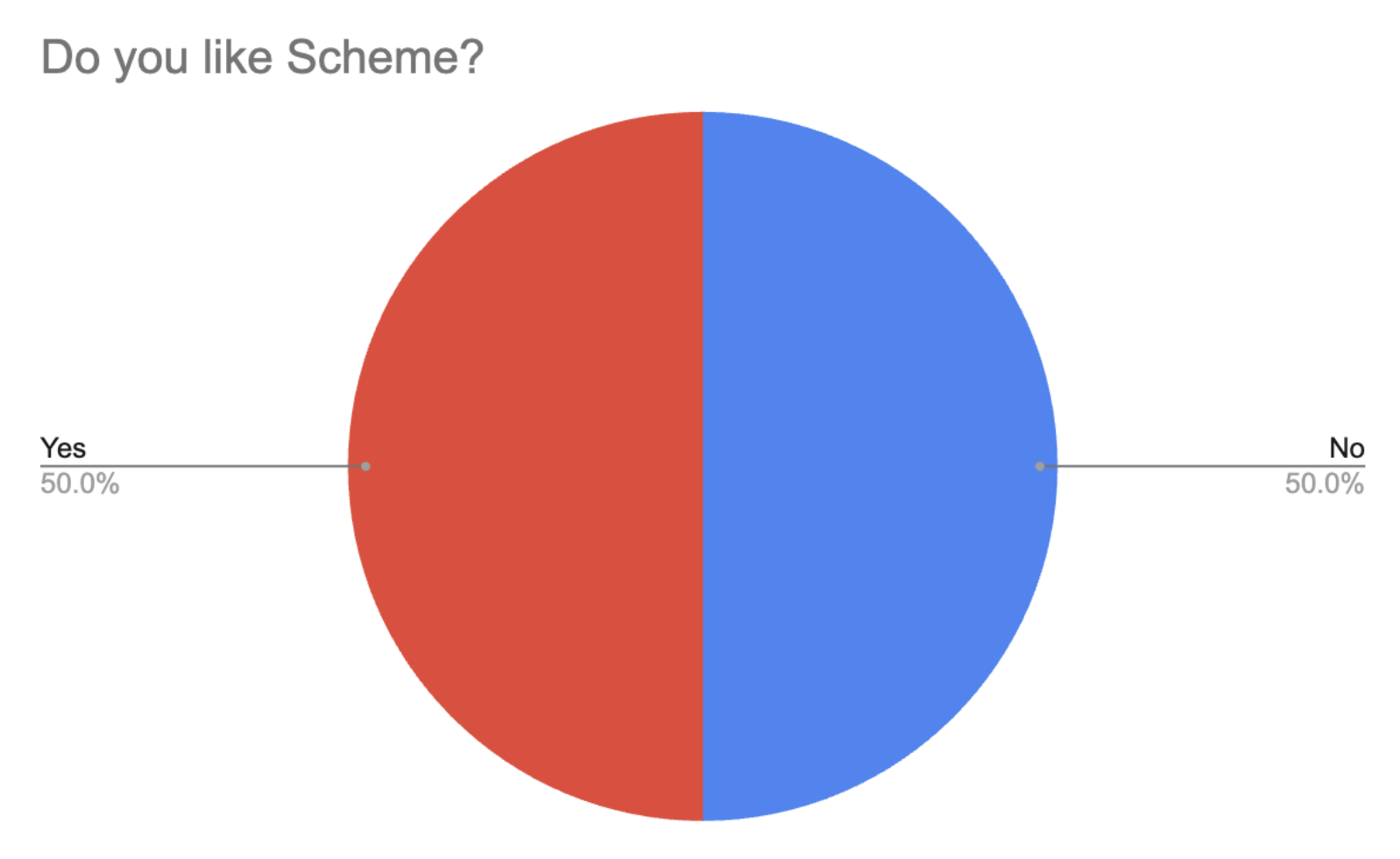
- `\[](){}+*?|$^.`
- Inside a character class, you need to escape `-`
- To escape these characters, you need to put a backslash before them
  - `The quick brown fox jumped over the lazy dog\.`

RegEx in Python

- For Python, we use *r-strings* to write regex patterns
- With *r-strings* in Python, backslashes are treated specially in some cases (for example with `\n`, or `\t`). What *r-strings* do in this case is that they automatically escape the backslashes for these special characters, so they look like real backslashes rather than what Python designates for those characters.
- `r"<insert string here>"` - the `r` at the front makes a string an *r-string*
  - Python also has *f-strings* which are pretty useful/interesting

Worksheet!

Results from last section (`links.rouxl.es/disc`)



Mental Health Resources

- CAPS:
  - If you need to talk to a professional, please call CAPS at 510-642-9494.
- After Hours Assistance
  - For any assistance after hours, details on what to do can be found at this link

Anonymous Feedback Form

`links.rouxl.es/feedback`

Thanks for coming! 🥳

Please give me feedback on what to improve!