Vim in your IDE

Our mouse is the essential tool for manipulating things on our computer, and our keyboard lets us input text. One advantage of the mouse is that it has a low learning curve: the screen shows us what we can do, and the mouse lets us do those things.

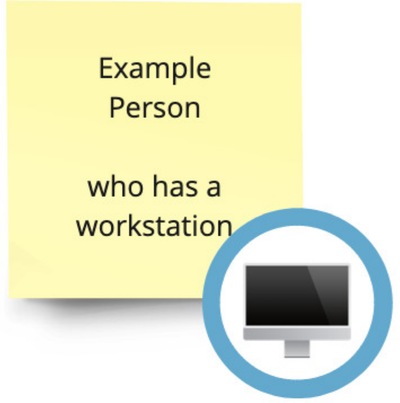
Vim provides a secondary tool for manipulating things on our computer. It's faster and more powerful than the mouse, and unlike an LLM its inputs are terse and latency free. If you'd like to learn the commands, you can more effectively manipulate text than you could with a mouse and keyboard.

In this session, we learn to:

- use the Vim extension in VS Code to move around code faster than we could with a mouse
  - toggle between Vim mode (Vim NORMAL) and regular IDE mode (Vim INSERT)
  - translate high level navigation (like "start writing on line 35 ½") into Vim commands to quickly drive to the place you're navigated
  -

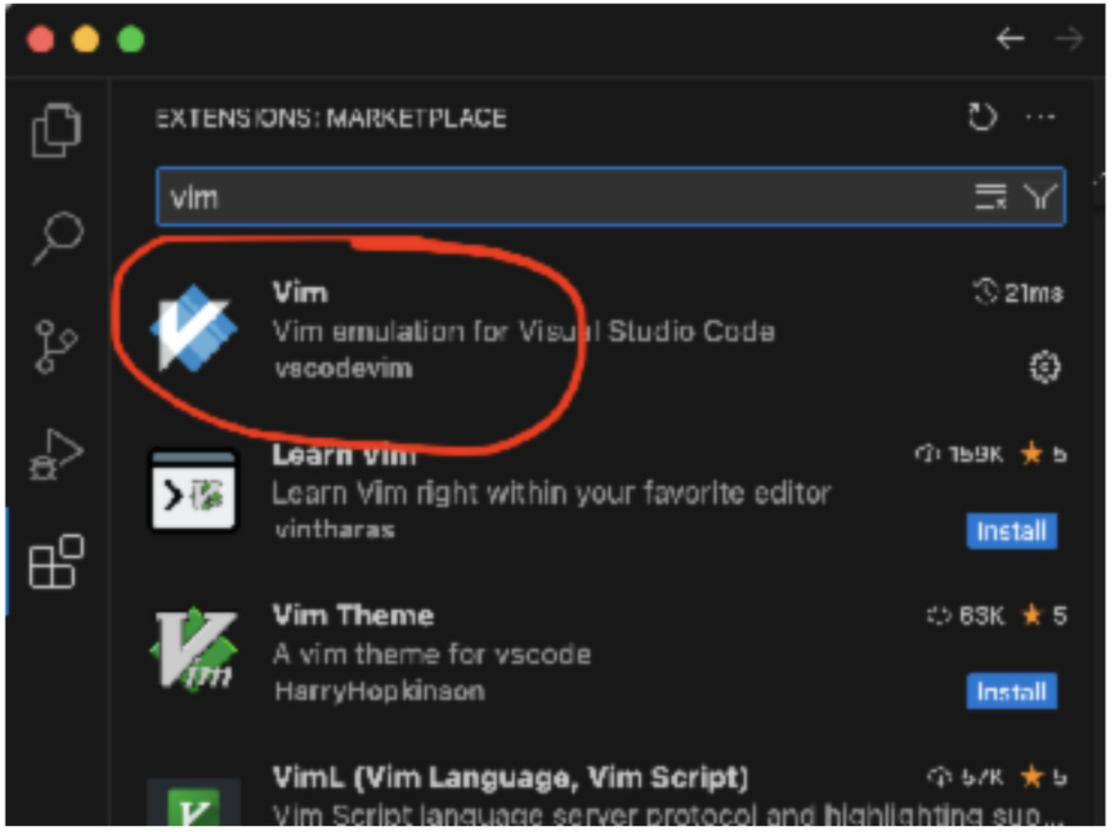
Place a card with your name on it

If your team workstation is available to use, please add a computer emoji to your card



Setup your laptop/workstation for the exercises

1. Install VS Code
2. Install Vim Extension in VS Code
    a. Ctrl+Shift+P
    b. Extensions: Install Extensions
    c. search: vim
3. Download and open desserts.ts



After our session, if you're not keen to use the Vim extension you can remove it or disable it 👍

What is the keyboard shortcut you use most often?
(In any application)

What interests you about Vim?

Have you avoided Vim in the past? How come?

Vim vs Mouse

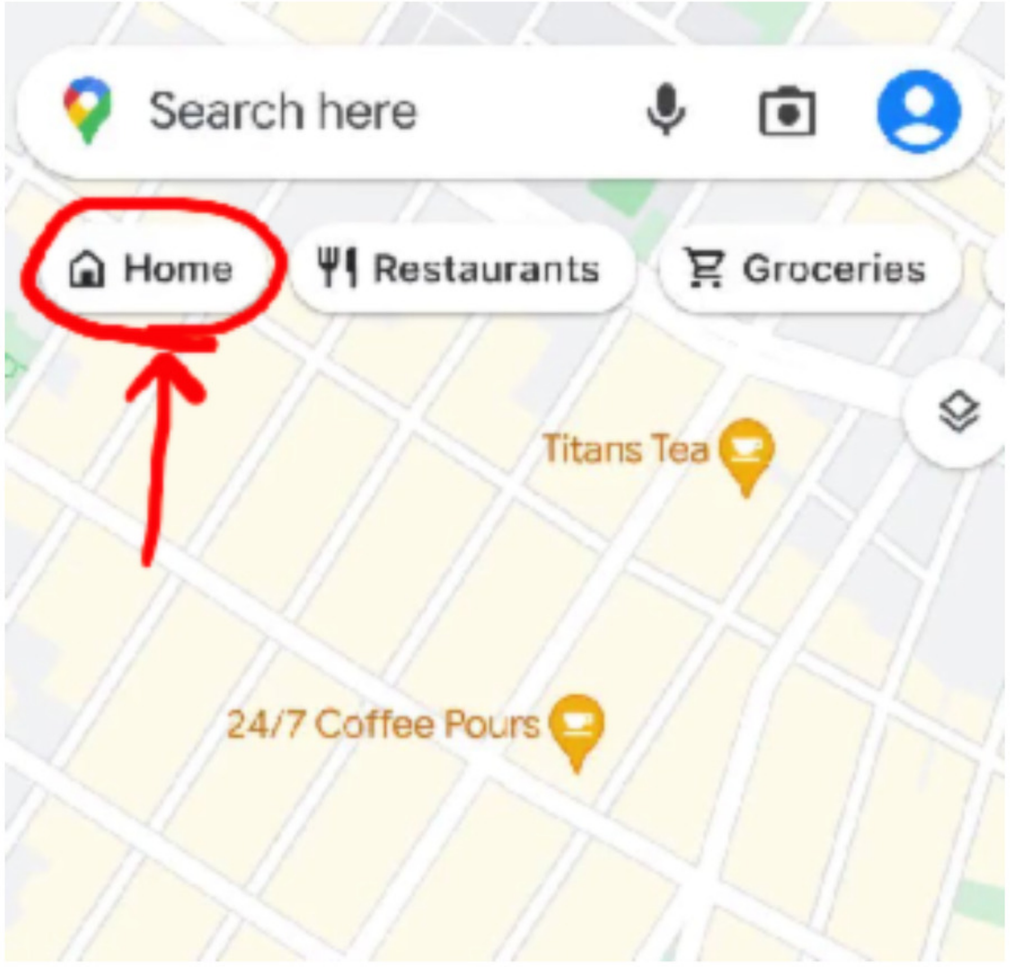
- intuitive
- good for discovering new actions
- more open ended

- keyboard shortcuts on steroids
- within it's scope, it's faster
- gestures are more repeatable

```
VIM - Vi IMproved

version 9.0.1544
by Bram Moolenaar et al.
Vim is open source and freely distributable
```

How do we go "home"?



Before setting out on an adventure, we make sure we'll have a way to get back home

NORMAL = new to us; INSERT = familiar

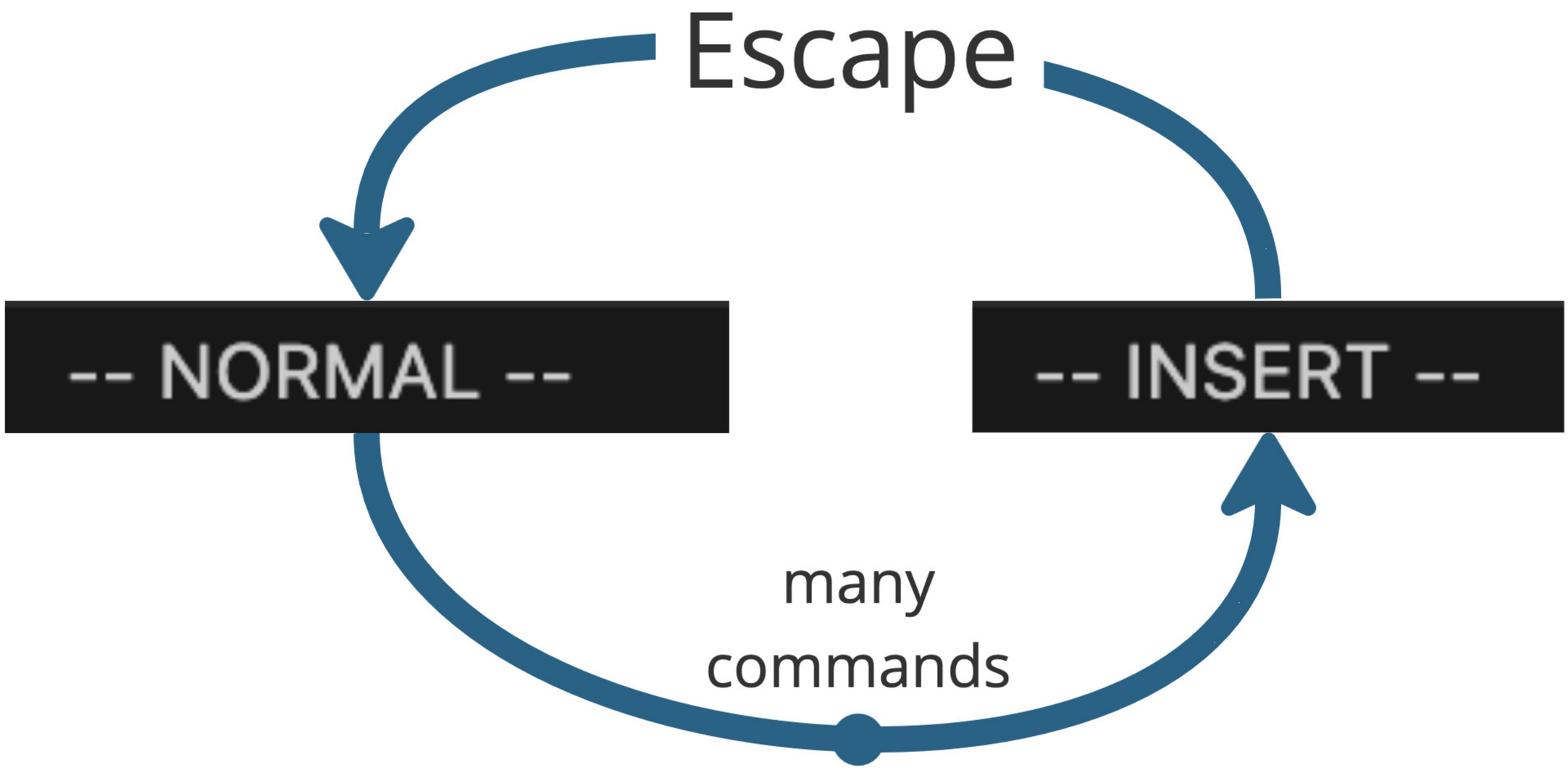
- this is the default mode in VIM
- our keyboard becomes a navigation device (a "second mouse")
- the new stuff we'll learn today takes place in Normal mode

-- NORMAL --

- in Insert mode, our IDE behaves the way we're used to
- in this mode, VS Code should behave in the way we're used to

-- INSERT --

Changing Modes



Changing Modes

Commands for entering INSERT Mode

hidden

line below the cursor

mode left of the cursor

mode *a*fter the cursor

e current line

line above the cursor

mode at the start of the line

In pairs, match the command with the behavior.

Use Escape to go back to Normal mode after each command.

Use your arrow keys or your mouse to move around the code

Changing Modes

Commands for entering INSERT Mode

`o` `O` `i` `I` `a` `A`

- begin a new line below the cursor
- enter *i*nsert mode left of the cursor
- enter insert mode *a*fter the cursor
- *a*ppend to the current line
- begin a new line above the cursor
- enter *i*nsert mode at the start of the line

In pairs, match the command with the behavior.

Use Escape to go back to Normal mode after each command.

Use your arrow keys or your mouse to move around the code

Moving Around

Note: <cr> means Enter

VIM	Navigation
`G`	go to the bottom of the file
`gg`	go to the top of the file
`:33<cr>`	on line 33
`:12<cr>`	just before line 12
`''`	go back to where we were before
`:22<cr>A`	at the end of line 22
`:23<cr>I`	at the start of line 23
`/blue<cr>`	go to the next `blue`. Press `n` to skip to subsequent matches
`gg/fish <cr>`	go to where we initialize `fish`
`gg/class <cr>`	go to the start of the class
`e`	after this word
`b`	before this word
`f)i`	just inside the close parenthesis (try this on the `Math.floor` line)

In pairs, take turns **navigating each other** through each of these commands.

After one person has **tried each of these** commands, **switch roles and repeat**.

Navigators: first, give the high level navigation. Then, if the driver needs it, give the low level Vim command.

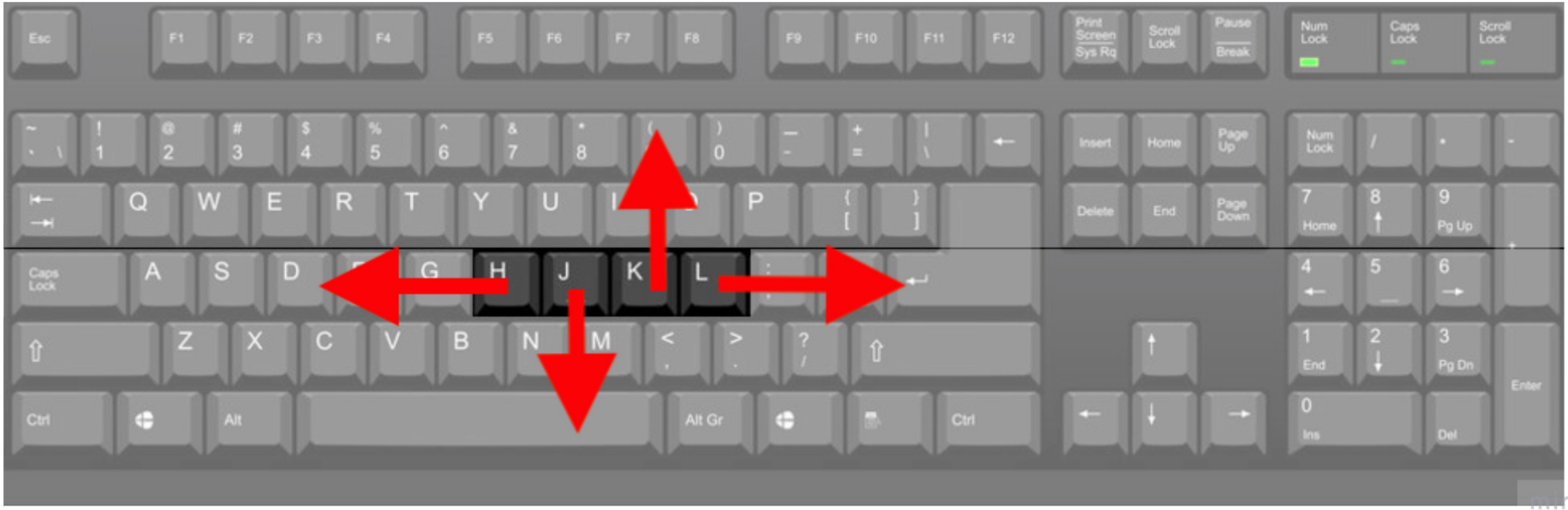
If you have extra time, repeat the exercise until the navigator can give only the high level navigation

Beyond Arrow Keys

Arrow keys are closer than the mouse, but still require moving away from our home row typing position.

Arrows from Home Row

Use h, j, k, and l to move the cursor
left, down, up, and right



Jump around

Commands to jump to a nearby word or letter

- **/word<cr>** to jump to the next occurrence of word
    - **n** to jump to the next match
- **/ord<cr>** if you don't know if W is capitalized
- **fx** to jump to the first **x** character on the line
- **e** to jump to the end of the current word
- **b** to jump to the beginning of the current word
- **%** to jump to the matching brace, bracket, or paren

Repeat Commands

Put a number before a command to do it **n** times

- **5j** to go down 5 lines
- **3e** to go the the end of the 3rd word
- likewise for h, k, l, b, and others

All The Other Commands

vim jump to

- vim jump to **end of file**
- vim jump to **line**
- vim jump to **end of line**
- vim jump to **column**
- vim jump to **function definition**
- vim jump to **line number**
- vim jump to **matching bracket**
- vim jump to **start of line**
- vim jump to **beginning of file**
- vim jump to **previous location**

Vim Tag

In pairs, chase each other around dessert.js

1. Choose someone to be "it"
2. The person who is "it" describes a place in the file
3. The other person must get there as quickly as possible and insert *HERE* at the place that's described
4. If the *HERE* is in the right spot, then swap who is "it" and repeat

Vim Tag

To make things more interesting when you're "it":

- Bingo: look back at the cheat sheet of commands and look for ones your pair hasn't used yet. Can you **get them to use each of those commands**?
- Taboo: try to get your pair to use a command **without directly stating it**. For example, for *ggA*, instead of saying "at the end of the first line", try "after the curly brace in Dessert interface"

Learning more Vim

1. practice navigating code without your mouse during your work
2. daily curiosity: web search "how to ___ in Vim"
3. subscribe to an email newsletter like VimTricks
4. attend the next department Vim training

Go learn something new

jump to next word in vim

find word under cursor in vim

go to matching parenthesis in vim

favorite vim command

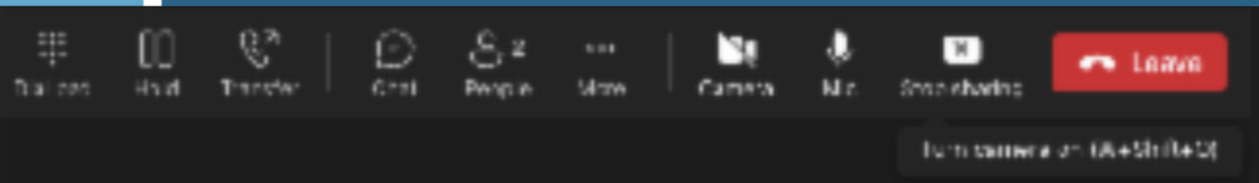
most obscure vim command

Write it down

Reply to this Teams post with:

1. the command you found
2. what it does

What's next?



It's 8:20
Standup is over
Everyone's on the teams call
What will you do??

AB

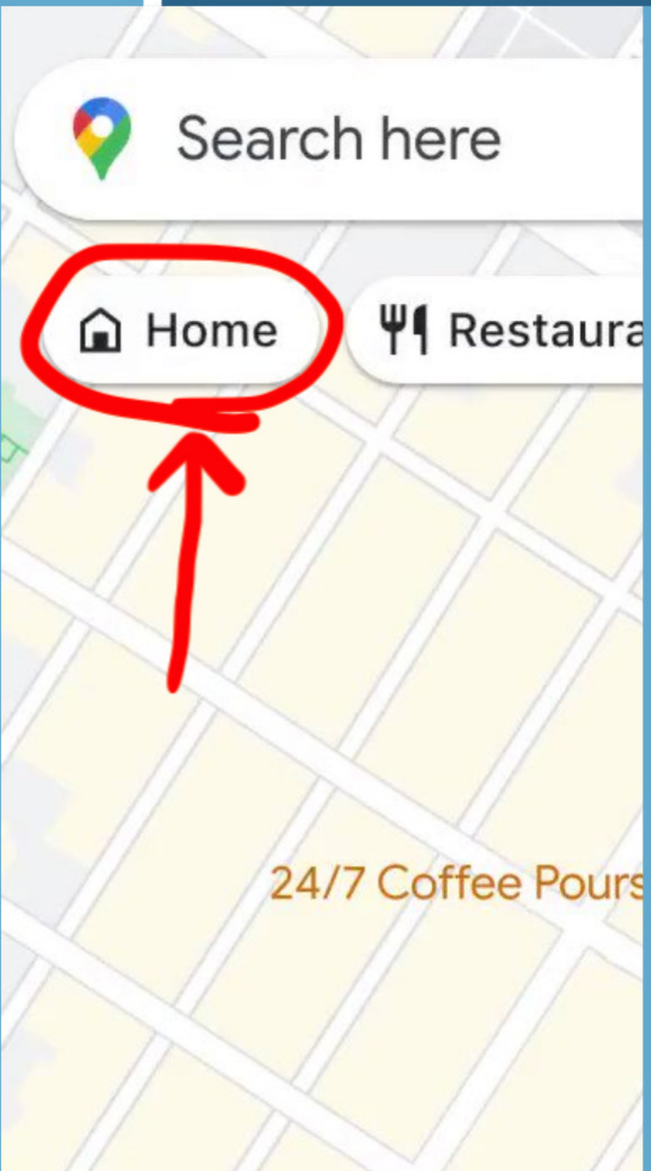
miro

Using Vim Today: a script for sharing with your team

- "**I've been learning** how to use the **Vim extension** to **speed up my code navigation**"
- "I'd like to **install it** to use **while I drive**"
- "I'll show you the **command to disable it** so **you don't have to use it**"
- "Also, if you'd like, I could **show you the basics** in one mob rotation"
- "**Would that be OK?**"

Disabling the Vim extension

how to "go home" to familiar IDE behavior



-- NORMAL --

Ctrl + Shift + P

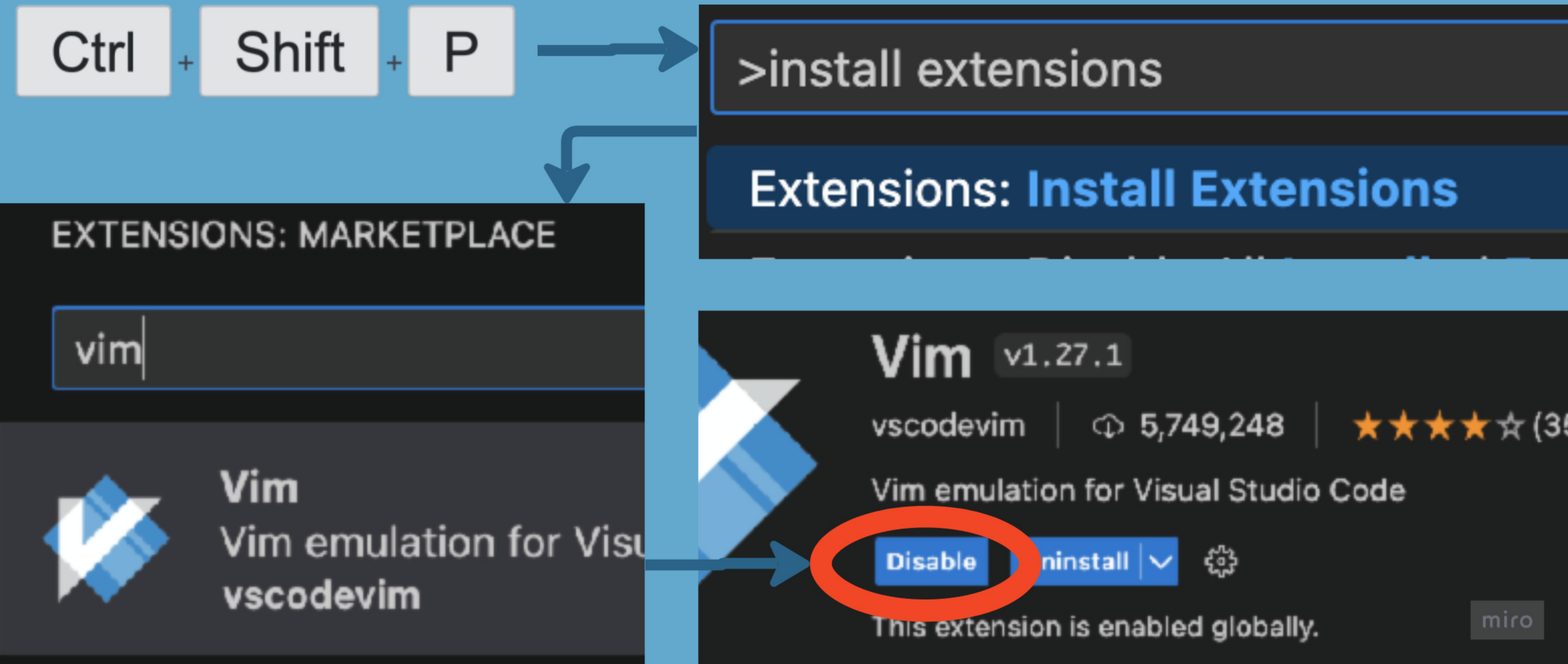
>toggle vim

Vim: Toggle Vim Mode recently used

-- VIM: DISABLED --

Disabling the Vim extension even more

how to fully disable or remove the Vim extension



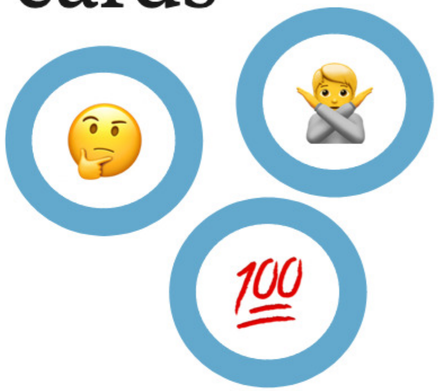
Using Vim Today: if your team is keen

the basics in one mob rotation

1. **Use *i* to make the IDE behave like it usually does** (Vim "INSERT" mode)
2. **Use *Escape* to use Vim commands** (Vim "NORMAL mode)
3. Use *h j k l* instead of arrow keys. Repeat them with e.g. *22j*
4. Use */search* to jump to the first instance of "search"
    a. Use *n* to go to the next instance of "search"
5. Use *:42* to jump to line 42
6. Use *o* to enter INSERT mode on a new line
7. There are many more commands we'll learn over time

Reflection

1. Reflect on this training by adding stickies to the cards
2. Clone one of these emoji and place them on the statement below to show your reaction.



Today, I could start using VS Code in Vim

I'm keen to start using...

I don't want to forget...

I want to learn more about...