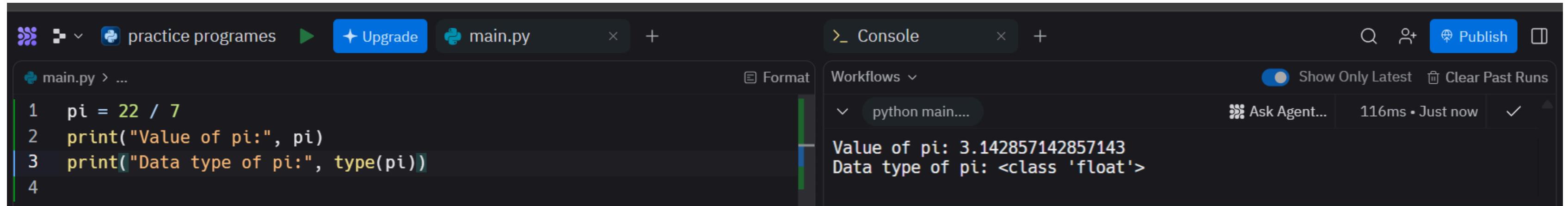


Task-1

Variables



practice programs ➔ Upgrade main.py +

Format

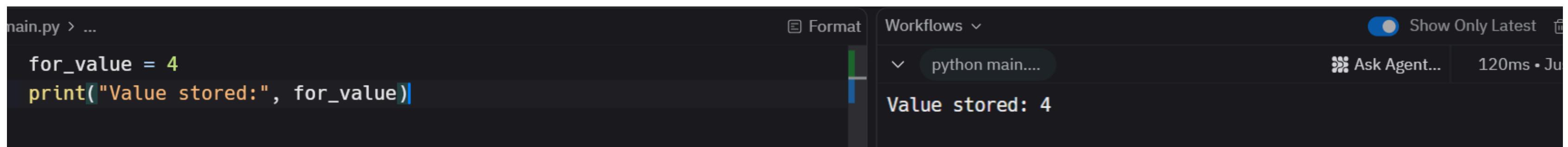
Workflows

Show Only Latest Clear Past Runs

python main.... Ask Agent... 116ms • Just now ✓

```
1 pi = 22 / 7
2 print("Value of pi:", pi)
3 print("Data type of pi:", type(pi))
4
```

Value of pi: 3.142857142857143
Data type of pi: <class 'float'>

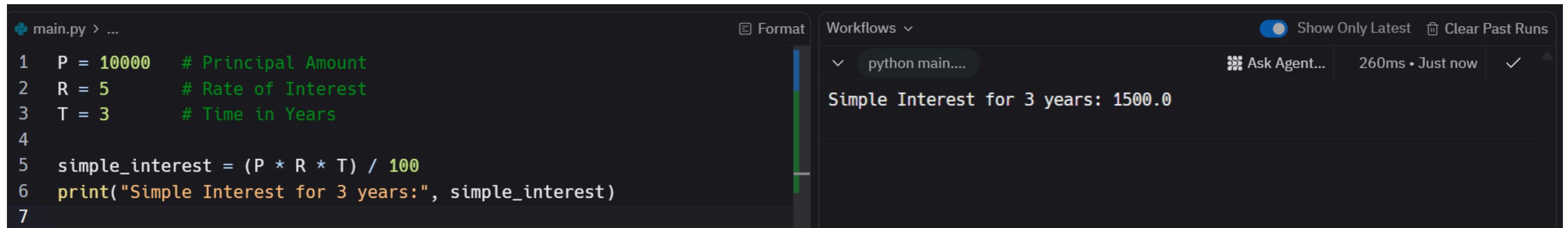


main.py > ... Format Workflows Show Only Latest Clear Past Runs

python main.... Ask Agent... 120ms • Ju

```
for_value = 4
print("Value stored:", for_value)
```

Value stored: 4



main.py > ... Format Workflows Show Only Latest Clear Past Runs

python main.... Ask Agent... 260ms • Just now ✓

```
P = 10000 # Principal Amount
R = 5 # Rate of Interest
T = 3 # Time in Years
simple_interest = (P * R * T) / 100
print("Simple Interest for 3 years:", simple_interest)
```

Simple Interest for 3 years: 1500.0

Numbers



```
main.py > ...
```

```
1 # Given values
2 distance = 490          # meters
3 time_minutes = 7
4
5 # Convert minutes to seconds
6 time_seconds = time_minutes * 60
7
8 # Calculate speed
9 speed = distance / time_seconds
10
11 # Print without decimal
12 print("Speed (m/s):", int(speed))
```

```
1 # Given values
2 distance = 490          # meters
3 time_minutes = 7
4
5 # Convert minutes to seconds
6 time_seconds = time_minutes * 60
7
8 # Calculate speed
9 speed = distance / time_seconds
10
11 # Print without decimal
12 print("Speed (m/s):", int(speed))
```

```
1 # Given values
2 distance = 490          # meters
3 time_minutes = 7
4
5 # Convert minutes to seconds
6 time_seconds = time_minutes * 60
7
8 # Calculate speed
9 speed = distance / time_seconds
10
11 # Print without decimal
12 print("Speed (m/s):", int(speed))
```

List



The screenshot shows a Python code editor with a script named `main.py`. The code performs several operations on a list of Justice League members:

- Step 0: Initial list**: Prints the initial list of members: `['Superman', 'Batman', 'Wonder Woman', 'Flash', 'Aquaman', 'Green Lantern']`.
- # 1. Number of members**: Prints the total number of members: `len(justice_league)`.
- # 2. Add Batgirl and Nightwing**: Adds `["Batgirl", "Nightwing"]` to the list.
- # 3. Make Wonder Woman the leader**: Removes `"Wonder Woman"` from the list and inserts it at index 0.

The right side of the interface shows the output of the code execution:

- Initial Justice League: `['Superman', 'Batman', 'Wonder Woman', 'Flash', 'Aquaman', 'Green Lantern']`
- Total members: 6
- After adding Batgirl and Nightwing: `['Superman', 'Batman', 'Wonder Woman', 'Flash', 'Aquaman', 'Green Lante rn', 'Batgirl', 'Nightwing']`
- After making Wonder Woman the leader: `['Wonder Woman', 'Superman', 'Batman', 'Flash', 'Aquaman', 'Green Lante rn', 'Batgirl', 'Nightwing']`
- After separating Aquaman and Flash: `['Wonder Woman', 'Superman', 'Batman', 'Flash', 'Aquaman', 'Green Lante rn', 'Batgirl', 'Nightwing']`
- After forming new Justice League: `['Cyborg', 'Shazam', 'Hawkgirl', 'Martian Manhunter', 'Green Arrow']`
- Sorted Justice League: `['Cyborg', 'Green Arrow', 'Hawkgirl', 'Martian Manhunter', 'Shazam']`
- New leader of Justice League: Cyborg

```
# 4. Separate Aquaman and Flash using Green Lantern
justice_league.remove("Green Lantern")

aquaman_index = justice_league.index("Aquaman")
justice_league.insert(aquaman_index + 1, "Green Lantern")

print("After separating Aquaman and Flash:")
print(justice_league)
print("-" * 60)
```

```
main.py > ...
32
33 # 5. Replace entire list with new team
34 justice_league = [
35     "Cyborg",
36     "Shazam",
37     "Hawkgirl",
38     "Martian Manhunter",
39     "Green Arrow"
40 ]
41 print("After forming new Justice League:")
42 print(justice_league)
43 print("-" * 60)
44
45 # 6. Sort alphabetically
46 justice_league.sort()
47 print("Sorted Justice League:")
48 print(justice_league)
49 print("-" * 60)
50
51 # New leader
52 print("New leader of Justice League:", justice_league[0])
53
54
```

Format Workflows Show Only Latest Clear Past Runs

python main.... Ask Agent... 288ms • 1 minute ago ✓

Initial Justice League: ['Superman', 'Batman', 'Wonder Woman', 'Flash', 'Aquaman', 'Green Lantern']

Total members: 6

After adding Batgirl and Nightwing:
['Superman', 'Batman', 'Wonder Woman', 'Flash', 'Aquaman', 'Green Lantern', 'Batgirl', 'Nightwing']

After making Wonder Woman the leader:
['Wonder Woman', 'Superman', 'Batman', 'Flash', 'Aquaman', 'Green Lantern', 'Batgirl', 'Nightwing']

After separating Aquaman and Flash:
['Wonder Woman', 'Superman', 'Batman', 'Flash', 'Aquaman', 'Green Lantern', 'Batgirl', 'Nightwing']

After forming new Justice League:
['Cyborg', 'Shazam', 'Hawkgirl', 'Martian Manhunter', 'Green Arrow']

Sorted Justice League:
['Cyborg', 'Green Arrow', 'Hawkgirl', 'Martian Manhunter', 'Shazam']

New leader of Justice League: Cyborg

IF Condition



```
1 # BMI Calculator Program
2
3 height = float(input("Enter height in meters: "))
4 weight = float(input("Enter weight in kilograms: "))
5
6 bmi = weight / (height ** 2)
7 print("BMI:", round(bmi, 2))
8
9 if bmi >= 30:
10     print("Obesity")
11 elif bmi >= 25:
12     print("Overweight")
13 elif bmi >= 18.5:
14     print("Normal")
15 else:
16     print("Underweight")
17
```

Ask Agent... 11s • Just now ✓

python main.... Enter height in meters: 25
Enter weight in kilograms: 12
BMI: 0.02
Underweight

```
1 # Lists of cities by country
2 australia = ["Sydney", "Melbourne", "Brisbane", "Perth"]
3 uae = ["Dubai", "Abu Dhabi", "Sharjah", "Ajman"]
4 india = ["Mumbai", "Bangalore", "Chennai", "Delhi"]
5
6 # User input
7 city1 = input("Enter the first city: ")
8 city2 = input("Enter the second city: ")
9
10 # Check if both cities belong to the same country
11 if city1 in india and city2 in india:
12     print("Both cities are in India")
13
14 elif city1 in australia and city2 in australia:
15     print("Both cities are in Australia")
16
17 elif city1 in uae and city2 in uae:
18     print("Both cities are in UAE")
19
20 else:
21     print("They don't belong to the same country")
```

python main.... Ask Agent... 17s •
Enter the first city: Anand
Enter the second city: Amdavad
They don't belong to the same country

For Loop

(Dice Roll Simulation)

The screenshot shows a Python development environment with the following details:

- File Explorer:** Shows a folder named "practice programs" containing "main.py".
- Code Editor:** The file "main.py" is open, displaying the following code:

```
1 import random
2
3 rolls = 20
4 count_6 = 0
5 count_1 = 0
6 two_sixes_in_row = 0
7 previous_roll = None
8
9 print("Dice rolls:")
10
11 for i in range(rolls):
12     current_roll = random.randint(1, 6)
13     print(current_roll, end=" ")
14
15     if current_roll == 6:
16         count_6 += 1
17     if current_roll == 1:
18         count_1 += 1
19     if previous_roll == 6 and current_roll == 6:
20         two_sixes_in_row += 1
21
22     previous_roll = current_roll
```
- Console:** Shows the output of running the script with "python main....".

```
Dice rolls:
6 2 4 3 3 2 1 3 4 6 5 3 2 2 2 2 1 2 1

Statistics:
Number of times rolled a 6: 2
Number of times rolled a 1: 3
Number of times rolled two 6s in a row: 0
```

```
print("\n\nStatistics:")
print("Number of times rolled a 6:", count_6)
print("Number of times rolled a 1:", count_1)
print("Number of times rolled two 6s in a row:", two_sixes_in_row)
```

(Jumping Jacks Workout)

The screenshot shows a code editor interface with a dark theme. On the left, the file `main.py` is open, displaying a Python script. The script initializes a total of 100 jumping jacks, performs 10 sets of 10 jacks each, and then asks if the user is tired. If the user says yes, it prints a summary and breaks the loop; otherwise, it prints the remaining jacks.

```
main.py > ...
1 total_jumping_jacks = 100
2 completed = 0
3
4 for _ in range(10):    # 10 sets of 10 jumping jacks
5     completed += 10
6     remaining = total_jumping_jacks - completed
7
8     print("\nYou completed", completed, "jumping jacks.")
9
10    if completed == total_jumping_jacks:
11        print("Congratulations! You completed the workout.")
12        break
13
14    tired = input("Are you tired? (yes/y or no/n): ").lower()
15
16    if tired in ["yes", "y"]:
17        print("You completed a total of", completed, "jumping jacks.")
18        break
19    else:
20        print(remaining, "jumping jacks remaining.")
21
```

On the right side of the interface, there is a sidebar titled "Workflows". It shows a single workflow named "python main...." with the status "25s • Just now". The workflow output is displayed below, showing the program's execution and user interaction:

You completed 10 jumping jacks.
Are you tired? (yes/y or no/n): no
90 jumping jacks remaining.

You completed 20 jumping jacks.
Are you tired? (yes/y or no/n): █