CS11-711 Advanced NLP

Agents

Sean Welleck

Carnegie Mellon University
Language Technologies Institute

https://cmu-l3.github.io/anlp-spring2025/

https://github.com/cmu-l3/anlp-spring2025-code

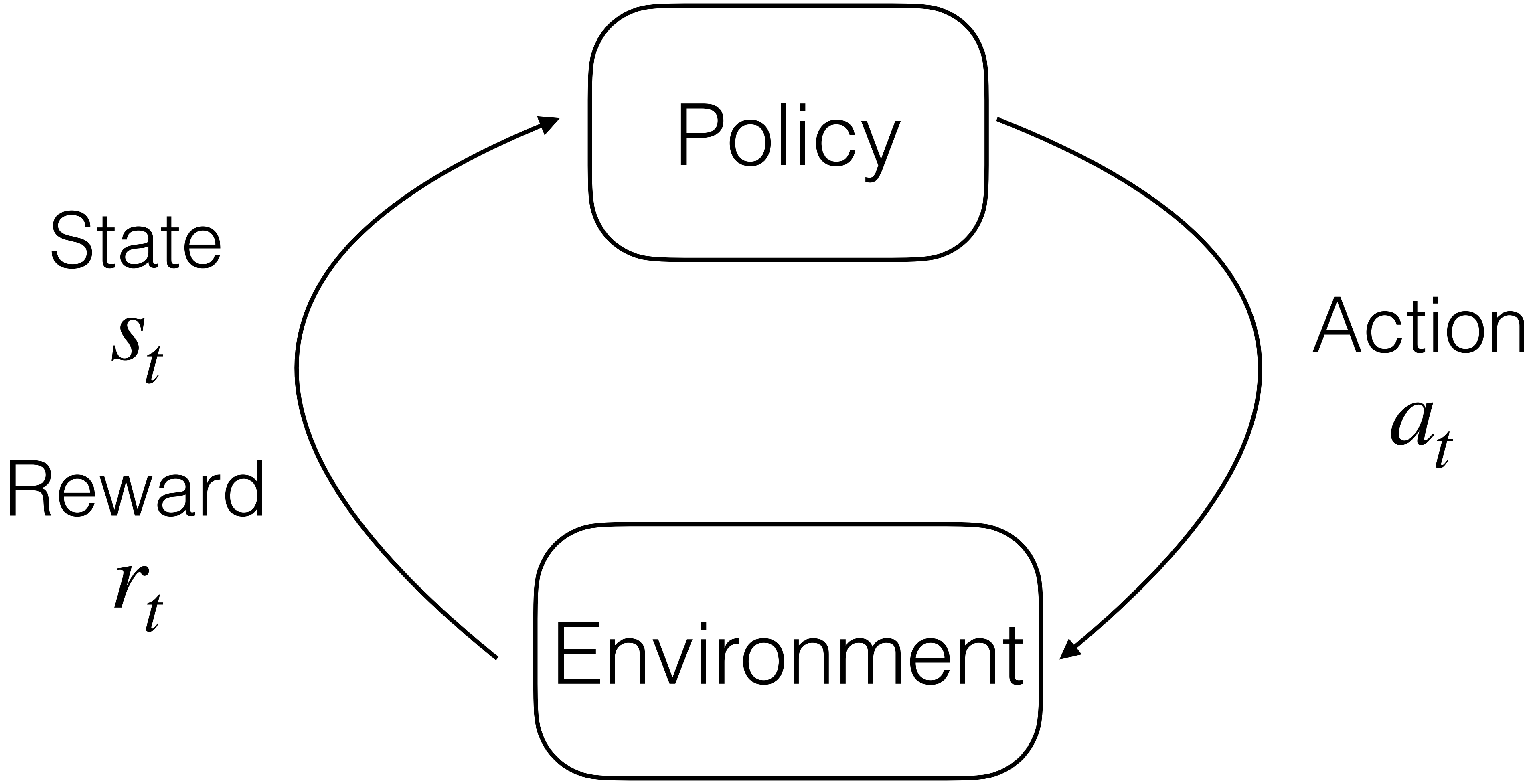
Recap: language models

Input text

LM

Output text

Recap: RL MDP



Policy

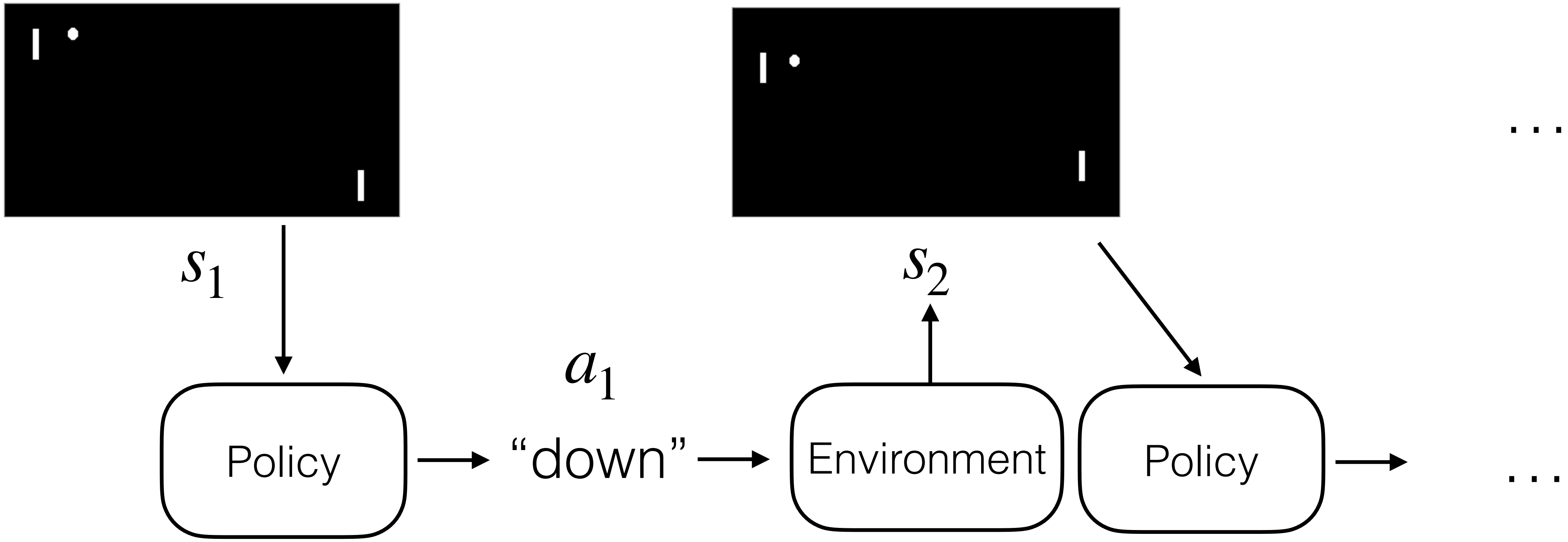
State s_t

Reward r_t

Action a_t

Environment

Recap: RL MDP



s_1

Policy

a_1

"down"

Environment

s_2

Policy

$\ldots$

$\ldots$

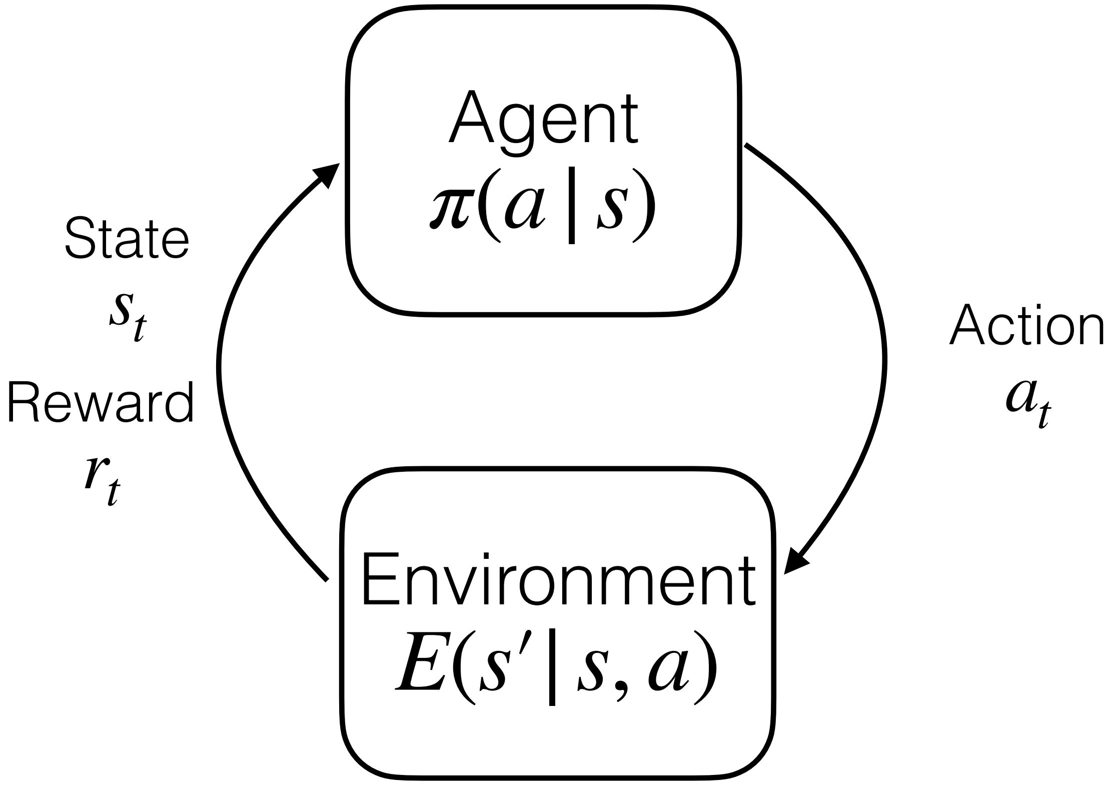
Today: (LM) Agents

- What is an agent?
- Agent environments
- Agent patterns

Code examples: https://github.com/cmu-l3/anlp-spring2025-code/tree/main/14_agents

What is an Agent?

- An agent is simply a policy!
  - Input: state
  - Output: action
- Operates in an environment



Agent

$\pi(a \mid s)$

Environment

$E(s' \mid s, a)$

State

s_t

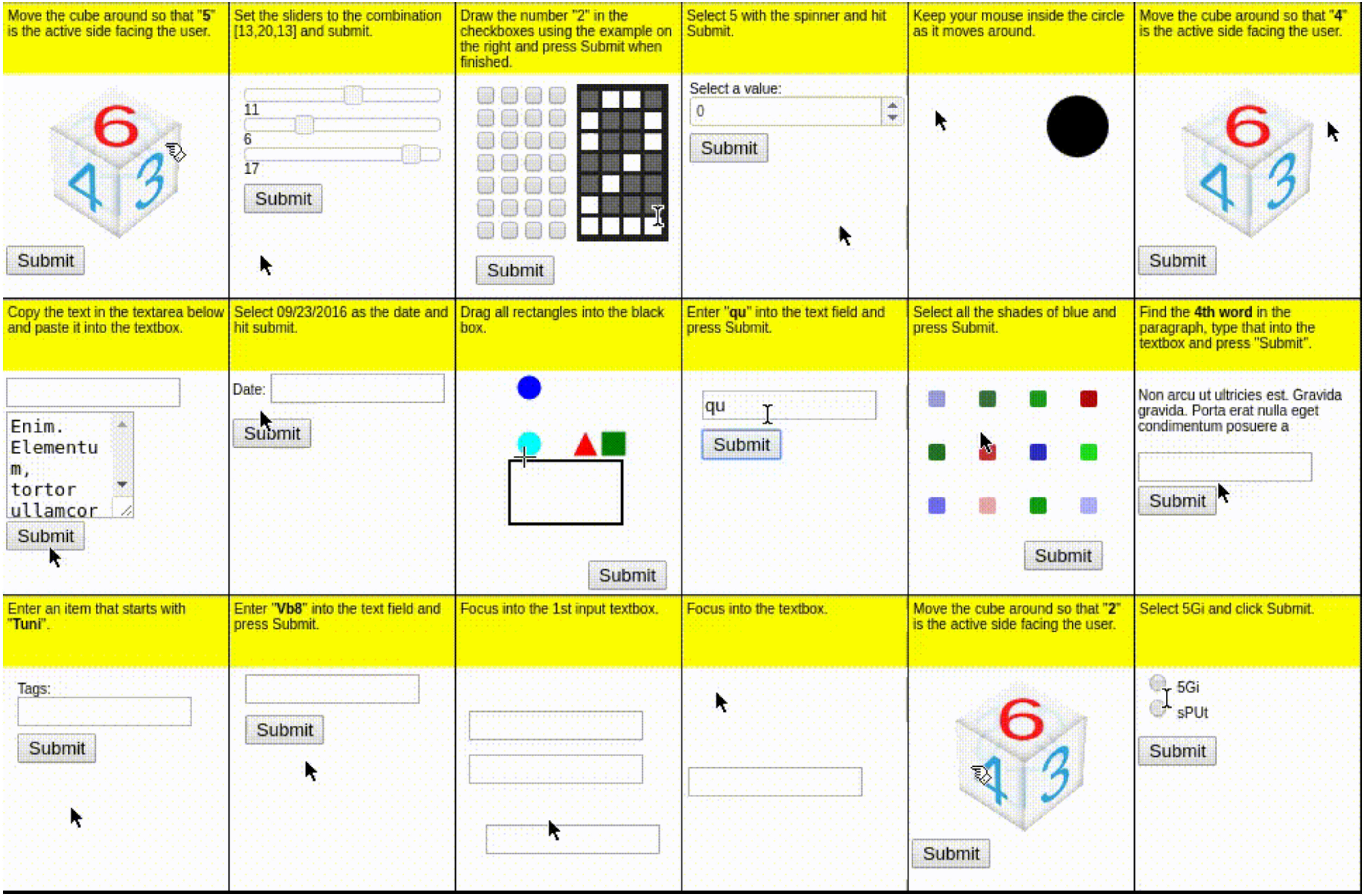
Reward

r_t

Action

a_t

Example: simple web environments



Move the cube around so that "**5**" is the active side facing the user.

Set the sliders to the combination [13,20,13] and submit.

Draw the number "2" in the checkboxes using the example on the right and press Submit when finished.

Select 5 with the spinner and hit Submit.

Keep your mouse inside the circle as it moves around.

Move the cube around so that "**4**" is the active side facing the user.

Copy the text in the textarea below and paste it into the textbox.

Enim. Elementum, tortor ullamcor

Select 09/23/2016 as the date and hit submit.

Drag all rectangles into the black box.

Enter "**qu**" into the text field and press Submit.

qu

Select all the shades of blue and press Submit.

Find the **4th word** in the paragraph, type that into the textbox and press "Submit".

Non arcu ut ultricies est. Gravida gravida. Porta erat nulla eget condimentum posuere a

Enter an item that starts with "**Tuni**".

Enter "**Vb8**" into the text field and press Submit.

Focus into the 1st input textbox.

Focus into the textbox.

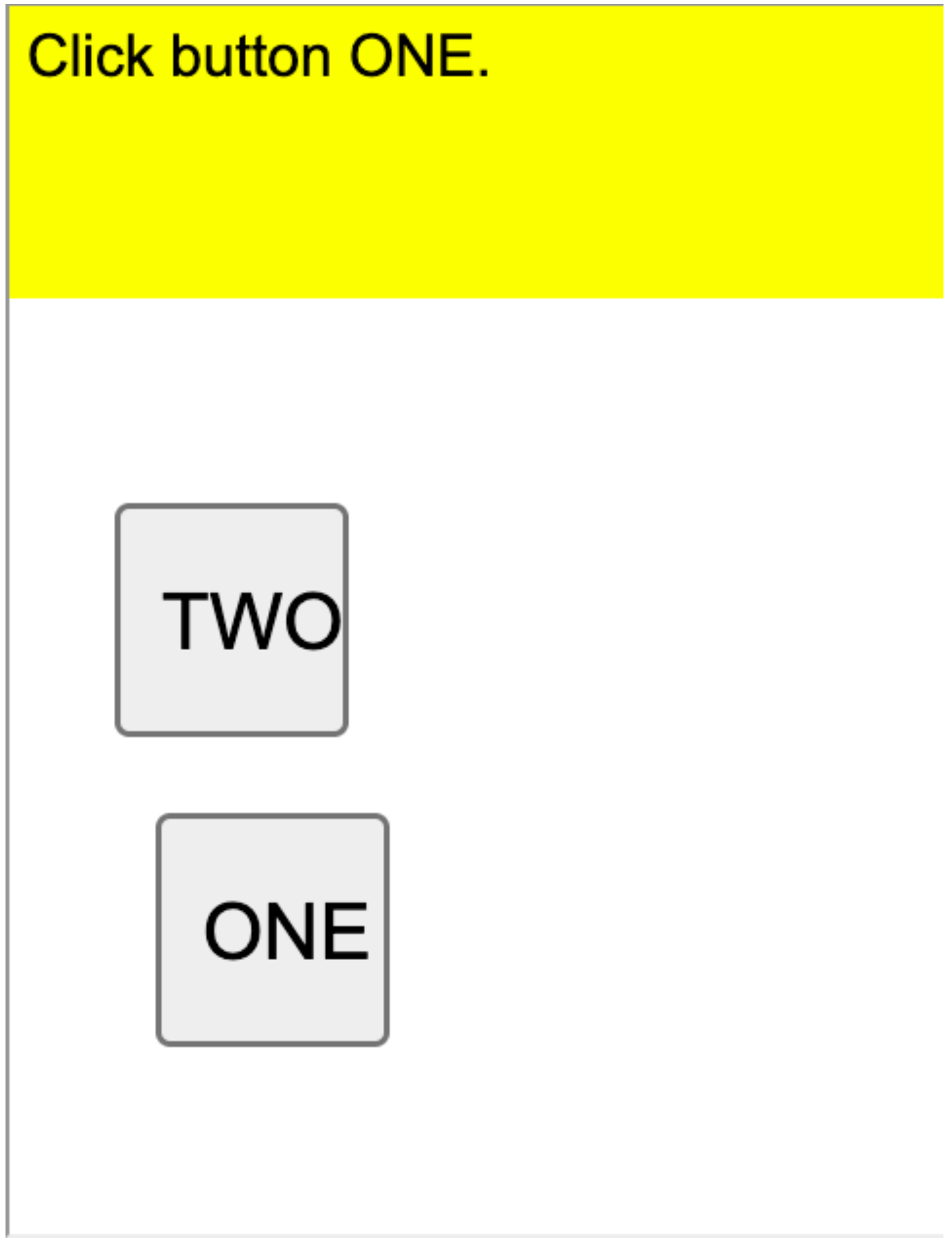
Move the cube around so that "**2**" is the active side facing the user.

Select 5Gi and click Submit.

https://miniwob.farama.org/

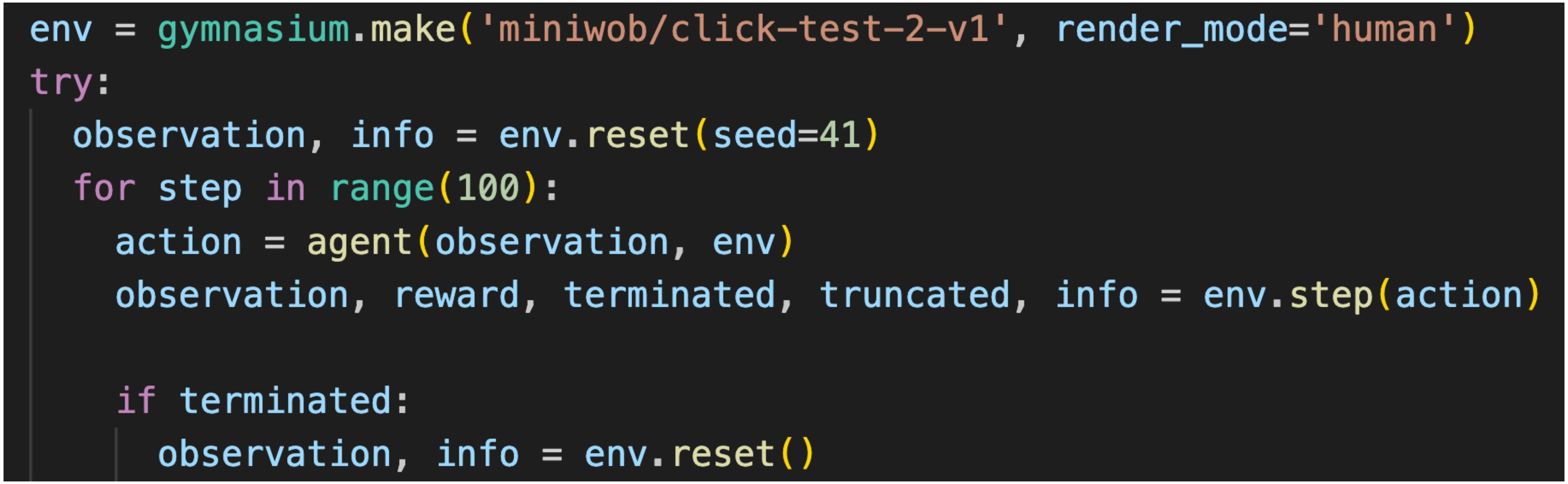
Example: click environment

- Environment:
  - Simple webpage
- Task:
  - Click indicated button



Click button ONE.

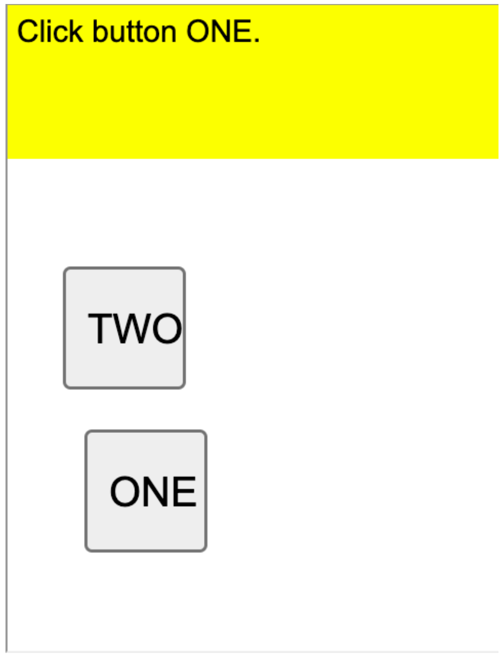
Example: click environment

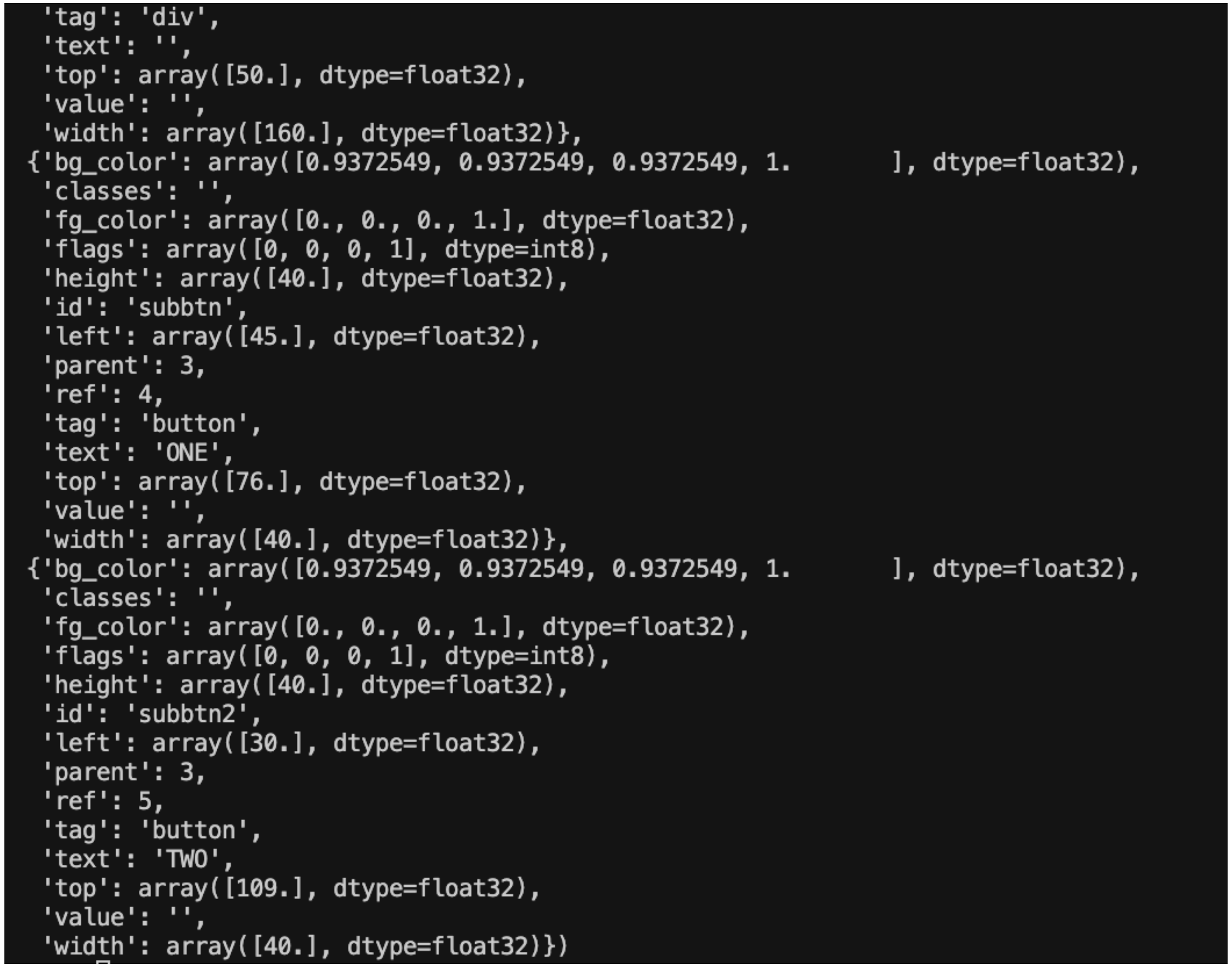


```
env = gymnasium.make('miniwob/click-test-2-v1', render_mode='human')
try:
  observation, info = env.reset(seed=41)
  for step in range(100):
    action = agent(observation, env)
    observation, reward, terminated, truncated, info = env.step(action)

    if terminated:
      observation, info = env.reset()
```

Example: click environment

- Observation/state
  - Image
  - DOM elements

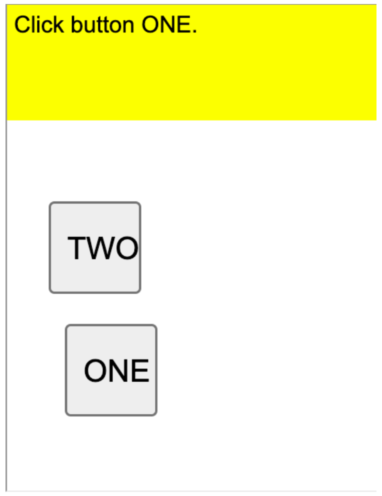

Click button ONE.


```
 'tag': 'div',
 'text': '',
 'top': array([50.], dtype=float32),
 'value': '',
 'width': array([160.], dtype=float32)},
{'bg_color': array([0.9372549, 0.9372549, 0.9372549, 1.       ], dtype=float32),
 'classes': '',
 'fg_color': array([0., 0., 0., 1.], dtype=float32),
 'flags': array([0, 0, 0, 1], dtype=int8),
 'height': array([40.], dtype=float32),
 'id': 'subbtn',
 'left': array([45.], dtype=float32),
 'parent': 3,
 'ref': 4,
 'tag': 'button',
 'text': 'ONE',
 'top': array([76.], dtype=float32),
 'value': '',
 'width': array([40.], dtype=float32)},
{'bg_color': array([0.9372549, 0.9372549, 0.9372549, 1.       ], dtype=float32),
 'classes': '',
 'fg_color': array([0., 0., 0., 1.], dtype=float32),
 'flags': array([0, 0, 0, 1], dtype=int8),
 'height': array([40.], dtype=float32),
 'id': 'subbtn2',
 'left': array([30.], dtype=float32),
 'parent': 3,
 'ref': 5,
 'tag': 'button',
 'text': 'TWO',
 'top': array([109.], dtype=float32),
 'value': '',
 'width': array([40.], dtype=float32)})
```

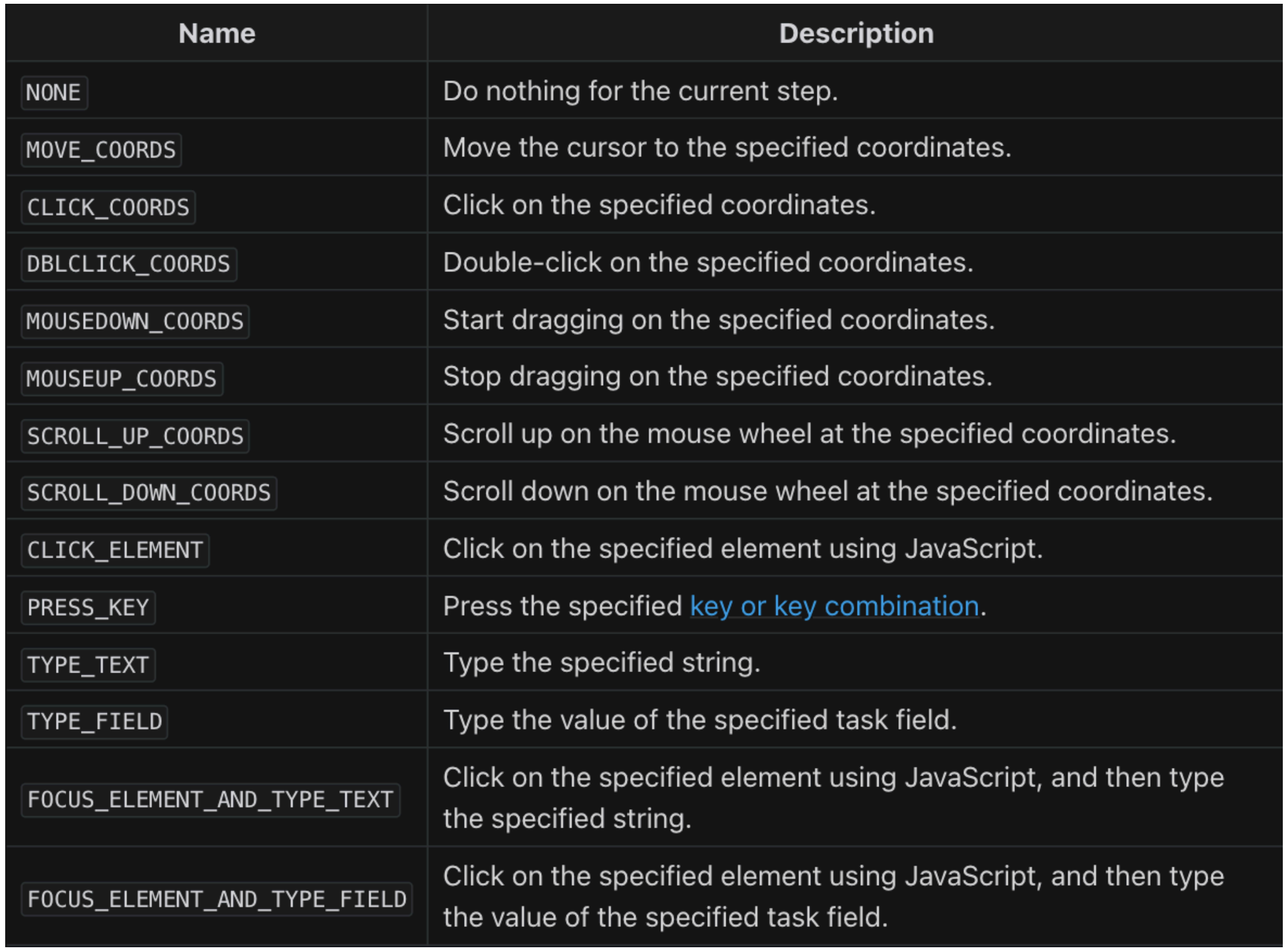
Example: click environment

- Actions

  - MOVE_COORDS <x, y>, … CLICK_ELEMENT <REF>, …



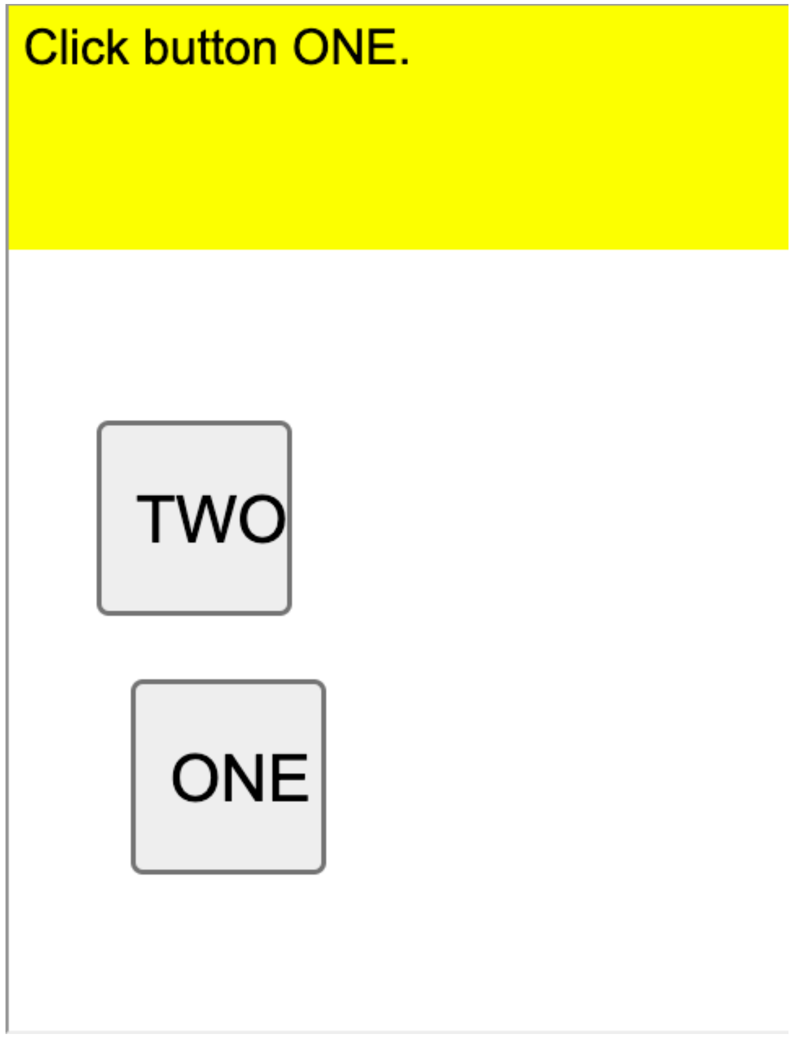
Click button ONE.



Name	Description
`NONE`	Do nothing for the current step.
`MOVE_COORDS`	Move the cursor to the specified coordinates.
`CLICK_COORDS`	Click on the specified coordinates.
`DBLCLICK_COORDS`	Double-click on the specified coordinates.
`MOUSEDOWN_COORDS`	Start dragging on the specified coordinates.
`MOUSEUP_COORDS`	Stop dragging on the specified coordinates.
`SCROLL_UP_COORDS`	Scroll up on the mouse wheel at the specified coordinates.
`SCROLL_DOWN_COORDS`	Scroll down on the mouse wheel at the specified coordinates.
`CLICK_ELEMENT`	Click on the specified element using JavaScript.
`PRESS_KEY`	Press the specified key or key combination.
`TYPE_TEXT`	Type the specified string.
`TYPE_FIELD`	Type the value of the specified task field.
`FOCUS_ELEMENT_AND_TYPE_TEXT`	Click on the specified element using JavaScript, and then type the specified string.
`FOCUS_ELEMENT_AND_TYPE_FIELD`	Click on the specified element using JavaScript, and then type the value of the specified task field.

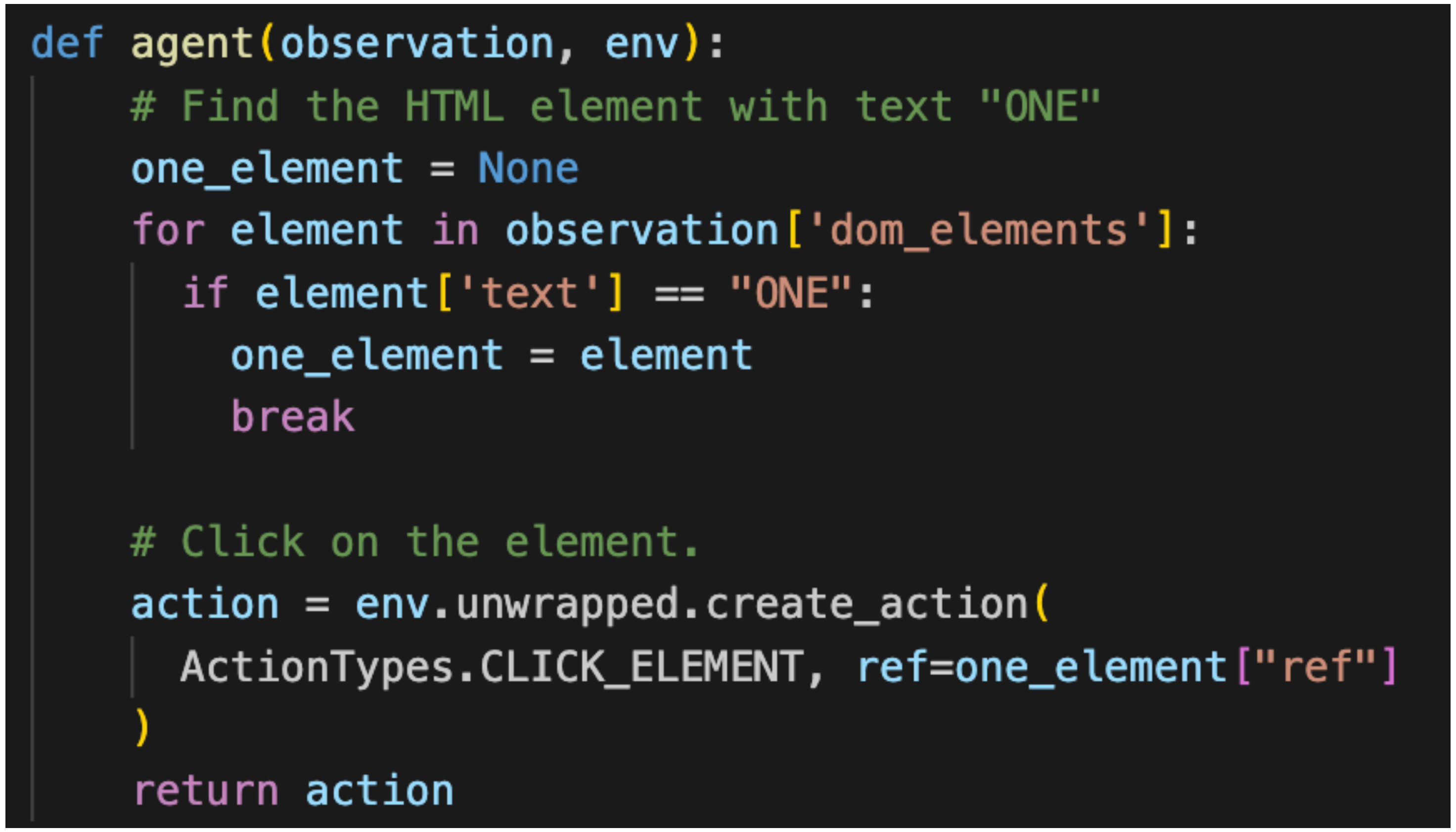
A simple rule-based agent

- Write a program that:
  - Finds the DOM element with text ‘ONE’
  - Get its REF
  - Output the action CLICK_ELEMENT <REF>

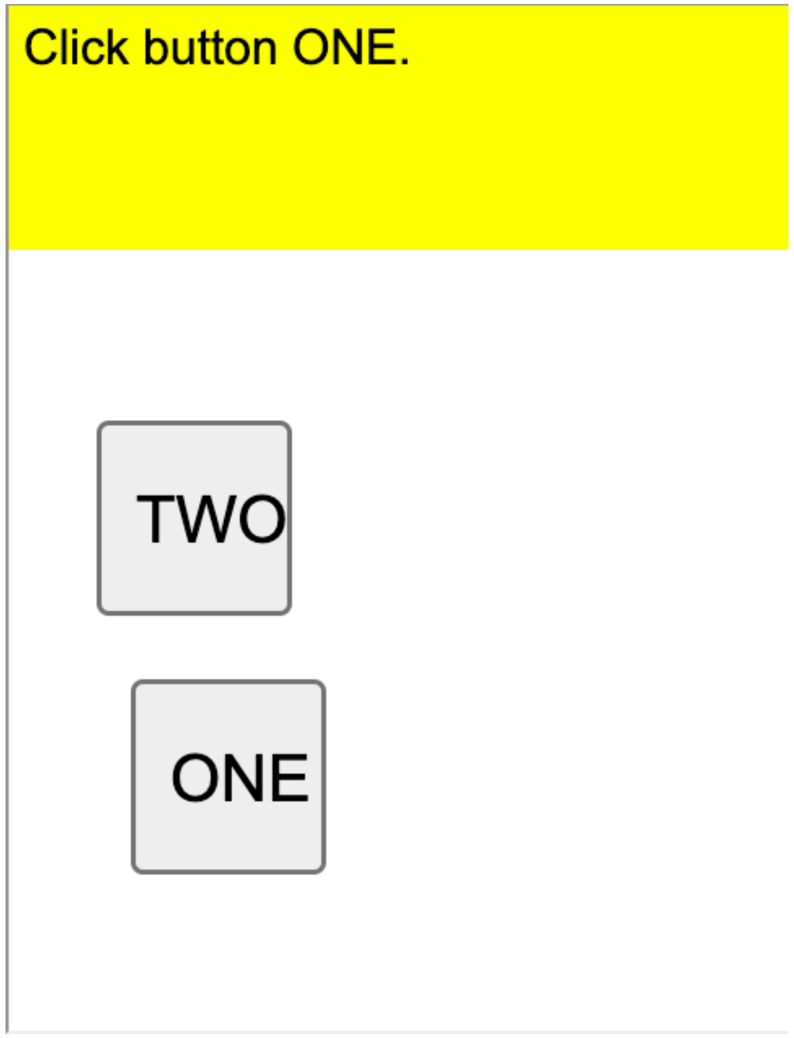


Click button ONE.

A simple rule-based agent



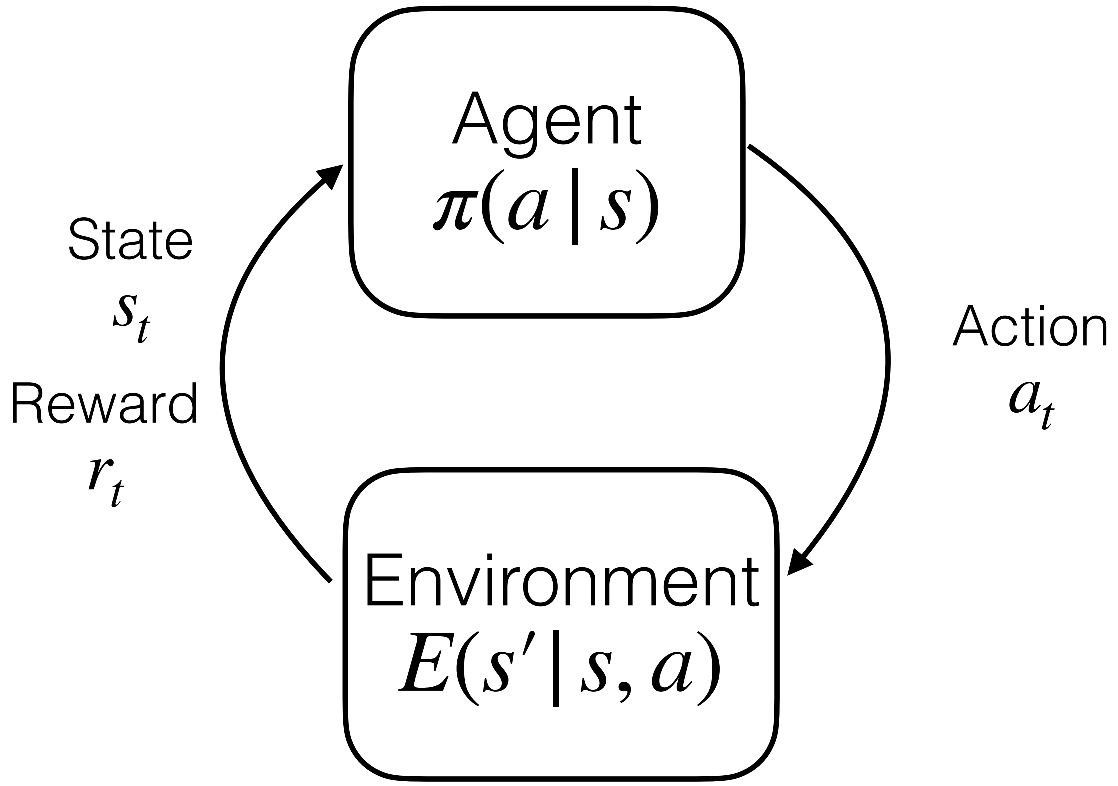
```
def agent(observation, env):
    # Find the HTML element with text "ONE"
    one_element = None
    for element in observation['dom_elements']:
      if element['text'] == "ONE":
        one_element = element
        break

    # Click on the element.
    action = env.unwrapped.create_action(
      ActionTypes.CLICK_ELEMENT, ref=one_element["ref"]
    )
    return action
```



Click button ONE.

What is an Agent?

- An agent is simply a policy!
  - Input: state
  - Output: action
- Operates in an environment



Agent
$\pi(a \mid s)$

Environment
$E(s' \mid s, a)$

State
s_t

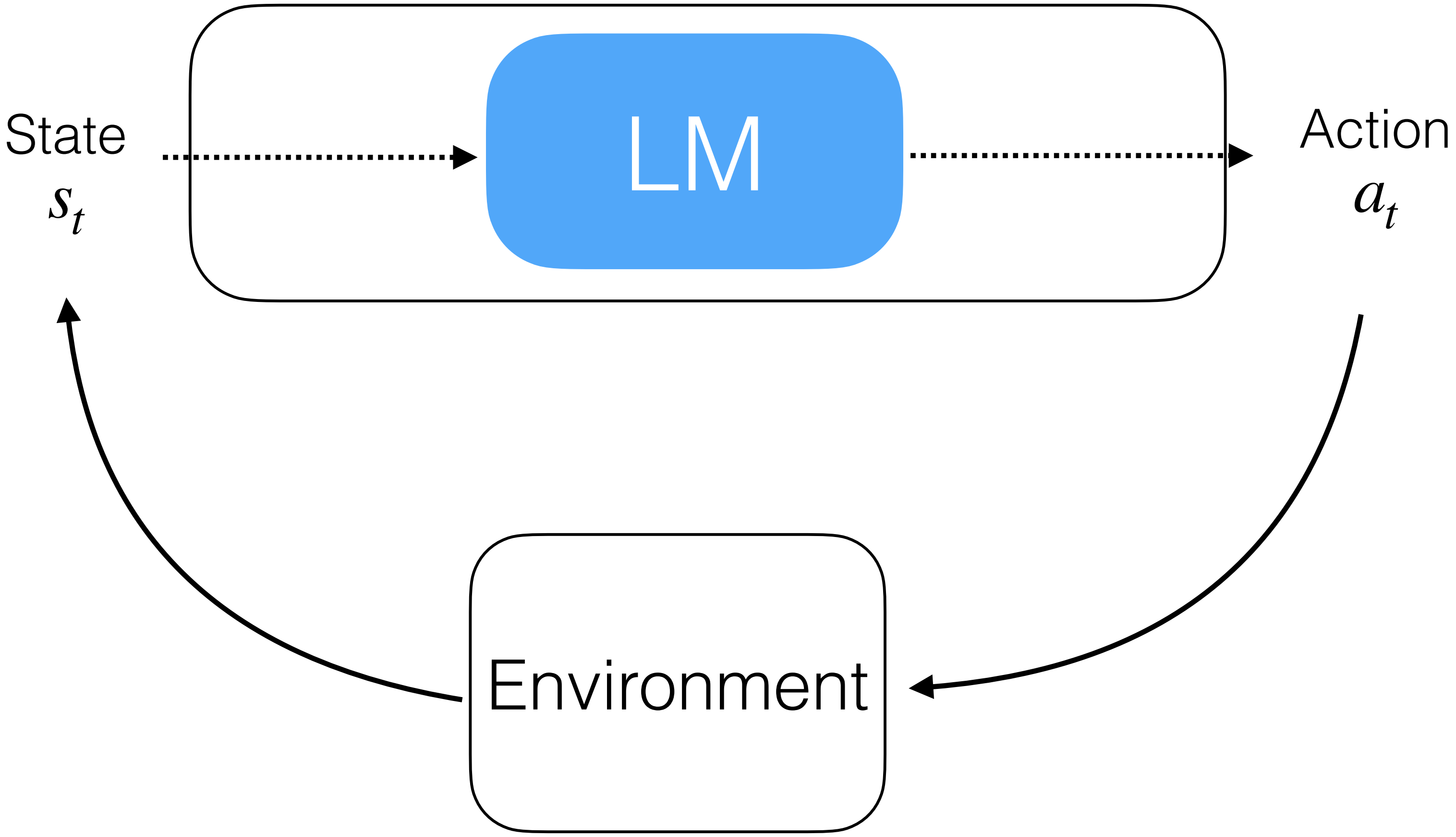
Reward
r_t

Action
a_t

Our rule-based agent: only works for one narrow task

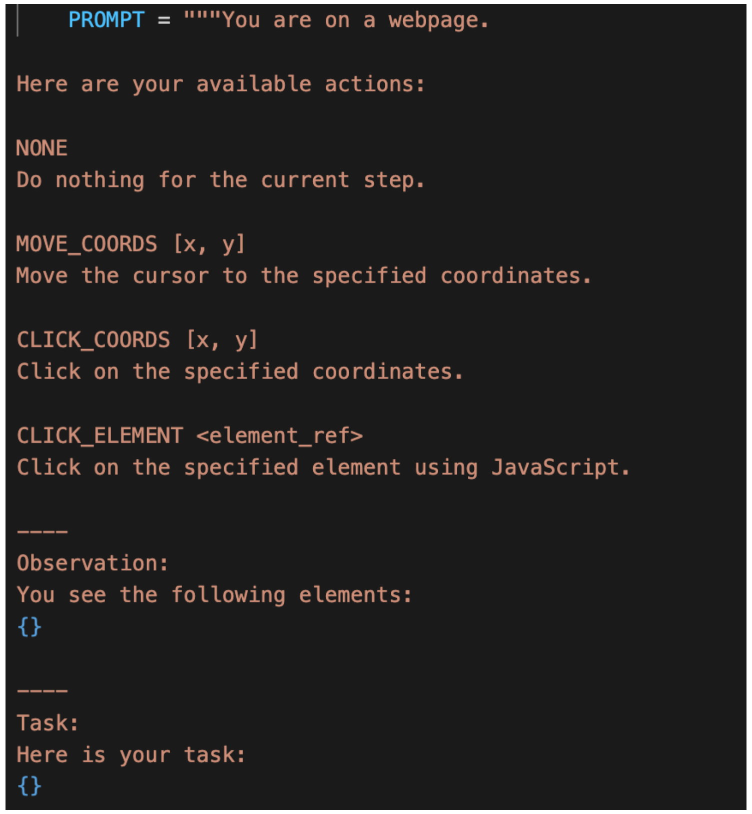
What is a **LLM** Agent?

- An agent where a language model helps determine actions



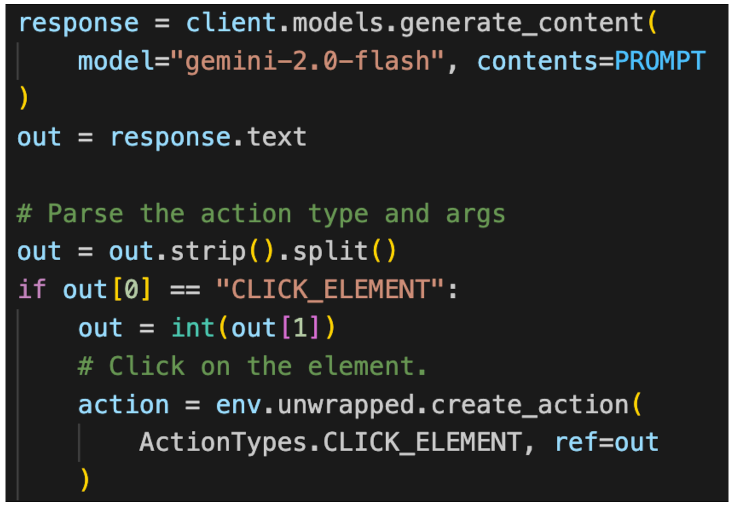
Example

Prompt

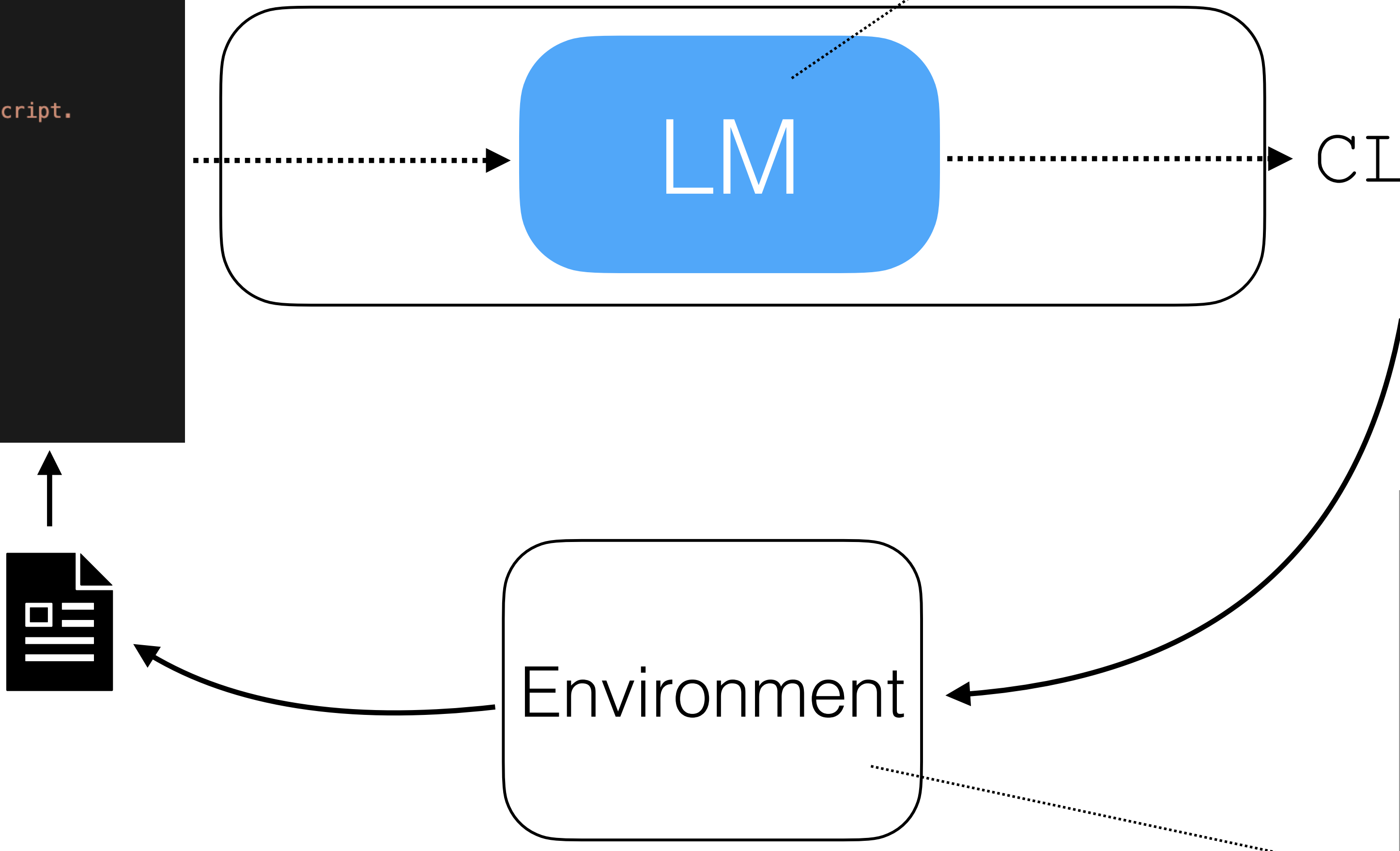


```
    PROMPT = """You are on a webpage.

Here are your available actions:

NONE
Do nothing for the current step.

MOVE_COORDS [x, y]
Move the cursor to the specified coordinates.

CLICK_COORDS [x, y]
Click on the specified coordinates.

CLICK_ELEMENT <element_ref>
Click on the specified element using JavaScript.

----
Observation:
You see the following elements:
{}

----
Task:
Here is your task:
{}
```

Call LLM and parse response

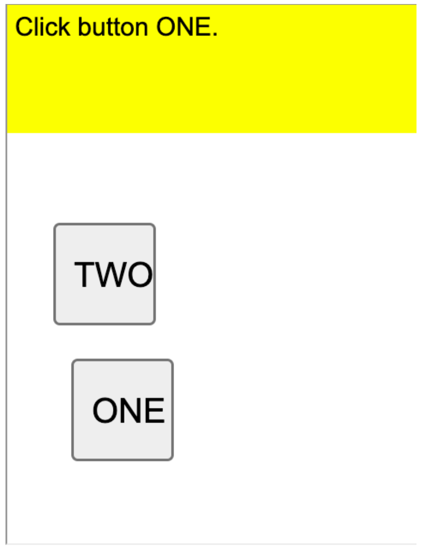


```
response = client.models.generate_content(
    model="gemini-2.0-flash", contents=PROMPT
)
out = response.text

# Parse the action type and args
out = out.strip().split()
if out[0] == "CLICK_ELEMENT":
    out = int(out[1])
    # Click on the element.
    action = env.unwrapped.create_action(
        ActionTypes.CLICK_ELEMENT, ref=out
    )
```



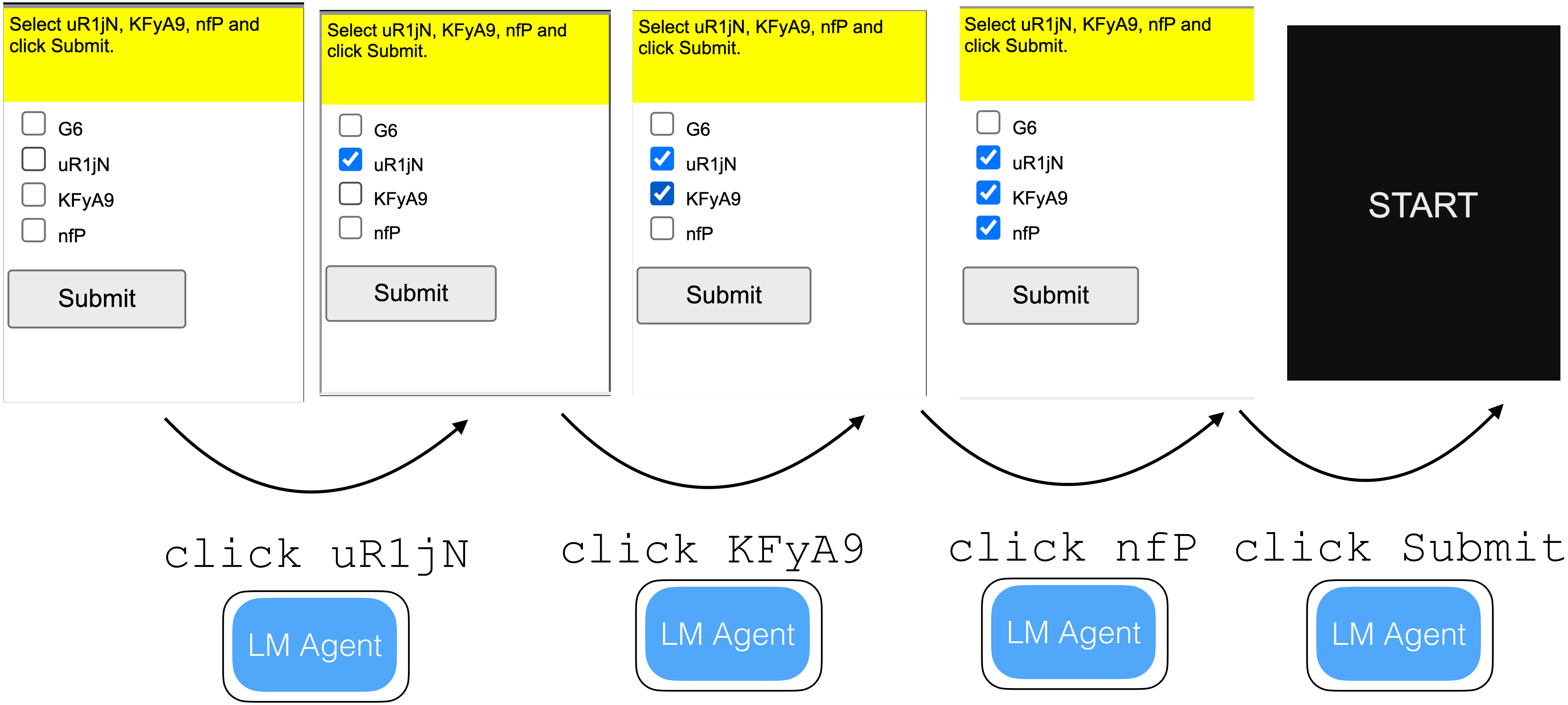
CLICK_ELEMENT 4

DOM Elements



Click button ONE.

Example



Select uR1jN, KFyA9, nfP and click Submit.

☐ G6
☐ uR1jN
☐ KFyA9
☐ nfP

Submit

Select uR1jN, KFyA9, nfP and click Submit.

☐ G6
☑ uR1jN
☐ KFyA9
☐ nfP

Submit

Select uR1jN, KFyA9, nfP and click Submit.

☐ G6
☑ uR1jN
☑ KFyA9
☐ nfP

Submit

Select uR1jN, KFyA9, nfP and click Submit.

☐ G6
☑ uR1jN
☑ KFyA9
☑ nfP

Submit

START

`click uR1jN`

`click KFyA9`

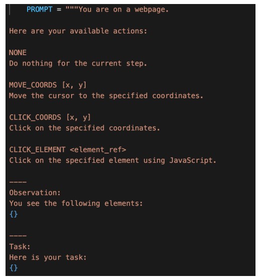
`click nfP`

`click Submit`

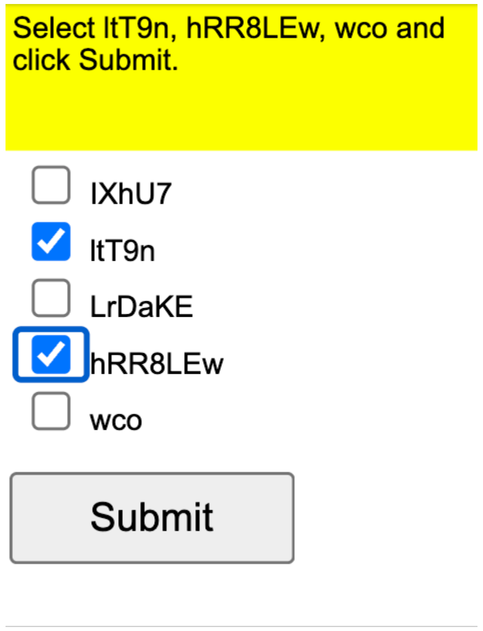
https://github.com/cmu-l3/anlp-spring2025-code/blob/main/14_agents/llm_agent.py

Vision Language Model (VLM) Agent

- An agent where a vision language model helps determine actions (VLMs in a later lecture)

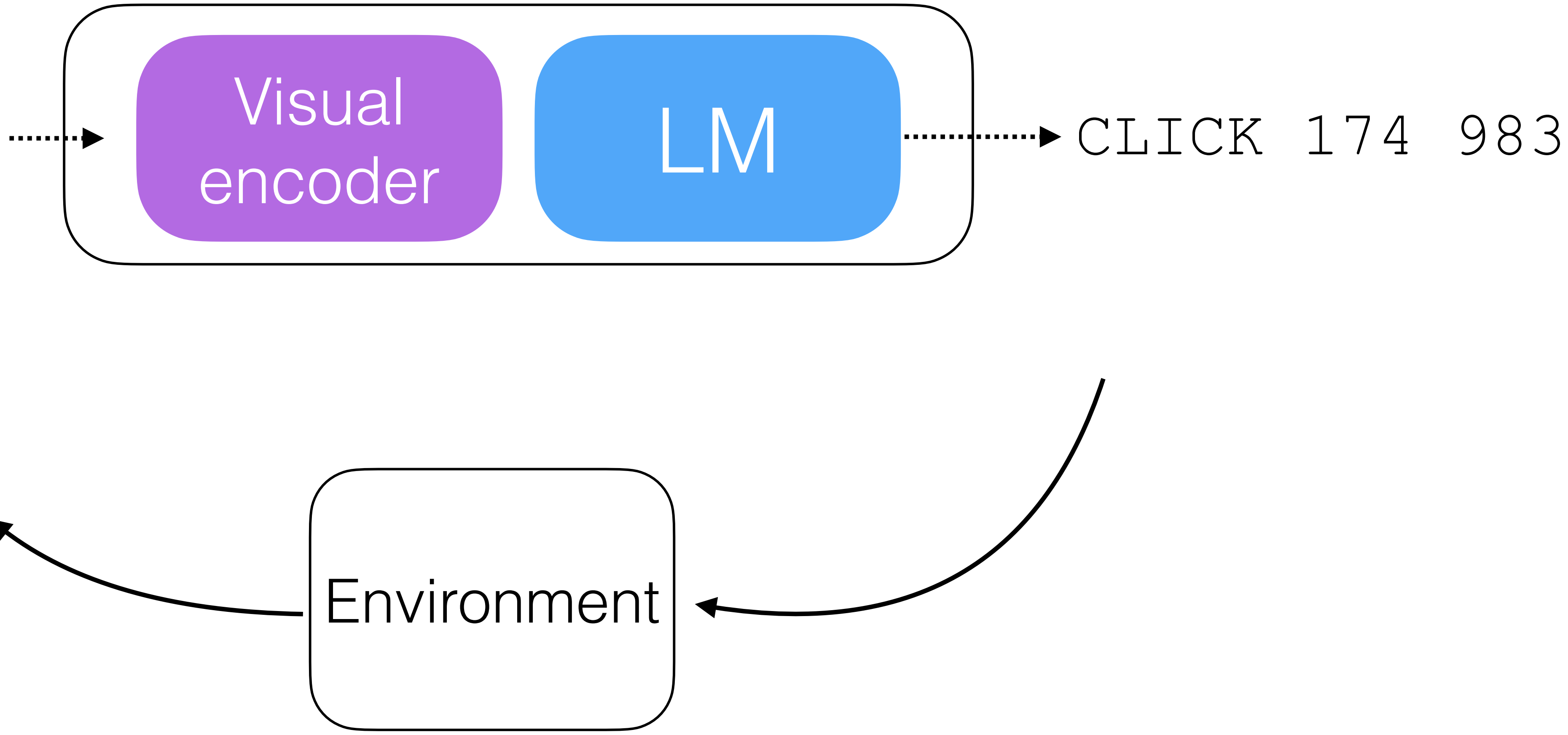


```
    PROMPT = """You are on a webpage.

Here are your available actions:

NONE
Do nothing for the current step.

MOVE_COORDS [x, y]
Move the cursor to the specified coordinates.

CLICK_COORDS [x, y]
Click on the specified coordinates.

CLICK_ELEMENT <element_ref>
Click on the specified element using JavaScript.

----
Observation:
You see the following elements:
{}

----
Task:
Here is your task:
{}
```



Select ItT9n, hRR8LEw, wco and click Submit.

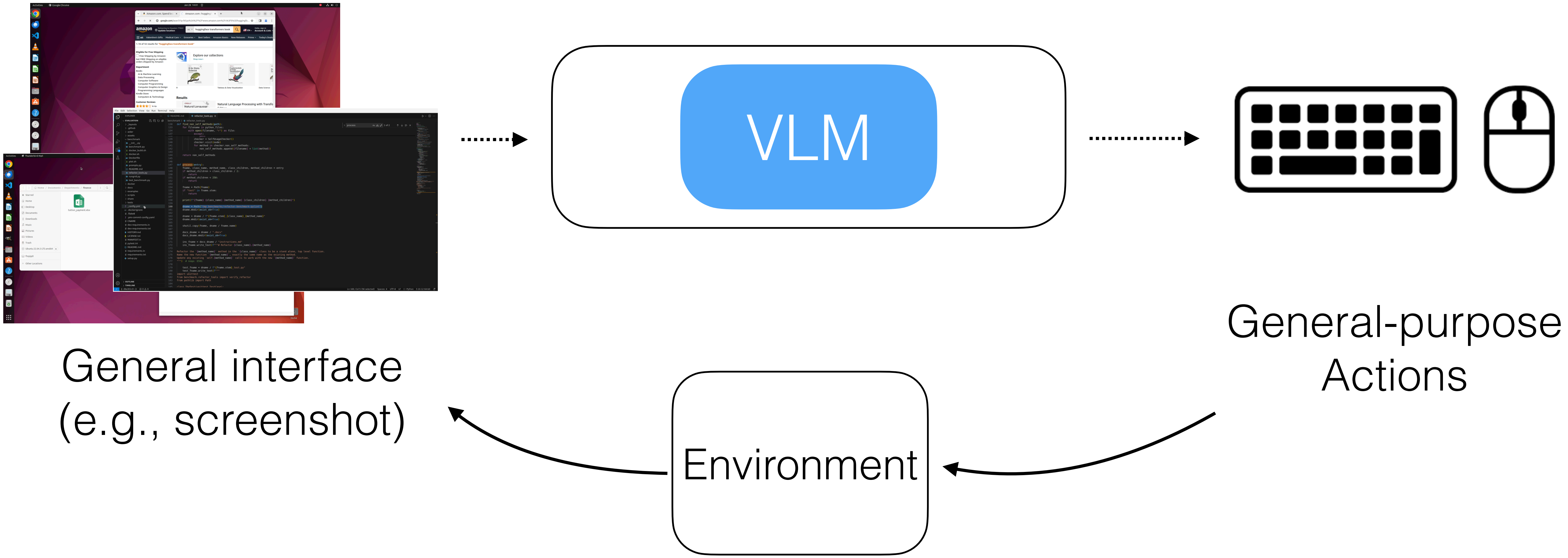
☐ lXhU7
☑ ItT9n
☐ LrDaKE
☑ hRR8LEw
☐ wco



https://github.com/cmu-l3/anlp-spring2025-code/blob/main/14_agents/vlm_agent.py

Computer-use agent

- An agent with a general-purpose observation and action space (i.e., works for any computer task)



VLM

General interface
(e.g., screenshot)

General-purpose
Actions

Environment

Recap: what is an agent?

- An agent is a policy $\pi(a \mid s)$ in an environment
- Text agents (e.g., LM agents)
    - Operate with text inputs and outputs
- Visual agents (e.g., VLM agents)
    - Additionally operate with images
- Computer-use agents
    - A task-agnostic visual agent for computer tasks

Lots of recent interest in agents

- **Software engineering**: e.g., “here’s a GitHub PR for my repository, can you close it?”
- **Research**: e.g., “compile a research report on how the AI industry has changed in the last 4 months”
- **Web**: e.g., ““Buy the cheapest color photo printer and send it to Emily's place (address in image).”

CHIPS

Nvidia CEO Says 2025 Is the Year of AI Agents

By Tae Kim Follow

Jan 07, 2025, 5:40 pm EST

Today: (LM) Agents

- What is an agent?
- **Agent environments**
- Agent patterns

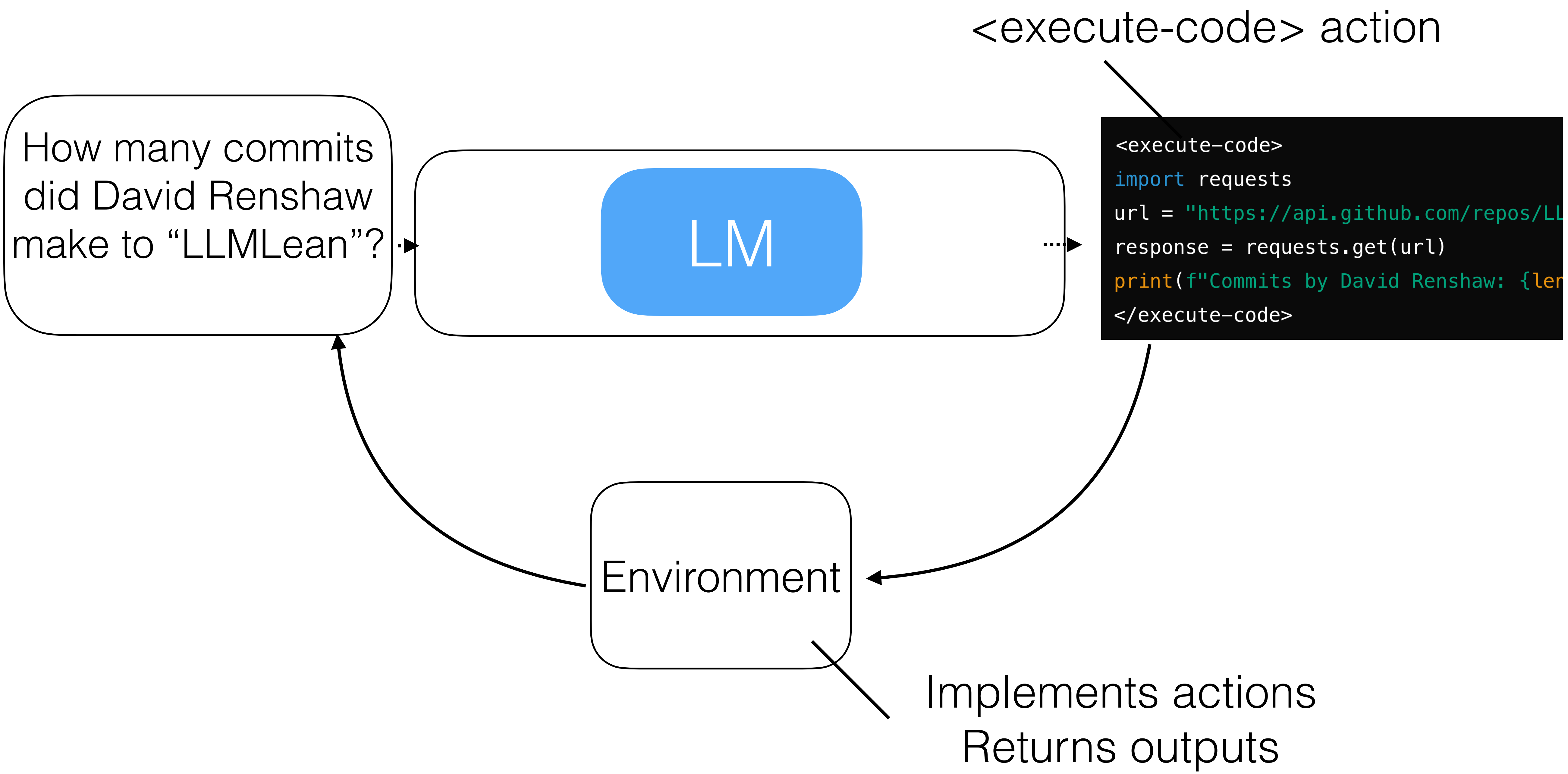
Agent environments

- Text agents
- Visual agents
- Computer-use agents

Domain-specific text environments

- Basic idea: engineer an environment for a domain
  - Example: software engineering agent
    - Actions: search_repo, execute code, edit file
    - Environment: implements the action API
    - Observations: returns outputs

Example



<execute-code> action

How many commits did David Renshaw make to “LLMLean”?

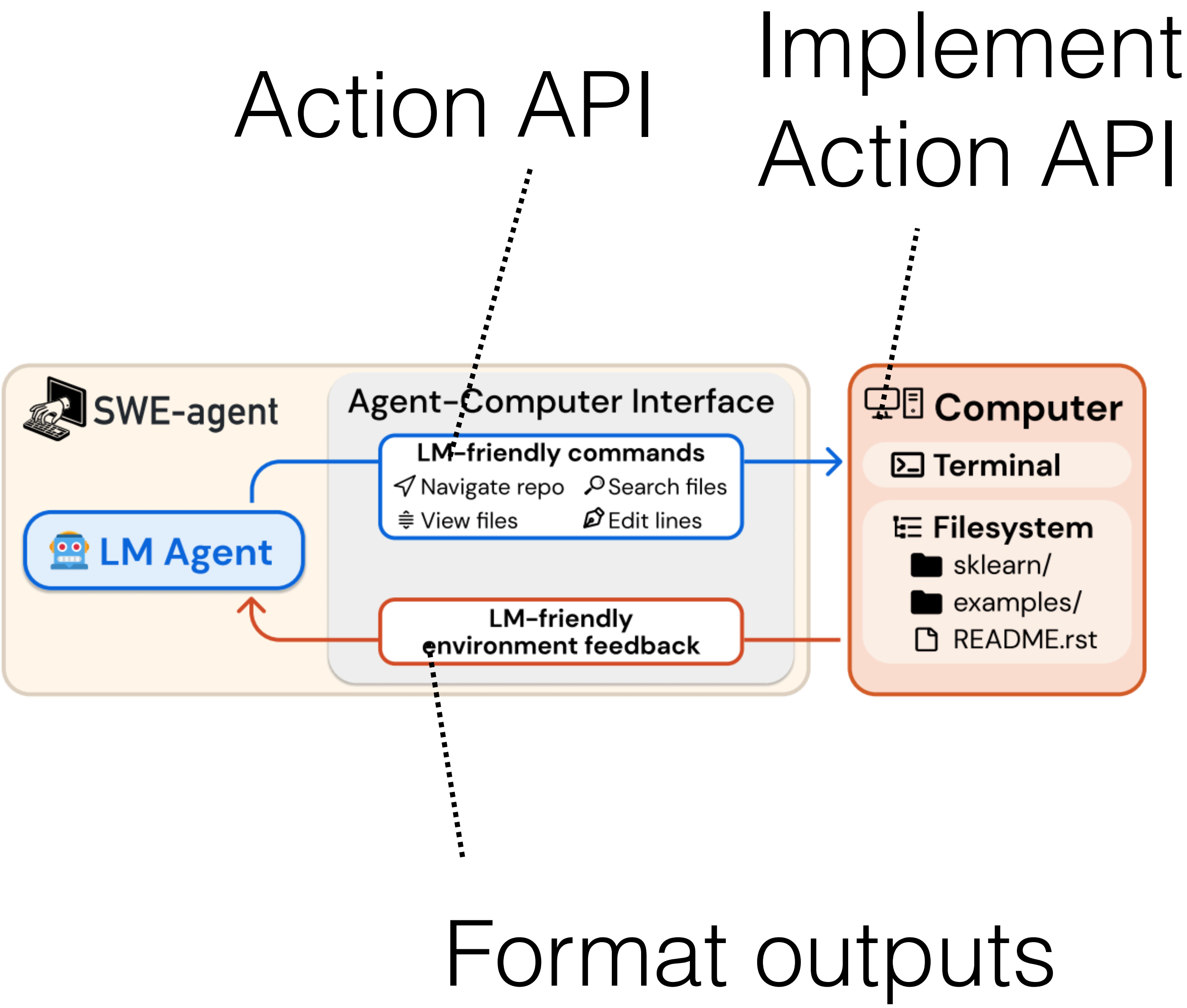
```
<execute-code>
import requests
url = "https://api.github.com/repos/LL
response = requests.get(url)
print(f"Commits by David Renshaw: {len
</execute-code>
```

Implements actions
Returns outputs

Example: SWE-agent environment

- Task: close pull requests
- Observations:
  - Action outputs
  - Task instruction
- Actions:
  - Open <path>
  - goto <line_number>
  - search_file <search_term> [<file>]

    …
- Environment: implements action API, formats outputs

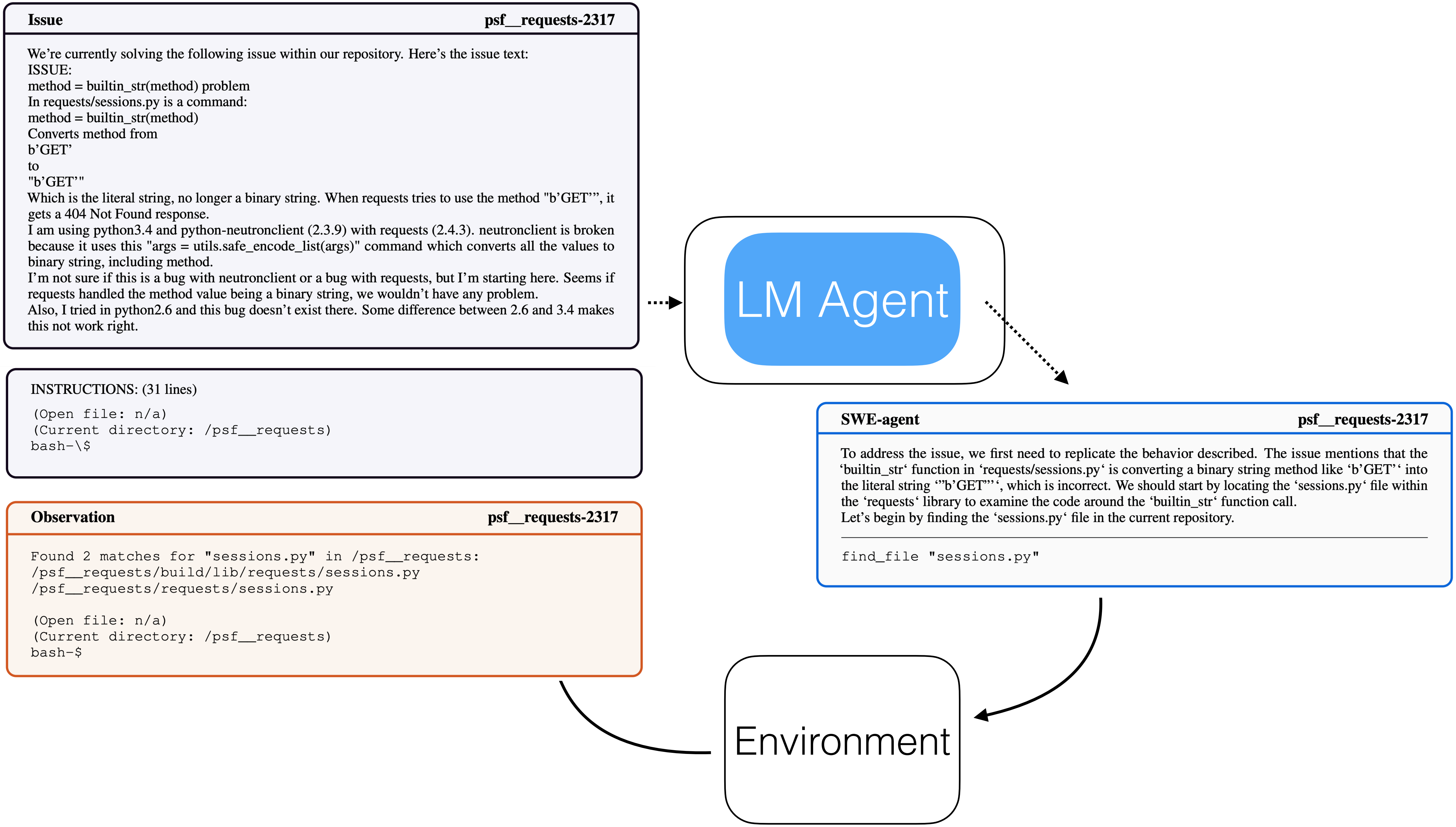


Action API

Implement Action API

Format outputs

SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering



Issue **psf__requests-2317**

We’re currently solving the following issue within our repository. Here’s the issue text:
ISSUE:
method = builtin_str(method) problem
In requests/sessions.py is a command:
method = builtin_str(method)
Converts method from
b’GET’
to
"b’GET’"
Which is the literal string, no longer a binary string. When requests tries to use the method "b’GET’”, it gets a 404 Not Found response.
I am using python3.4 and python-neutronclient (2.3.9) with requests (2.4.3). neutronclient is broken because it uses this "args = utils.safe_encode_list(args)" command which converts all the values to binary string, including method.
I’m not sure if this is a bug with neutronclient or a bug with requests, but I’m starting here. Seems if requests handled the method value being a binary string, we wouldn’t have any problem.
Also, I tried in python2.6 and this bug doesn’t exist there. Some difference between 2.6 and 3.4 makes this not work right.

INSTRUCTIONS: (31 lines)

```
(Open file: n/a)
(Current directory: /psf__requests)
bash-\$
```

SWE-agent **psf__requests-2317**

To address the issue, we first need to replicate the behavior described. The issue mentions that the ‘builtin_str‘ function in ‘requests/sessions.py‘ is converting a binary string method like ‘b’GET’‘ into the literal string ‘”b’GET”’‘, which is incorrect. We should start by locating the ‘sessions.py‘ file within the ‘requests‘ library to examine the code around the ‘builtin_str‘ function call.
Let’s begin by finding the ‘sessions.py‘ file in the current repository.

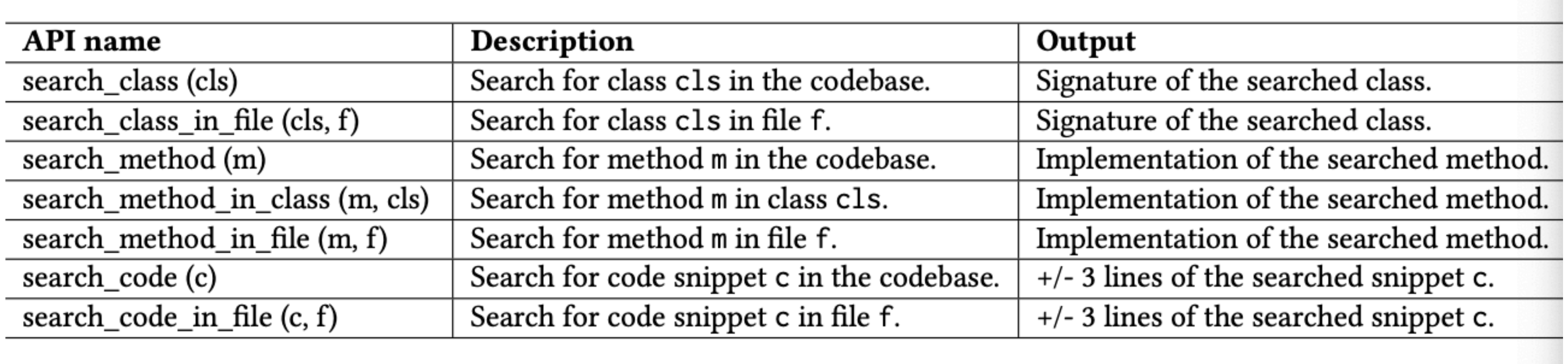
```
find_file "sessions.py"
```

Observation **psf__requests-2317**

```
Found 2 matches for "sessions.py" in /psf__requests:
/psf__requests/build/lib/requests/sessions.py
/psf__requests/requests/sessions.py

(Open file: n/a)
(Current directory: /psf__requests)
bash-$
```

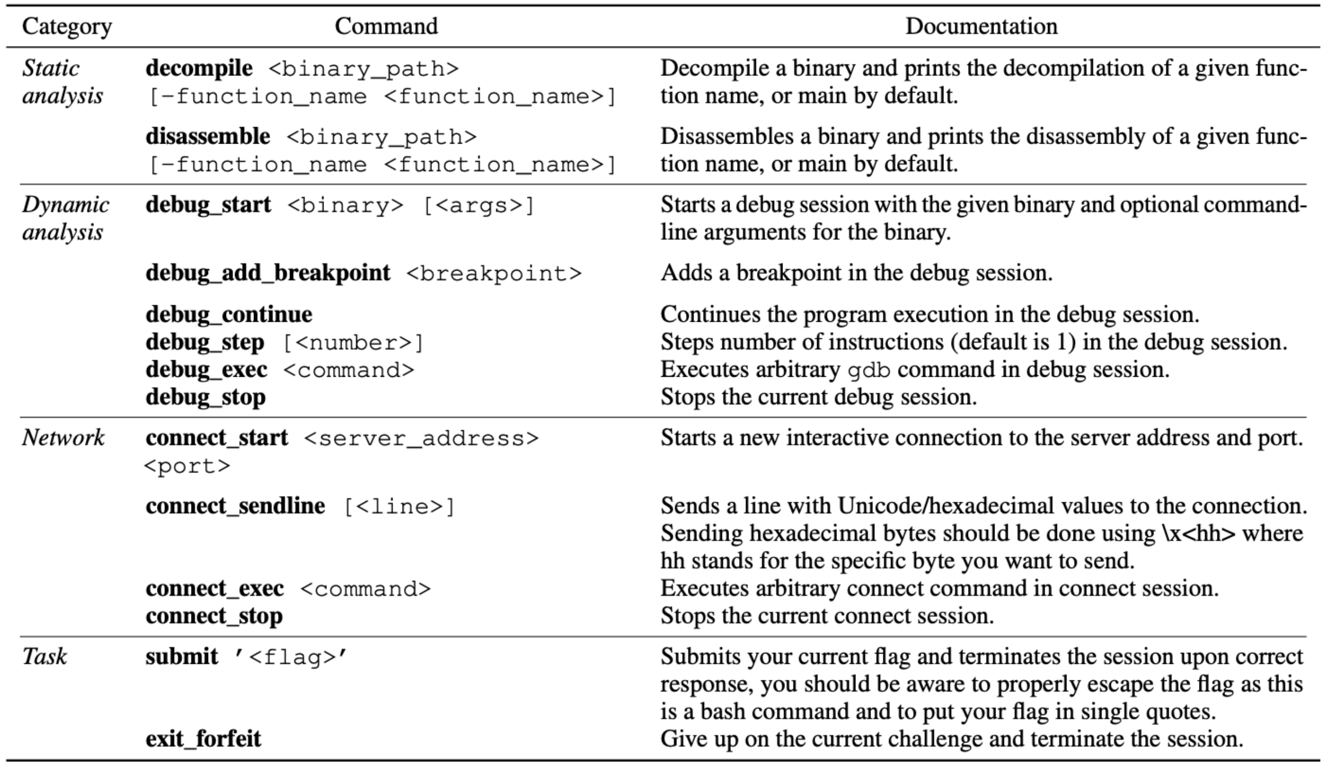
Basic idea: support new domains/tasks by implementing more action API calls

- AutoCodeRover:



API name	Description	Output
search_class (cls)	Search for class `cls` in the codebase.	Signature of the searched class.
search_class_in_file (cls, f)	Search for class `cls` in file `f`.	Signature of the searched class.
search_method (m)	Search for method `m` in the codebase.	Implementation of the searched method.
search_method_in_class (m, cls)	Search for method `m` in class `cls`.	Implementation of the searched method.
search_method_in_file (m, f)	Search for method `m` in file `f`.	Implementation of the searched method.
search_code (c)	Search for code snippet `c` in the codebase.	+/- 3 lines of the searched snippet `c`.
search_code_in_file (c, f)	Search for code snippet `c` in file `f`.	+/- 3 lines of the searched snippet `c`.

- CTF:

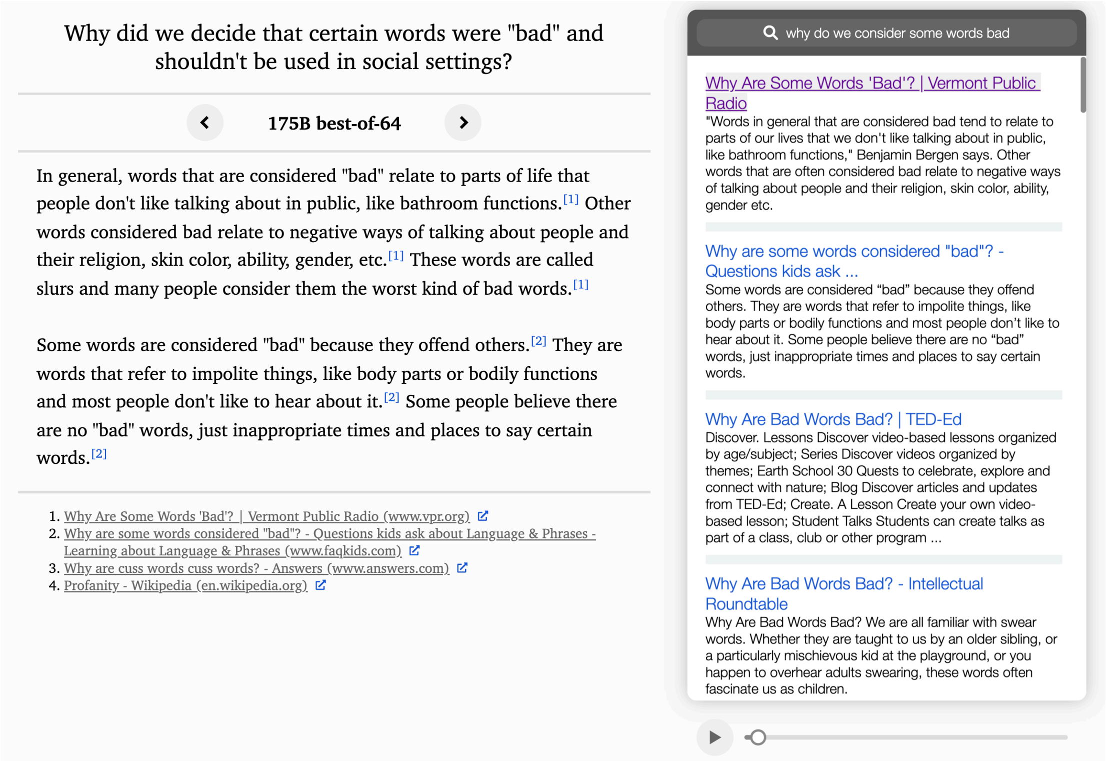


Category	Command	Documentation
Static analysis	**decompile** `<binary_path> [-function_name <function_name>]`	Decompile a binary and prints the decompilation of a given function name, or main by default.
	disassemble `<binary_path> [-function_name <function_name>]`	Disassembles a binary and prints the disassembly of a given function name, or main by default.
Dynamic analysis	**debug_start** `<binary> [<args>]`	Starts a debug session with the given binary and optional command-line arguments for the binary.
	debug_add_breakpoint `<breakpoint>`	Adds a breakpoint in the debug session.
	debug_continue	Continues the program execution in the debug session.
	debug_step `[<number>]`	Steps number of instructions (default is 1) in the debug session.
	debug_exec `<command>`	Executes arbitrary `gdb` command in debug session.
	debug_stop	Stops the current debug session.
Network	**connect_start** `<server_address> <port>`	Starts a new interactive connection to the server address and port.
	connect_sendline `[<line>]`	Sends a line with Unicode/hexadecimal values to the connection. Sending hexadecimal bytes should be done using \x<hh> where hh stands for the specific byte you want to send.
	connect_exec `<command>`	Executes arbitrary connect command in connect session.
	connect_stop	Stops the current connect session.
Task	**submit** `'<flag>'`	Submits your current flag and terminates the session upon correct response, you should be aware to properly escape the flag as this is a bash command and to put your flag in single quotes.
	exit_forfeit	Give up on the current challenge and terminate the session.

Example: text-based web browsing

- Task: answer questions
- Observations:
  - Action outputs
  - Question
- Actions:
  - Search <query>
  - Quote <text>
  - End: Answer

    …
- Environment: implements action API, formats outputs (see Appendix A)



Why did we decide that certain words were "bad" and shouldn't be used in social settings?

In general, words that are considered "bad" relate to parts of life that people don't like talking about in public, like bathroom functions.[1] Other words considered bad relate to negative ways of talking about people and their religion, skin color, ability, gender, etc.[1] These words are called slurs and many people consider them the worst kind of bad words.[1]

Some words are considered "bad" because they offend others.[2] They are words that refer to impolite things, like body parts or bodily functions and most people don't like to hear about it.[2] Some people believe there are no "bad" words, just inappropriate times and places to say certain words.[2]

1. Why Are Some Words 'Bad'? | Vermont Public Radio (www.vpr.org)
2. Why are some words considered "bad"? - Questions kids ask about Language & Phrases - Learning about Language & Phrases (www.faqkids.com)
3. Why are cuss words cuss words? - Answers (www.answers.com)
4. Profanity - Wikipedia (en.wikipedia.org)

Why Are Some Words 'Bad'? | Vermont Public Radio

"Words in general that are considered bad tend to relate to parts of our lives that we don't like talking about in public, like bathroom functions," Benjamin Bergen says. Other words that are often considered bad relate to negative ways of talking about people and their religion, skin color, ability, gender etc.

Why are some words considered "bad"? - Questions kids ask ...

Some words are considered “bad” because they offend others. They are words that refer to impolite things, like body parts or bodily functions and most people don’t like to hear about it. Some people believe there are no “bad” words, just inappropriate times and places to say certain words.

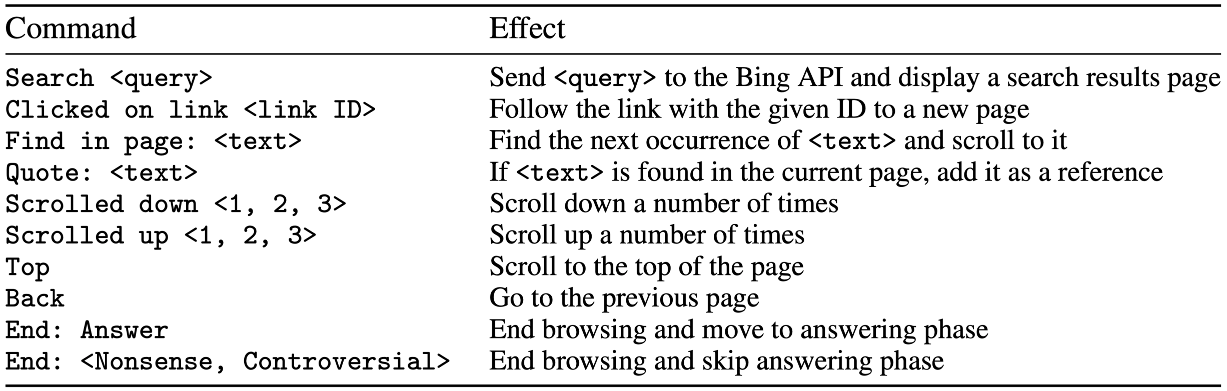
Why Are Bad Words Bad? | TED-Ed

Discover. Lessons Discover video-based lessons organized by age/subject; Series Discover videos organized by themes; Earth School 30 Quests to celebrate, explore and connect with nature; Blog Discover articles and updates from TED-Ed; Create. A Lesson Create your own video-based lesson; Student Talks Students can create talks as part of a class, club or other program ...

Why Are Bad Words Bad? - Intellectual Roundtable

Why Are Bad Words Bad? We are all familiar with swear words. Whether they are taught to us by an older sibling, or a particularly mischievous kid at the playground, or you happen to overhear adults swearing, these words often fascinate us as children.

Table 1: Actions the model can take. If a model generates any other text, it is considered to be an invalid action. Invalid actions still count towards the maximum, but are otherwise ignored.



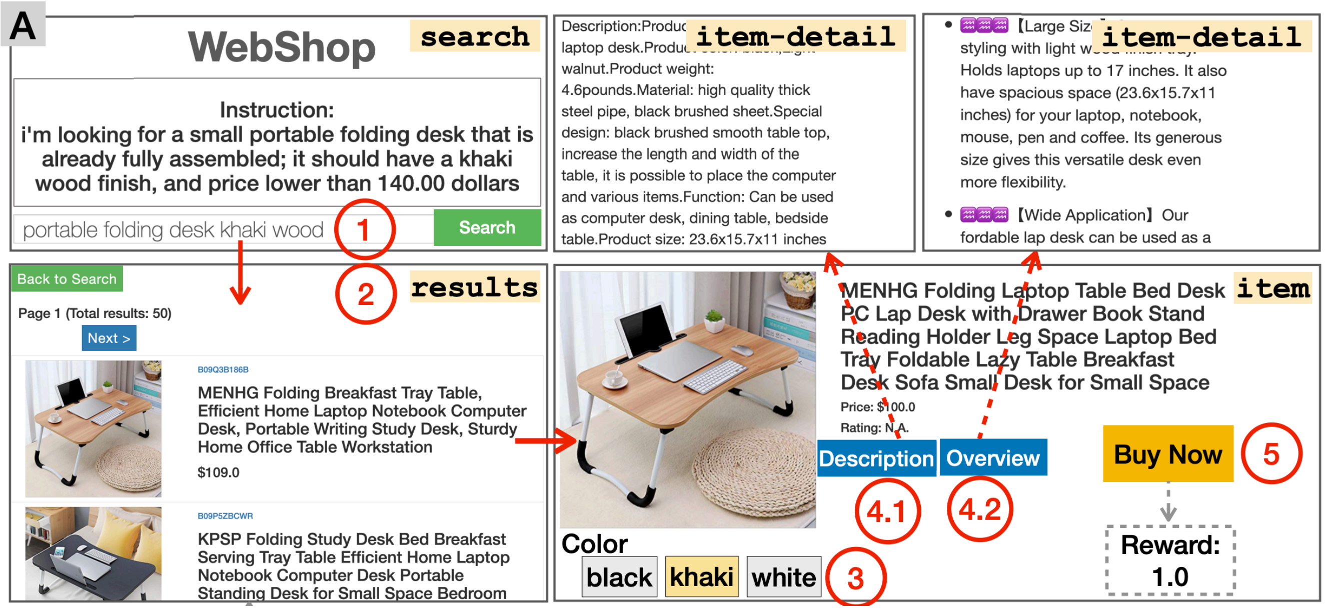
Command	Effect
`Search <query>`	Send `<query>` to the Bing API and display a search results page
`Clicked on link <link ID>`	Follow the link with the given ID to a new page
`Find in page: <text>`	Find the next occurrence of `<text>` and scroll to it
`Quote: <text>`	If `<text>` is found in the current page, add it as a reference
`Scrolled down <1, 2, 3>`	Scroll down a number of times
`Scrolled up <1, 2, 3>`	Scroll up a number of times
`Top`	Scroll to the top of the page
`Back`	Go to the previous page
`End: Answer`	End browsing and move to answering phase
`End: <Nonsense, Controversial>`	End browsing and skip answering phase

WebGPT: Browser-assisted question-answering with human feedback

https://openaipublic.blob.core.windows.net/webgpt-answer-viewer/index.html

Example: text-based web shopping

- Task: buy products
- Observations:
  - HTML of web page
  - Instruction
- Environment: simulates an e-commerce website



Instruction:
i'm looking for a small portable folding desk that is already fully assembled; it should have a khaki wood finish, and price lower than 140.00 dollars

portable folding desk khaki wood

Page 1 (Total results: 50)

B09Q3B186B
MENHG Folding Breakfast Tray Table, Efficient Home Laptop Notebook Computer Desk, Portable Writing Study Desk, Sturdy Home Office Table Workstation
$109.0

B09P5ZBCWR
KPSP Folding Study Desk Bed Breakfast Serving Tray Table Efficient Home Laptop Notebook Computer Desk Portable Standing Desk for Small Space Bedroom

Description:Produc
laptop desk.Produc
walnut.Product weight: 4.6pounds.Material: high quality thick steel pipe, black brushed sheet.Special design: black brushed smooth table top, increase the length and width of the table, it is possible to place the computer and various items.Function: Can be used as computer desk, dining table, bedside table.Product size: 23.6x15.7x11 inches

- ♒♒♒ 【Large Siz
  styling with light w
  Holds laptops up to 17 inches. It also have spacious space (23.6x15.7x11 inches) for your laptop, notebook, mouse, pen and coffee. Its generous size gives this versatile desk even more flexibility.
- ♒♒♒ 【Wide Application】 Our fordable lap desk can be used as a

MENHG Folding Laptop Table Bed Desk PC Lap Desk with Drawer Book Stand Reading Holder Leg Space Laptop Bed Tray Foldable Lazy Table Breakfast Desk Sofa Small Desk for Small Space
Price: $100.0
Rating: N.A.

Type	Argument	State → Next State
`search`	[*Query*]	Search → Results
`choose`	Back to search	$*$ → Search
`choose`	Prev/Next page	Results → Results
`choose`	[*Product title*]	Results → Item
`choose`	[*Option*]	Item → Item
`choose`	Desc/Overview	Item → Item-Detail
`choose`	Previous	Item-Detail → Item
`choose`	Buy	Item → Episode End

Table 1: Actions in WebShop.

WebShop: Towards Scalable Real-World Web Interaction with Grounded Language Agents

Recap: domain-specific text environments

- Text observations and actions
- Environment implements an action API
- Pros and cons?

Agent environments

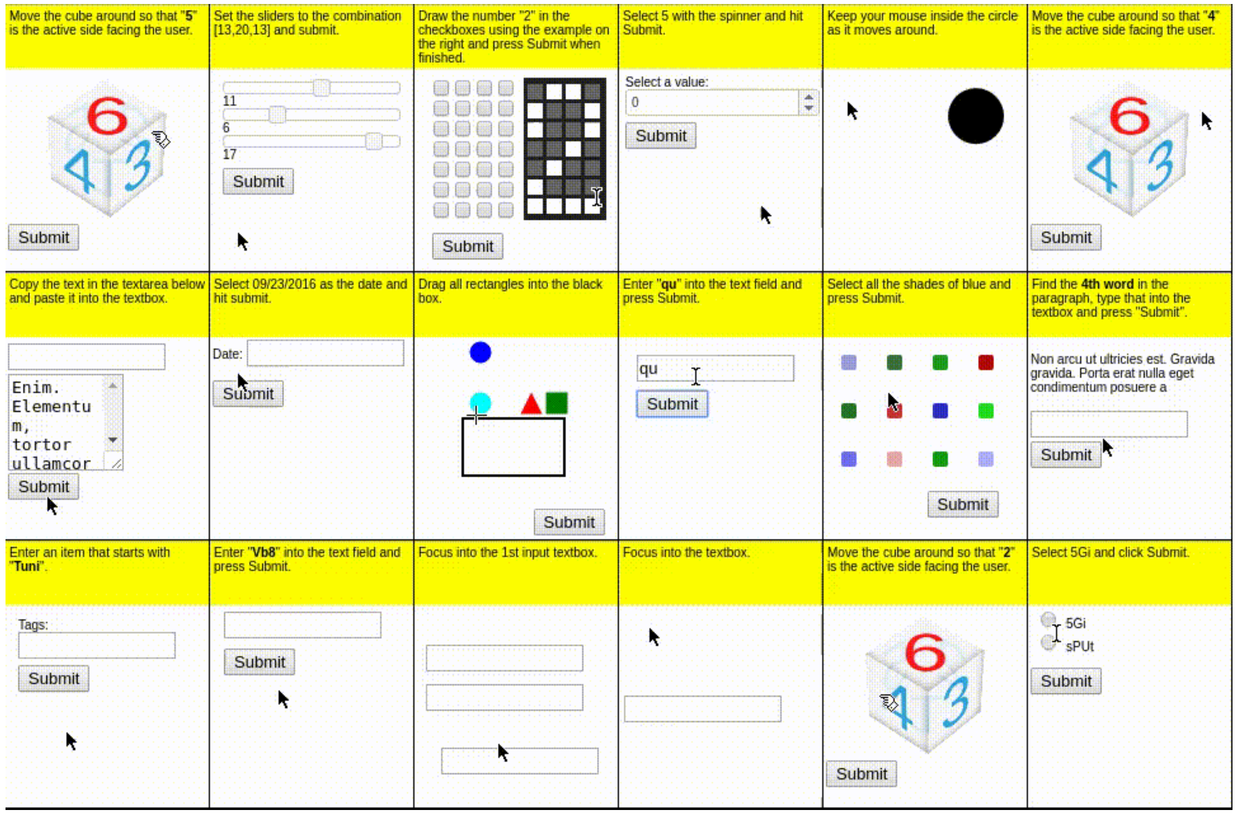
- Text agents
- **Visual agents**
- Computer-use agents

Visual agent environments

- Text + image observations

Example: basic web browsing

- Task: described in instruction (e.g., “click button”)
- Observations:
  - Image of webpage
- Actions:
  - Move, click, scroll, type, …
- Environment: implements action API, runs web browser and pages, keeps track of reward



Move the cube around so that "**5**" is the active side facing the user.

Set the sliders to the combination [13,20,13] and submit.

Draw the number "2" in the checkboxes using the example on the right and press Submit when finished.

Select 5 with the spinner and hit Submit.

Select a value: 0

Keep your mouse inside the circle as it moves around.

Move the cube around so that "**4**" is the active side facing the user.

Copy the text in the textarea below and paste it into the textbox.

Enim. Elementum, tortor ullamcor

Select 09/23/2016 as the date and hit submit.

Drag all rectangles into the black box.

Enter "**qu**" into the text field and press Submit.

qu

Select all the shades of blue and press Submit.

Find the **4th word** in the paragraph, type that into the textbox and press "Submit".

Non arcu ut ultricies est. Gravida gravida. Porta erat nulla eget condimentum posuere a

Enter an item that starts with "**Tuni**".

Enter "**Vb8**" into the text field and press Submit.

Focus into the 1st input textbox.

Focus into the textbox.

Move the cube around so that "**2**" is the active side facing the user.

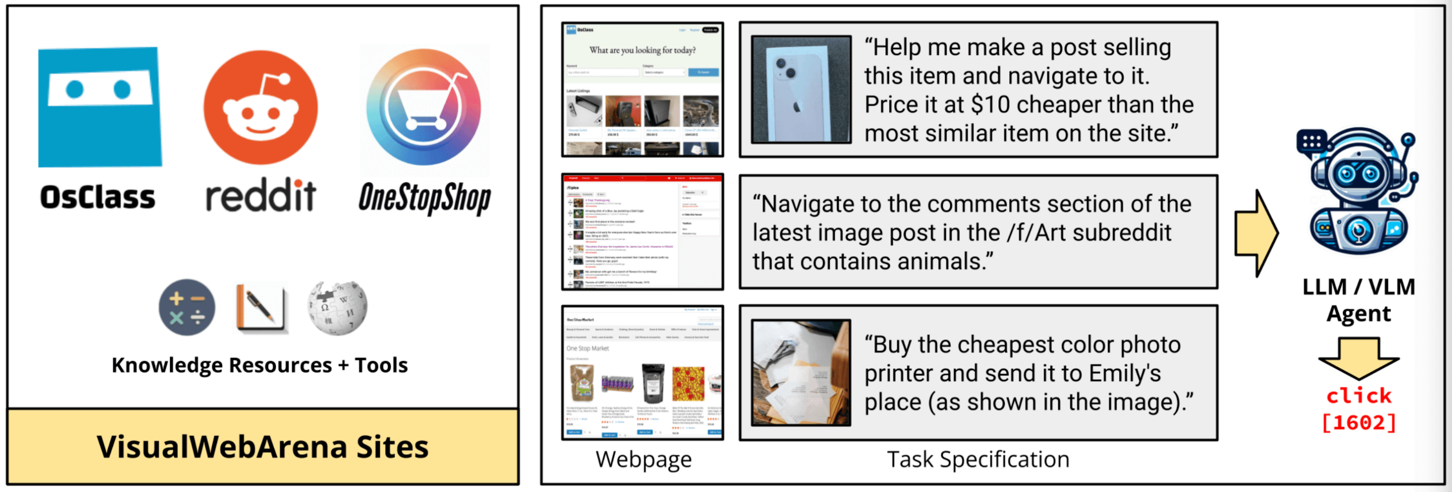
Select 5Gi and click Submit.

World of Bits: An Open-Domain Platform for Web-Based Agents, Shi et al 2017

https://miniwob.farama.org/index.html

Example: web tasks

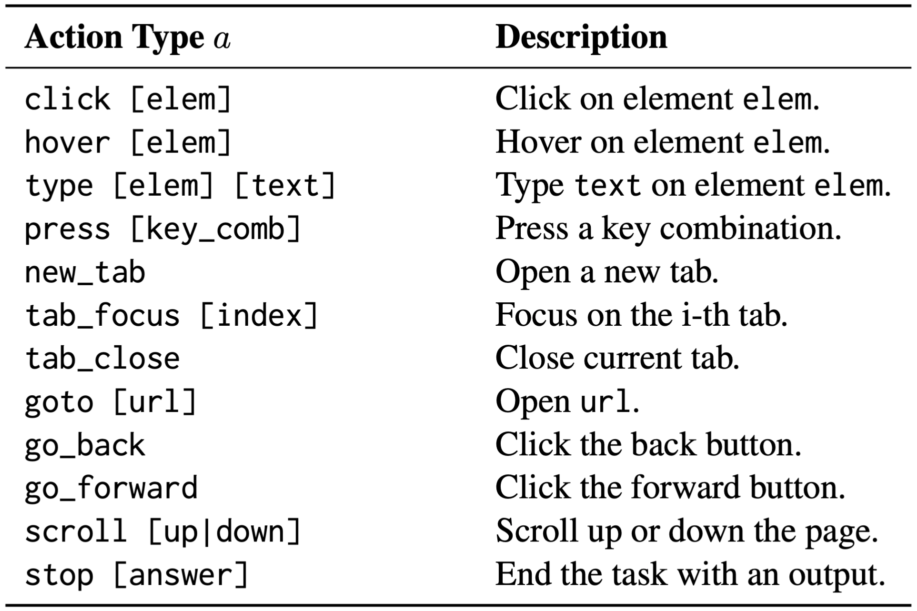
- Task: described in instruction
- Observations:
  - Screenshot of webpage
- Actions:
  - Click element, type on element, close tab, go back, …
- Environment: simulates websites:
  - Web shopping
  - Reddit
  - Classifieds marketplace



"Help me make a post selling this item and navigate to it. Price it at $10 cheaper than the most similar item on the site."

"Navigate to the comments section of the latest image post in the /f/Art subreddit that contains animals."

"Buy the cheapest color photo printer and send it to Emily's place (as shown in the image)."



Action Type a	**Description**
`click [elem]`	Click on element `elem`.
`hover [elem]`	Hover on element `elem`.
`type [elem] [text]`	Type `text` on element `elem`.
`press [key_comb]`	Press a key combination.
`new_tab`	Open a new tab.
`tab_focus [index]`	Focus on the i-th tab.
`tab_close`	Close current tab.
`goto [url]`	Open `url`.
`go_back`	Click the back button.
`go_forward`	Click the forward button.
`scroll [up|down]`	Scroll up or down the page.
`stop [answer]`	End the task with an output.

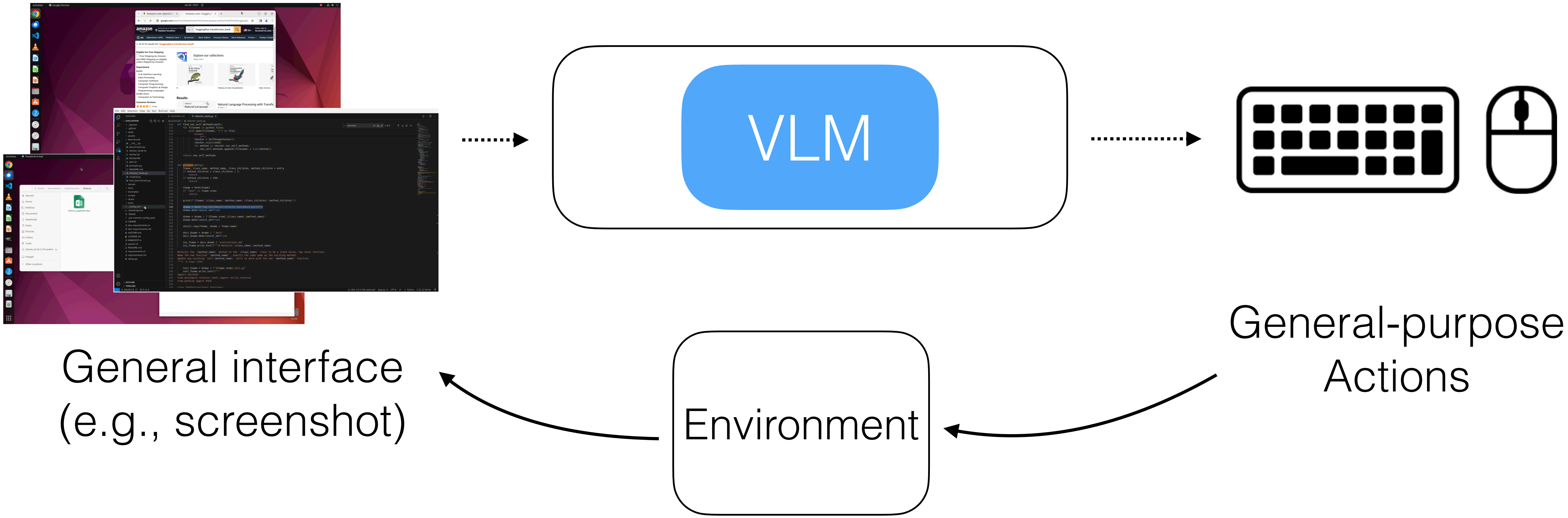
VisualWebArena: Evaluating Multimodal Agents on Realistic Visual Web Tasks, Koh et al 2024

Agent environments

- Text agents
- Visual agents
- **Computer-use agents**

Computer-use

- Basic idea: interact with a computer like a human
  - **Expressive:** in principle, agents can perform any task a human can on a computer



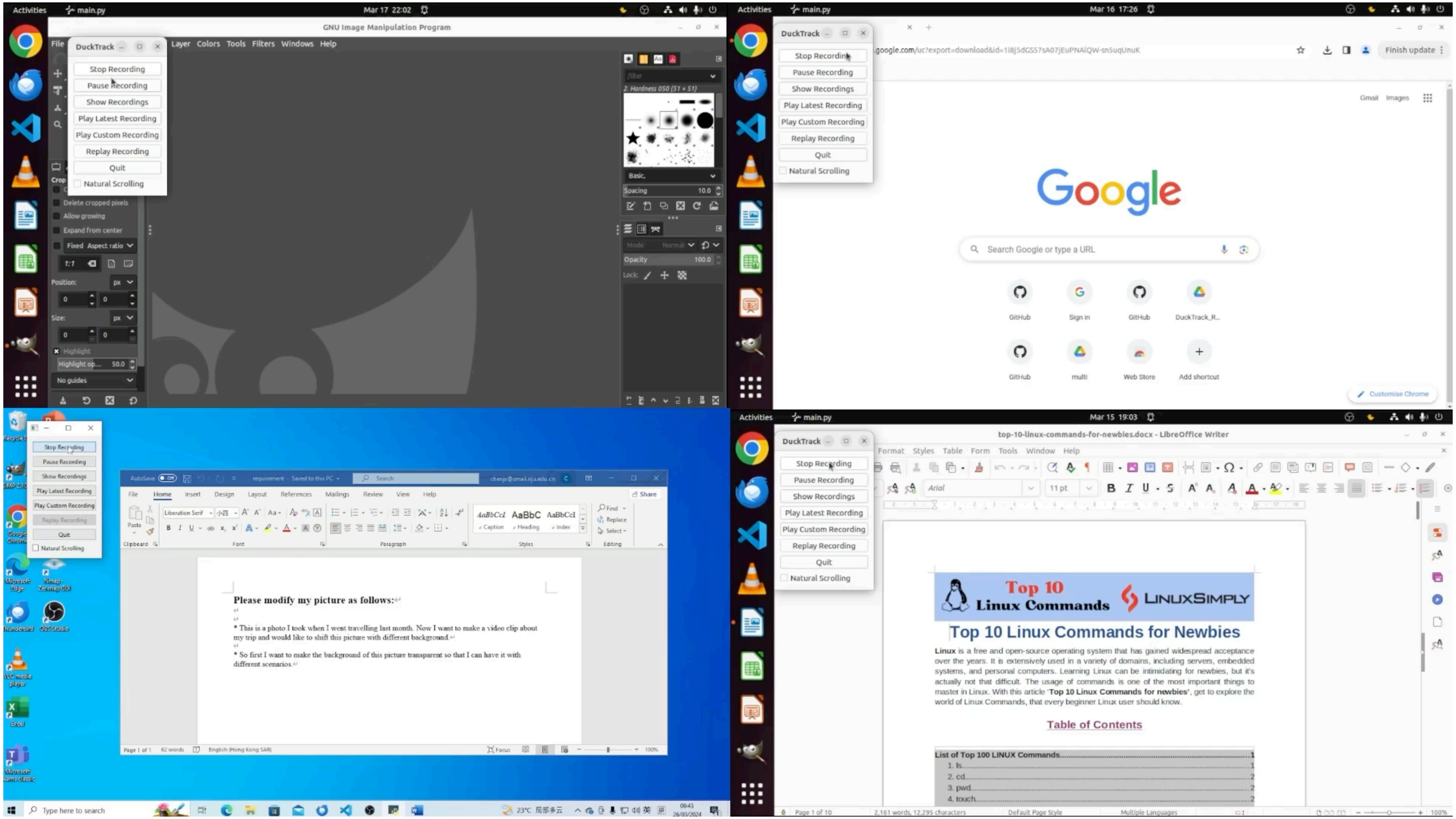
VLM

General interface
(e.g., screenshot)

General-purpose
Actions

Environment

Example: OS World



Please modify my picture as follows:

* This is a photo I took when I went travelling last month. Now I want to make a video clip about my trip and would like to shift this picture with different background.

* So first I want to make the background of this picture transparent so that I can have it with different scenarios.

Top 10 Linux Commands LINUXSIMPLY

Top 10 Linux Commands for Newbies

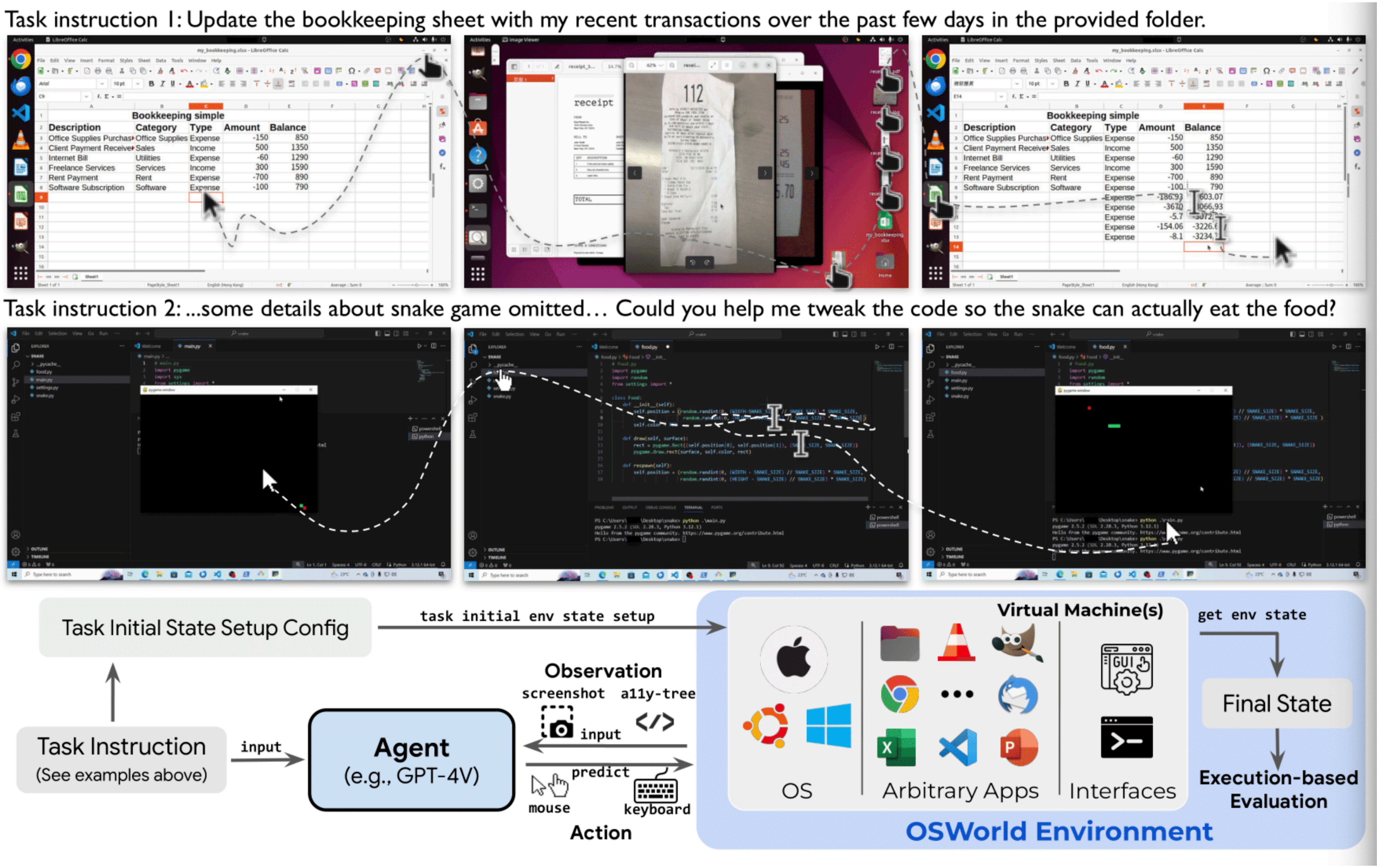
Linux is a free and open-source operating system that has gained widespread acceptance over the years. It is extensively used in a variety of domains, including servers, embedded systems, and personal computers. Learning Linux can be intimidating for newbies, but it's actually not that difficult. The usage of commands is one of the most important things to master in Linux. With this article '**Top 10 Linux Commands for newbies**', get to explore the world of Linux Commands, that every beginner Linux user should know.

Table of Contents

List of Top 100 LINUX Commands......1
1. ls......1
2. cd......2
3. pwd......2
4. touch......2

OSWorld: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments , Xie et al 2024

Example: OS World



Task instruction 1: Update the bookkeeping sheet with my recent transactions over the past few days in the provided folder.

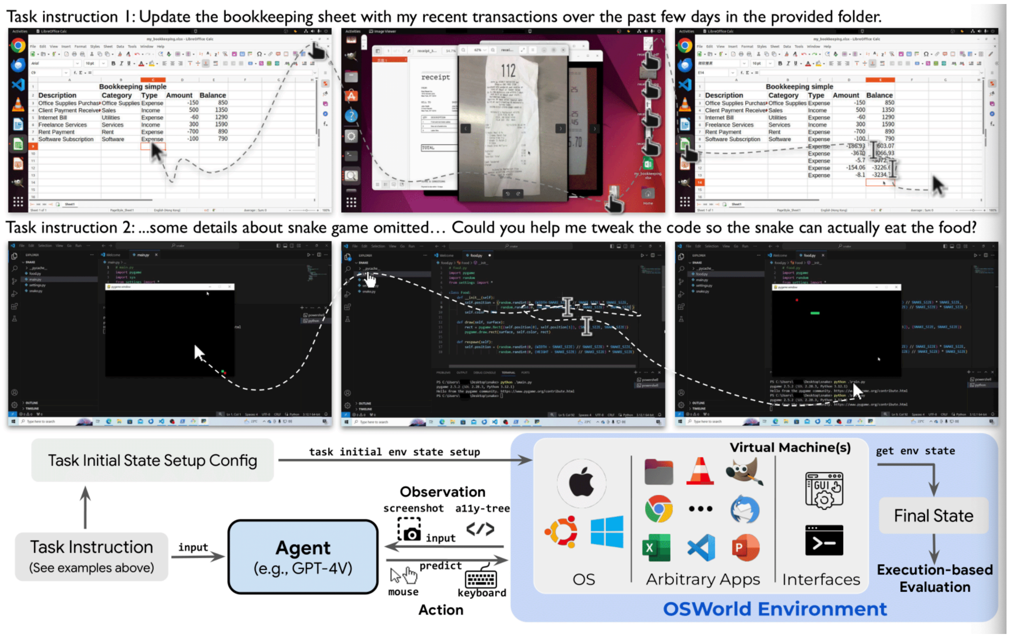
Bookkeeping simple

Description	Category	Type	Amount	Balance
Office Supplies Purchas	Office Supplies	Expense	-150	850
Client Payment Receive	Sales	Income	500	1350
Internet Bill	Utilities	Expense	-60	1290
Freelance Services	Services	Income	300	1590
Rent Payment	Rent	Expense	-700	890
Software Subscription	Software	Expense	-100	790

Task instruction 2: ...some details about snake game omitted… Could you help me tweak the code so the snake can actually eat the food?

Example: OS World

- Task
  - Any task achievable on an OS
- Observations
  - Screenshot and/or accessibility tree
  - Task instruction
- Actions
  - Mouse, keyboard
- Rewards
  - Execute an evaluation script for the task
- Environment: runs an operating system

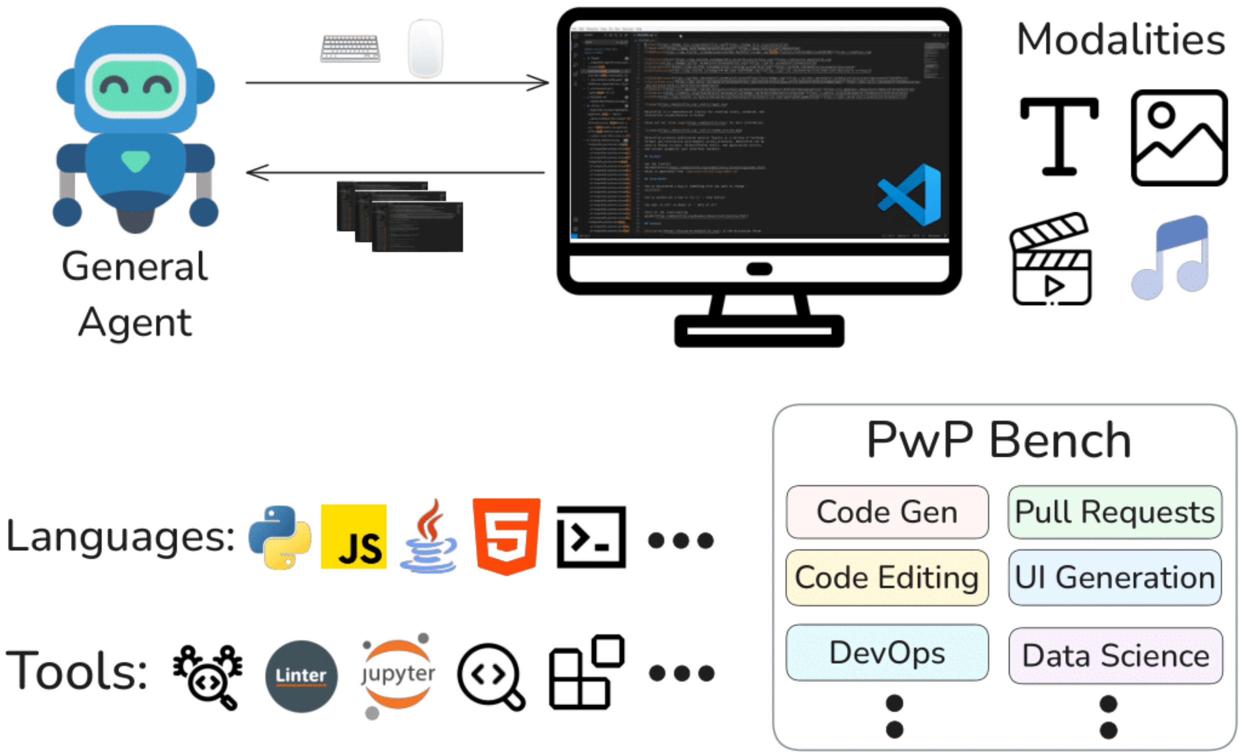


Task instruction 1: Update the bookkeeping sheet with my recent transactions over the past few days in the provided folder.

Task instruction 2: ...some details about snake game omitted… Could you help me tweak the code so the snake can actually eat the food?

Example: Programming with Pixels

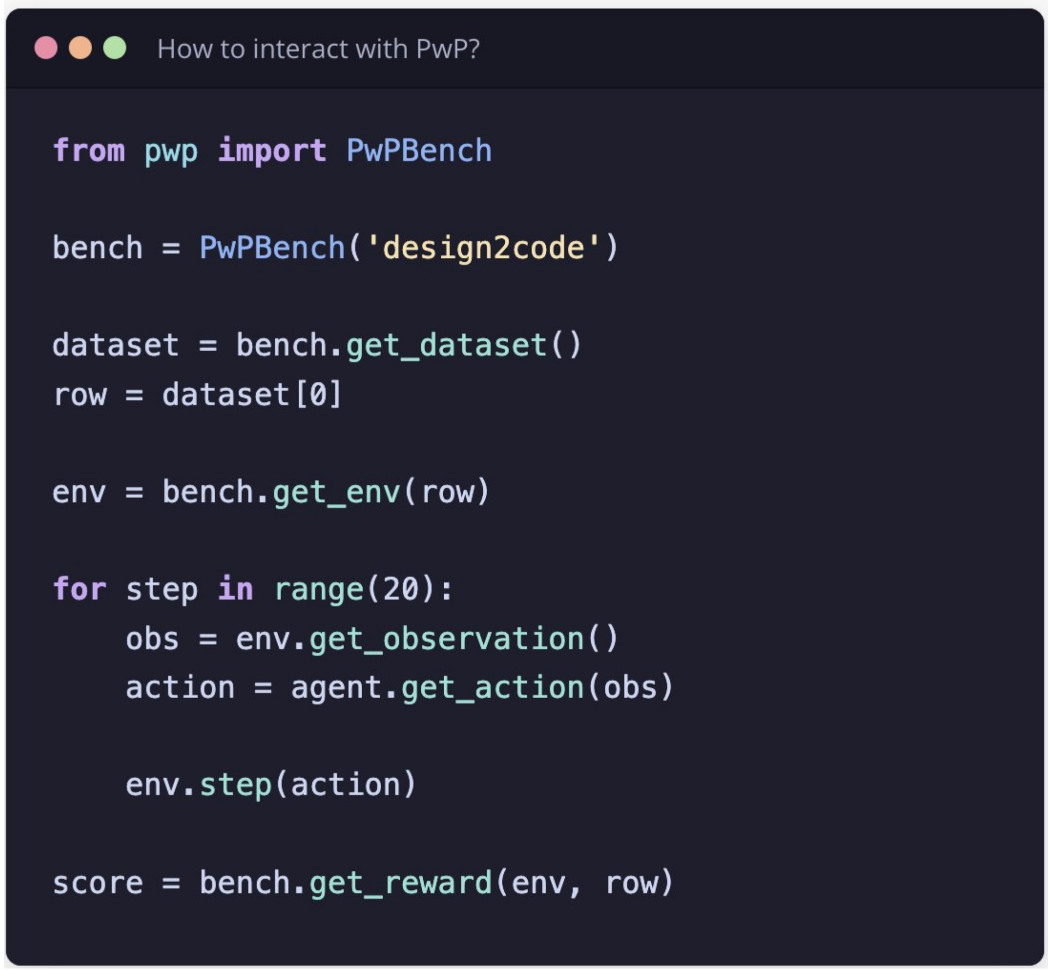
- Tasks
  - Any task achievable in an IDE (VS Code)
- Observations
  - Screenshot
  - Task instruction
- Actions
  - Mouse, keyboard
- Rewards
  - Execute an evaluation script for the task
- Environment: runs a modified VS Code



Programming with Pixels: Computer-Use Agents Meet Software Engineering, Aggarwal & Welleck 2025

Example: Programming with Pixels

- Tasks
    - Any task achievable in an IDE (VS Code)
- Observations
    - Screenshot
    - Task instruction
- Actions
    - Mouse, keyboard
- Rewards
    - Execute an evaluation script for the task
- Environment: runs a modified VS Code



```python
from pwp import PwPBench

bench = PwPBench('design2code')

dataset = bench.get_dataset()
row = dataset[0]

env = bench.get_env(row)

for step in range(20):
    obs = env.get_observation()
    action = agent.get_action(obs)

    env.step(action)

score = bench.get_reward(env, row)
```

Programming with Pixels: Computer-Use Agents Meet Software Engineering, Aggarwal & Welleck 2025

Today: (LM) Agents

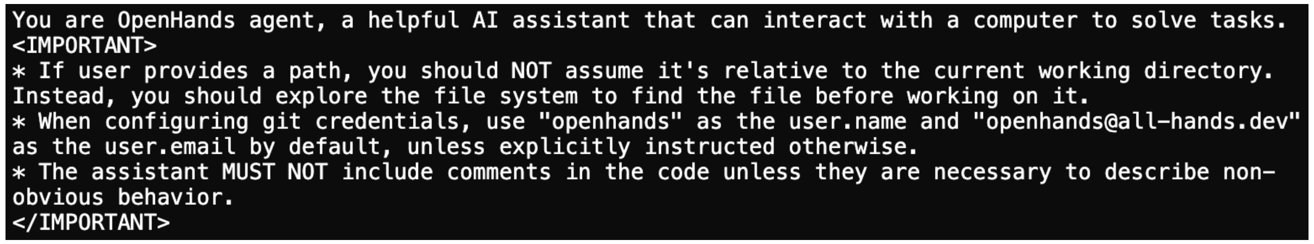
- What is an agent?
- Agent environments
- **Agent patterns**

Agent Patterns

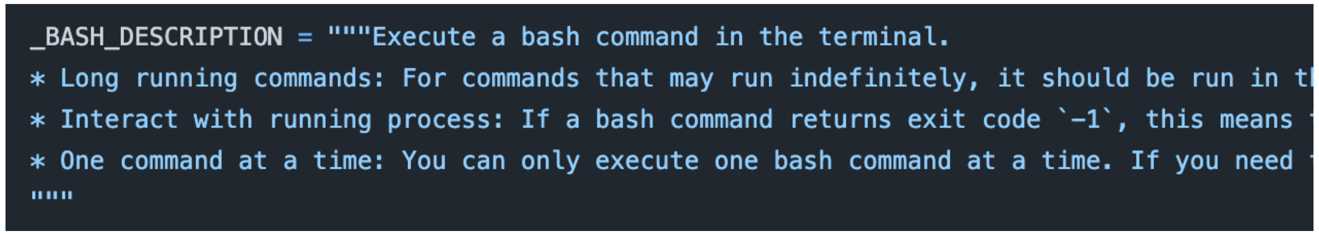
- Prompting and chain-of-thought
- Representing screenshots
- Computer use API

Prompting

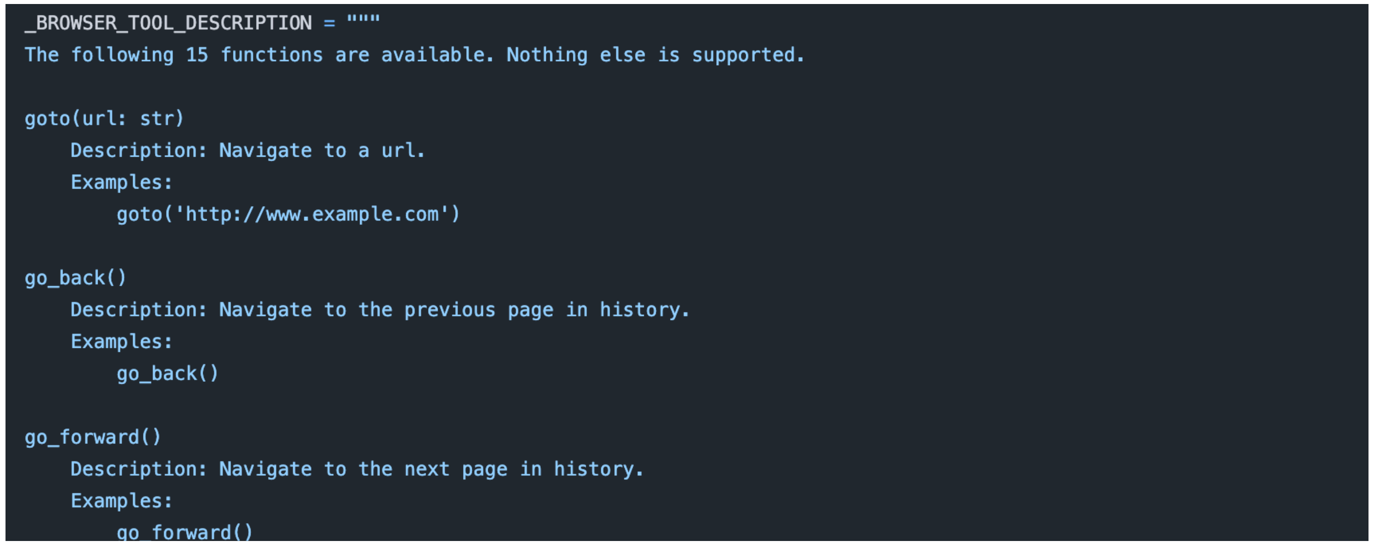
- Describe task (system prompt)
- Describe actions (“tools”)
- Format observations and actions



```
You are OpenHands agent, a helpful AI assistant that can interact with a computer to solve tasks.
<IMPORTANT>
* If user provides a path, you should NOT assume it's relative to the current working directory.
Instead, you should explore the file system to find the file before working on it.
* When configuring git credentials, use "openhands" as the user.name and "openhands@all-hands.dev"
as the user.email by default, unless explicitly instructed otherwise.
* The assistant MUST NOT include comments in the code unless they are necessary to describe non-
obvious behavior.
</IMPORTANT>
```



```
_BASH_DESCRIPTION = """Execute a bash command in the terminal.
* Long running commands: For commands that may run indefinitely, it should be run in t
* Interact with running process: If a bash command returns exit code `-1`, this means
* One command at a time: You can only execute one bash command at a time. If you need
"""
```

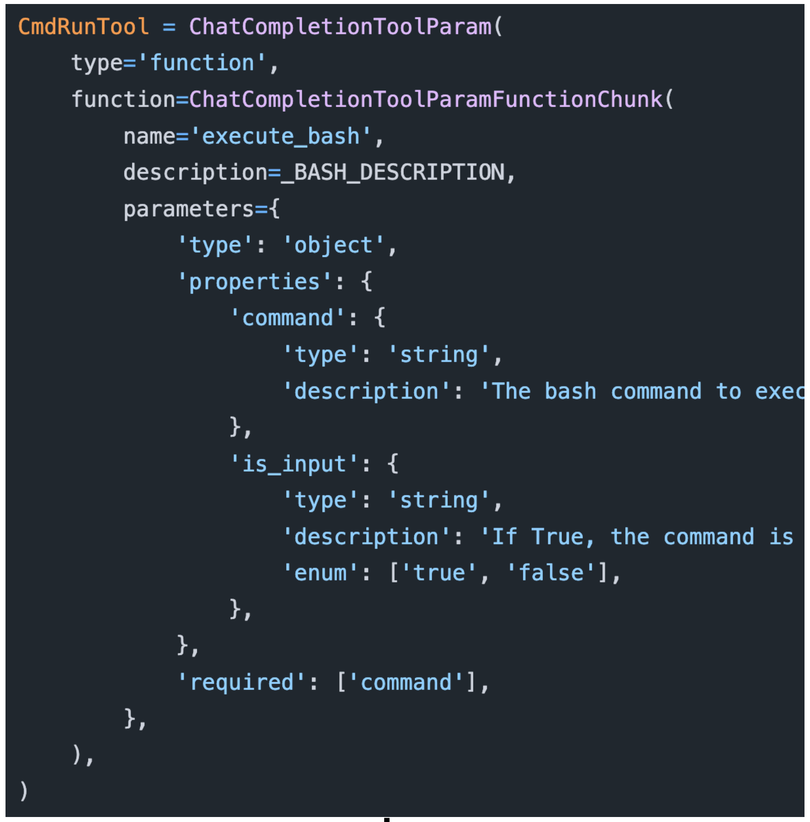


```
_BROWSER_TOOL_DESCRIPTION = """
The following 15 functions are available. Nothing else is supported.

goto(url: str)
    Description: Navigate to a url.
    Examples:
        goto('http://www.example.com')

go_back()
    Description: Navigate to the previous page in history.
    Examples:
        go_back()

go_forward()
    Description: Navigate to the next page in history.
    Examples:
        go_forward()
```

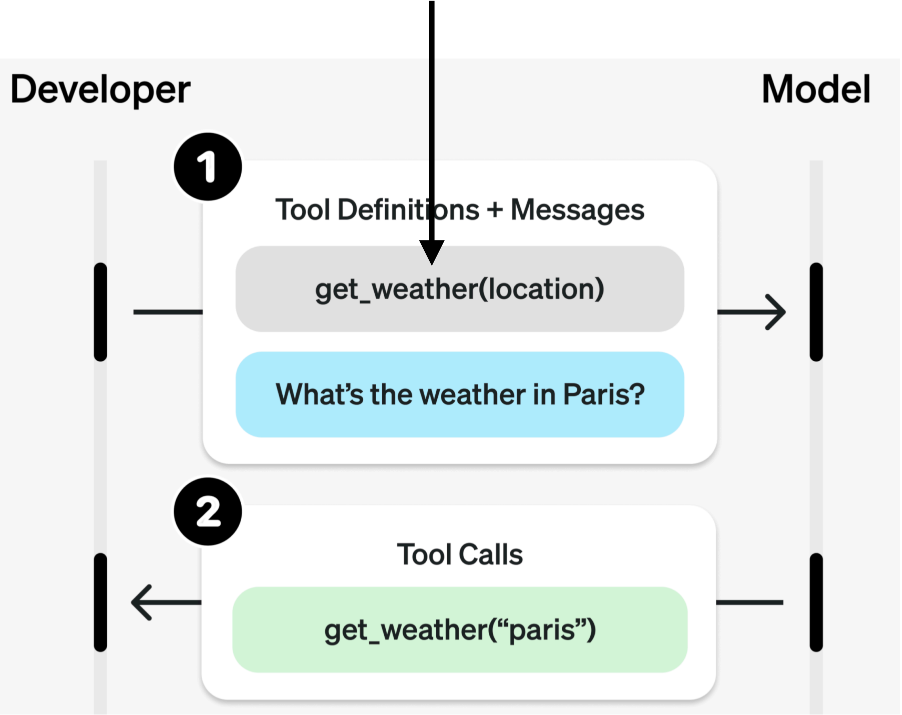
https://github.com/All-Hands-AI/OpenHands/blob/main/openhands/agenthub/codeact_agent

Function calling interface

- Commercial language model APIs provide a function calling interface
  - Describe the action (“function call”) as JSON
  - Language model can then know to generate an output that contains a properly-formatted action
  - We still need to implement the action’s execution in an environment

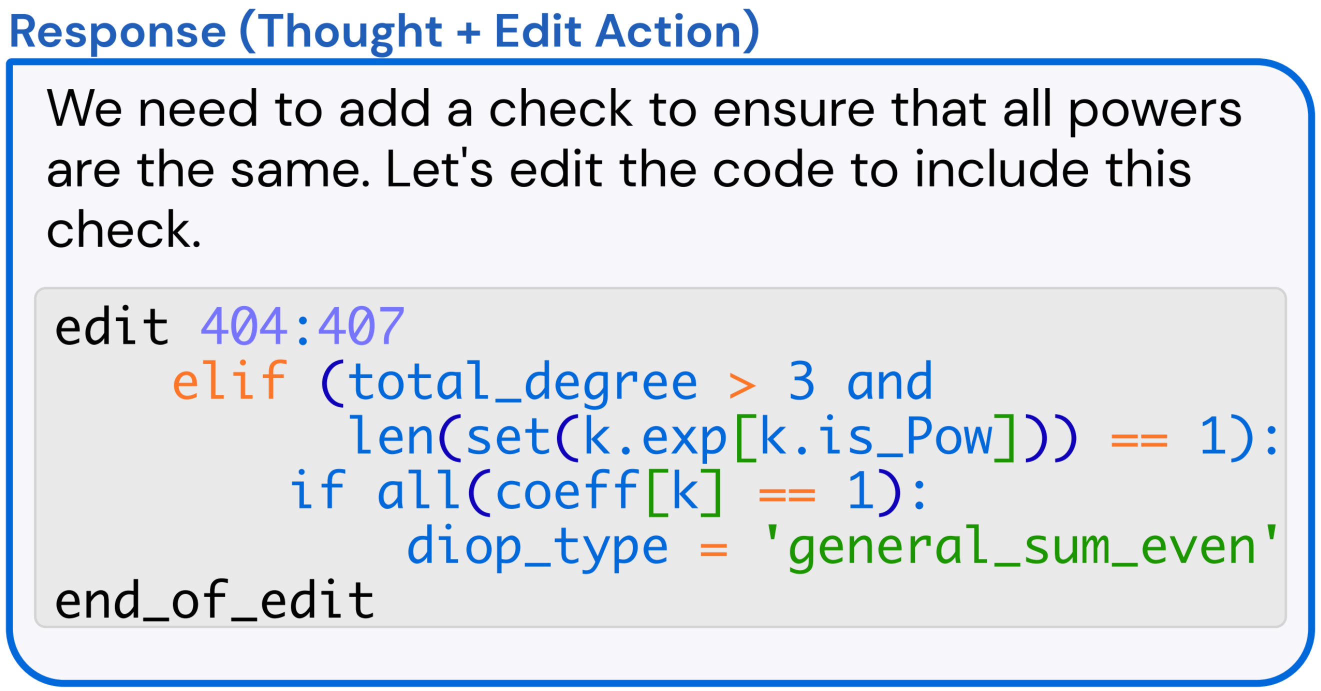


```
CmdRunTool = ChatCompletionToolParam(
    type='function',
    function=ChatCompletionToolParamFunctionChunk(
        name='execute_bash',
        description=_BASH_DESCRIPTION,
        parameters={
            'type': 'object',
            'properties': {
                'command': {
                    'type': 'string',
                    'description': 'The bash command to exec
                },
                'is_input': {
                    'type': 'string',
                    'description': 'If True, the command is
                    'enum': ['true', 'false'],
                },
            },
            'required': ['command'],
        },
    ),
)
```



Chain of thought

- Prompt the model to output a thought prior to the action



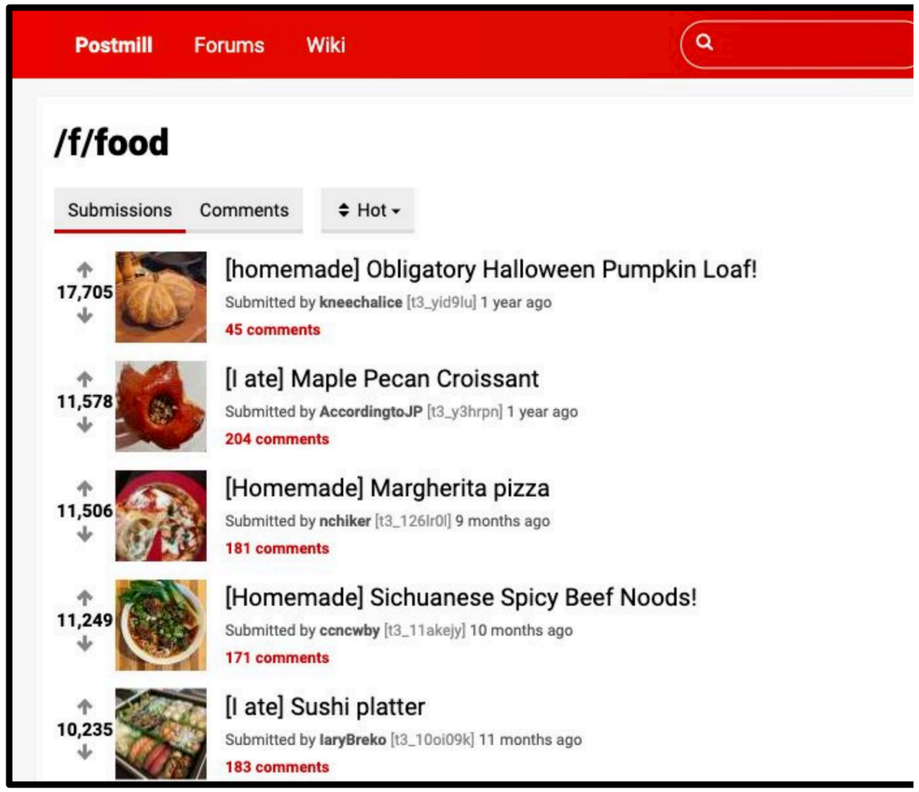
Response (Thought + Edit Action)

We need to add a check to ensure that all powers are the same. Let's edit the code to include this check.

```
edit 404:407
    elif (total_degree > 3 and
          len(set(k.exp[k.is_Pow])) == 1):
        if all(coeff[k] == 1):
            diop_type = 'general_sum_even'
end_of_edit
```

Representing screenshots

Screenshot



/f/food

17,705 [homemade] Obligatory Halloween Pumpkin Loaf!
Submitted by kneechalice [t3_yid9lu] 1 year ago
45 comments

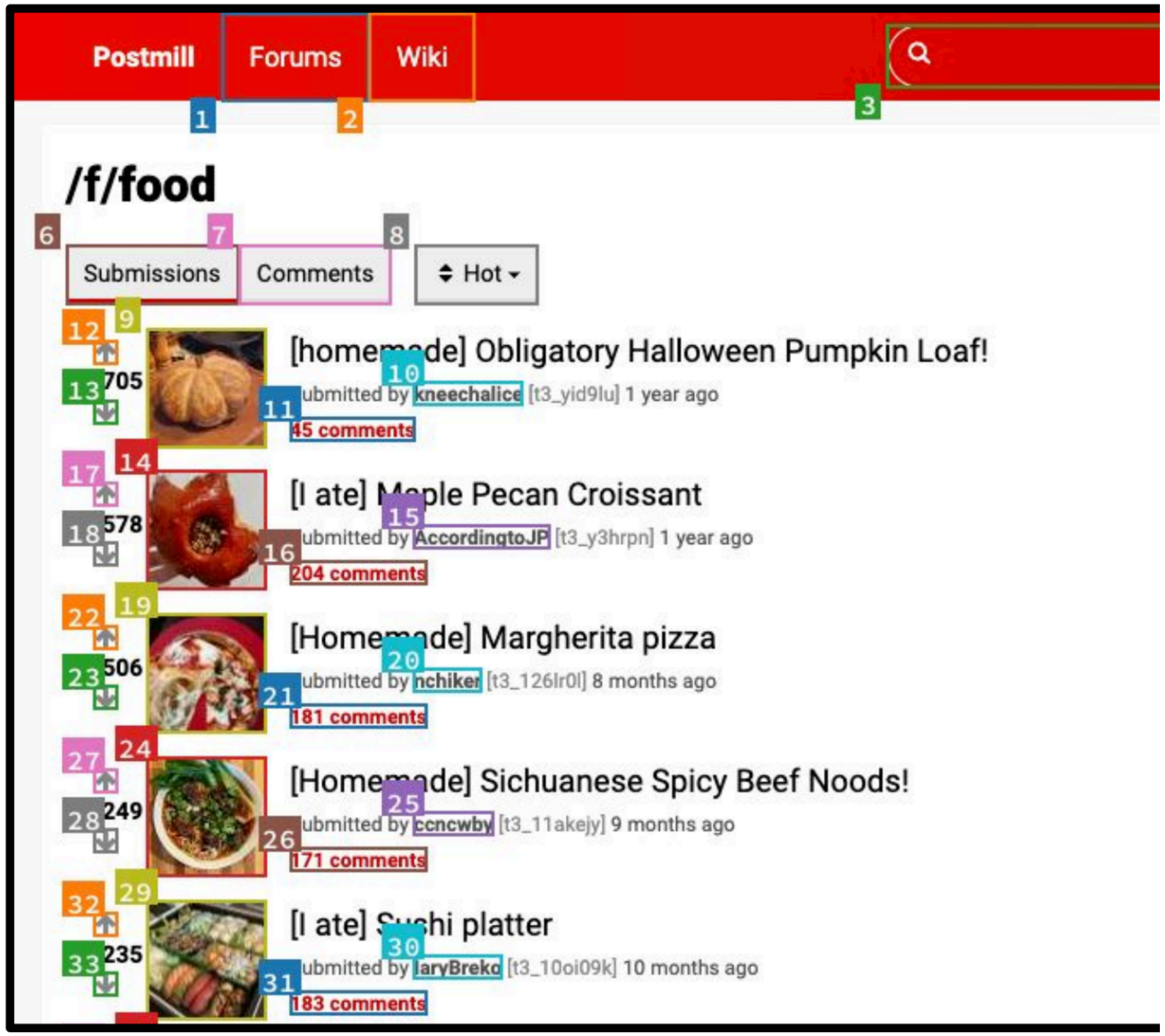
11,578 [I ate] Maple Pecan Croissant
Submitted by AccordingtoJP [t3_y3hrpn] 1 year ago
204 comments

11,506 [Homemade] Margherita pizza
Submitted by nchiker [t3_126lr0l] 9 months ago
181 comments

11,249 [Homemade] Sichuanese Spicy Beef Noods!
Submitted by ccncwby [t3_11akejy] 10 months ago
171 comments

10,235 [I ate] Sushi platter
Submitted by laryBreko [t3_10oi09k] 11 months ago
183 comments

Set-of-Marks



/f/food

705 [homemade] Obligatory Halloween Pumpkin Loaf!
ubmitted by kneechalice [t3_yid9lu] 1 year ago
45 comments

578 [I ate] Maple Pecan Croissant
ubmitted by AccordingtoJP [t3_y3hrpn] 1 year ago
204 comments

506 [Homemade] Margherita pizza
ubmitted by nchiker [t3_126lr0l] 8 months ago
181 comments

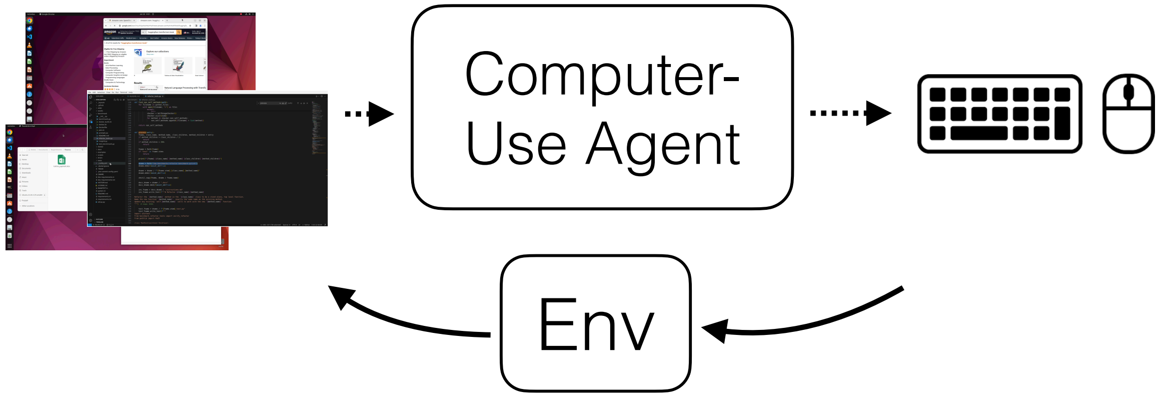
249 [Homemade] Sichuanese Spicy Beef Noods!
ubmitted by ccncwby [t3_11akejy] 9 months ago
171 comments

235 [I ate] Sushi platter
ubmitted by laryBreko [t3_10oi09k] 10 months ago
183 comments

```
...
[7] [A] [Comments]
[8] [BUTTON] [Hot]
[9] [IMG] [description: picture of a pumpkin]
[10] [A] [kneechalice]
...
```

Computer-use with basic API

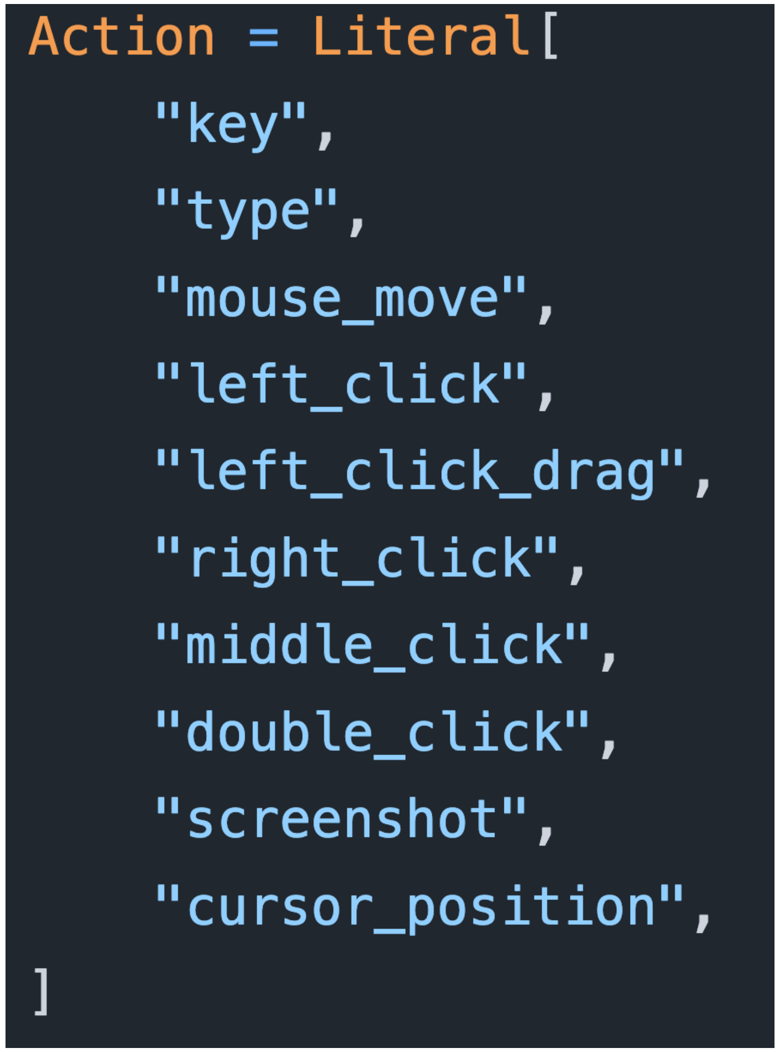
- Observations:
  - Action output
- Actions:
  - Screenshot
  - Click, type
  - File, bash operations



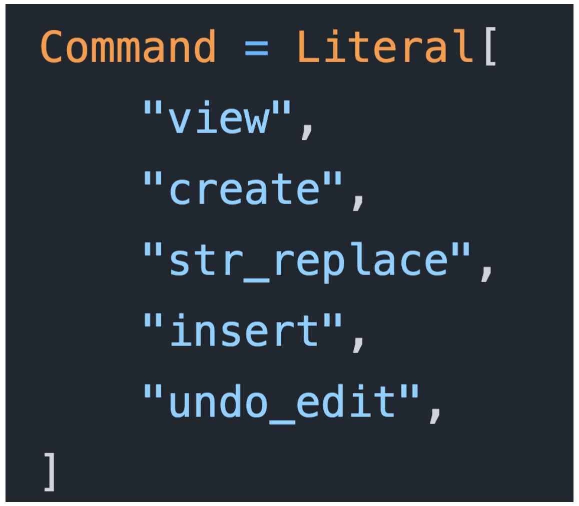
Computer-Use Agent

Env

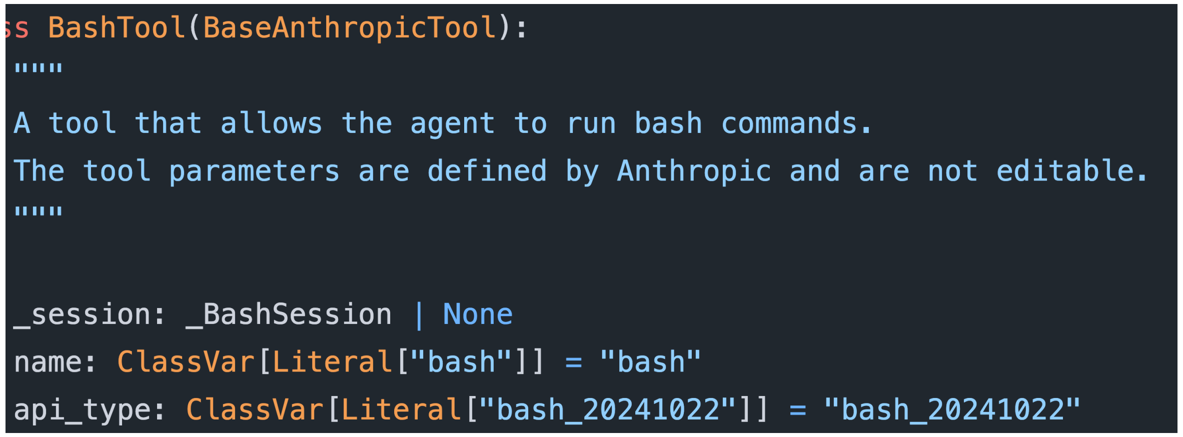
Example: Claude computer-use



```
Action = Literal[
    "key",
    "type",
    "mouse_move",
    "left_click",
    "left_click_drag",
    "right_click",
    "middle_click",
    "double_click",
    "screenshot",
    "cursor_position",
]
```



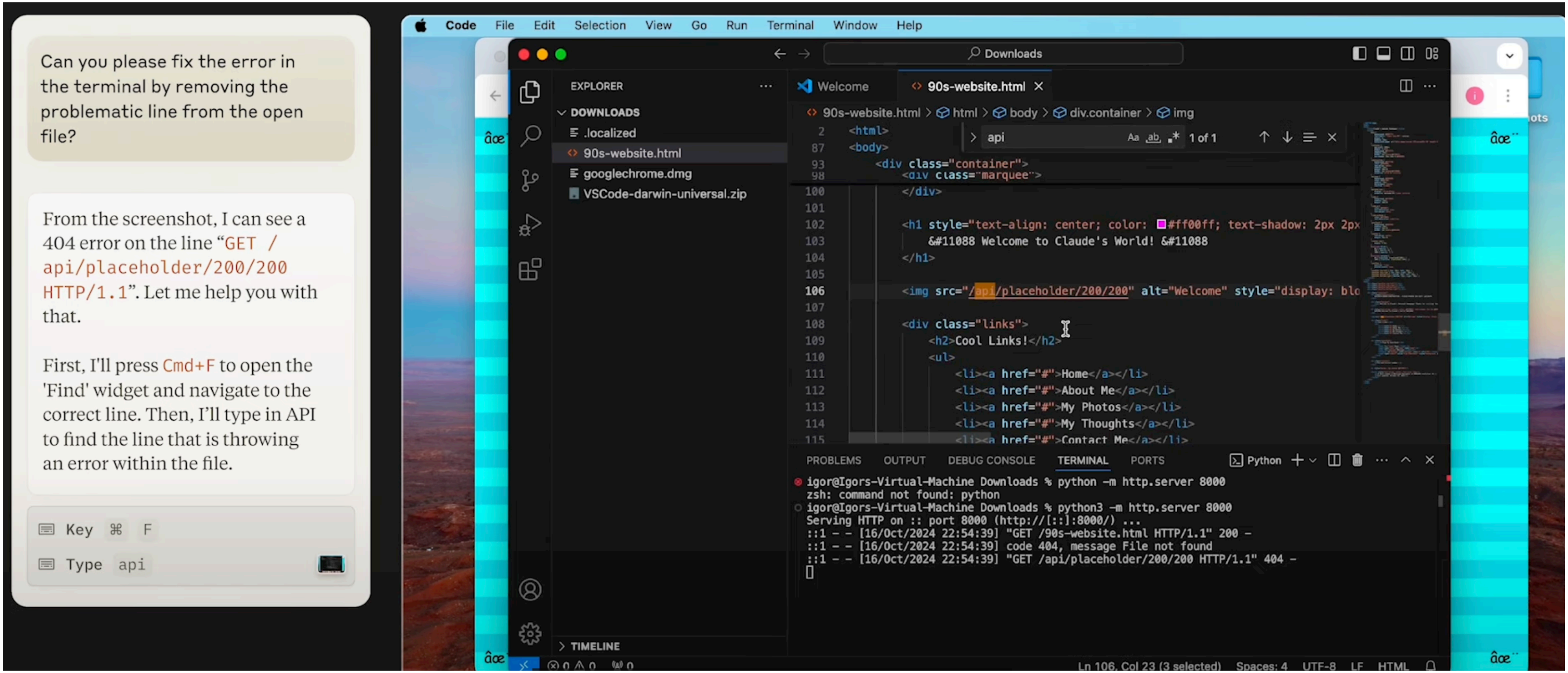
```
Command = Literal[
    "view",
    "create",
    "str_replace",
    "insert",
    "undo_edit",
]
```



```
ss BashTool(BaseAnthropicTool):
"""
A tool that allows the agent to run bash commands.
The tool parameters are defined by Anthropic and are not editable.
"""

_session: _BashSession | None
name: ClassVar[Literal["bash"]] = "bash"
api_type: ClassVar[Literal["bash_20241022"]] = "bash_20241022"
```

https://github.com/anthropics/anthropic-quickstarts/tree/main/computer-use-demo/computer_use_demo/tools

Example: Claude computer-use



Can you please fix the error in the terminal by removing the problematic line from the open file?

From the screenshot, I can see a 404 error on the line "`GET /api/placeholder/200/200 HTTP/1.1`". Let me help you with that.

First, I'll press `Cmd+F` to open the 'Find' widget and navigate to the correct line. Then, I'll type in API to find the line that is throwing an error within the file.

Key ⌘ F

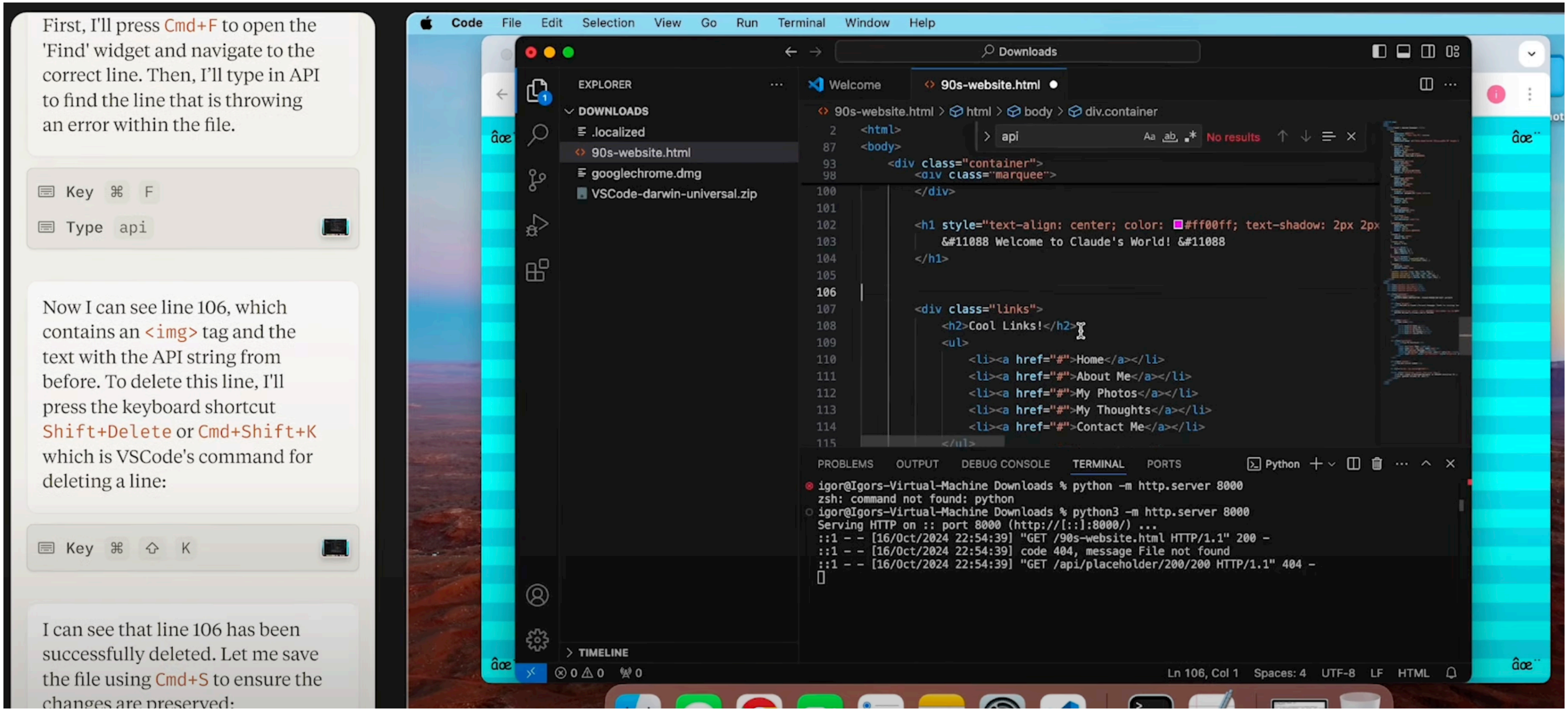
Type `api`

```html
<html>
<body>
    <div class="container">
        <div class="marquee">
        </div>

        <h1 style="text-align: center; color: #ff00ff; text-shadow: 2px 2px
            &#11088 Welcome to Claude's World! &#11088
        </h1>

        <img src="/api/placeholder/200/200" alt="Welcome" style="display: blo

        <div class="links">
            <h2>Cool Links!</h2>
            <ul>
                <li><a href="#">Home</a></li>
                <li><a href="#">About Me</a></li>
                <li><a href="#">My Photos</a></li>
                <li><a href="#">My Thoughts</a></li>
                <li><a href="#">Contact Me</a></li>
```

```
igor@Igors-Virtual-Machine Downloads % python -m http.server 8000
zsh: command not found: python
igor@Igors-Virtual-Machine Downloads % python3 -m http.server 8000
Serving HTTP on :: port 8000 (http://[::]:8000/) ...
::1 - - [16/Oct/2024 22:54:39] "GET /90s-website.html HTTP/1.1" 200 -
::1 - - [16/Oct/2024 22:54:39] code 404, message File not found
::1 - - [16/Oct/2024 22:54:39] "GET /api/placeholder/200/200 HTTP/1.1" 404 -
```

https://www.youtube.com/watch?v=vH2f7cjXjKI

Example: Claude computer-use



First, I'll press `Cmd+F` to open the 'Find' widget and navigate to the correct line. Then, I'll type in API to find the line that is throwing an error within the file.

Key ⌘ F

Type api

Now I can see line 106, which contains an `<img>` tag and the text with the API string from before. To delete this line, I'll press the keyboard shortcut `Shift+Delete` or `Cmd+Shift+K` which is VSCode's command for deleting a line:

Key ⌘ ⇧ K

I can see that line 106 has been successfully deleted. Let me save the file using `Cmd+S` to ensure the changes are preserved:

```html
2    <html>
87   <body>
93       <div class="container">
98           <div class="marquee">
100          </div>
101
102          <h1 style="text-align: center; color: #ff00ff; text-shadow: 2px 2px
103              &#11088 Welcome to Claude's World! &#11088
104          </h1>
105
106
107          <div class="links">
108              <h2>Cool Links!</h2>
109              <ul>
110                  <li><a href="#">Home</a></li>
111                  <li><a href="#">About Me</a></li>
112                  <li><a href="#">My Photos</a></li>
113                  <li><a href="#">My Thoughts</a></li>
114                  <li><a href="#">Contact Me</a></li>
115              </ul>
```

```
igor@Igors-Virtual-Machine Downloads % python -m http.server 8000
zsh: command not found: python
igor@Igors-Virtual-Machine Downloads % python3 -m http.server 8000
Serving HTTP on :: port 8000 (http://[::]:8000/) ...
::1 - - [16/Oct/2024 22:54:39] "GET /90s-website.html HTTP/1.1" 200 -
::1 - - [16/Oct/2024 22:54:39] code 404, message File not found
::1 - - [16/Oct/2024 22:54:39] "GET /api/placeholder/200/200 HTTP/1.1" 404 -
```

https://www.youtube.com/watch?v=vH2f7cjXjKI

Discussion: pros and cons

- Text-based API agent
- Vision-text API agent
- Computer use agent

Recap: Agent Basics

- What is an agent?
  - A policy that operates in an environment
  - Text/LLM, vision/VLM, computer-use agents
- Agent environments
- Agent patterns

Questions?