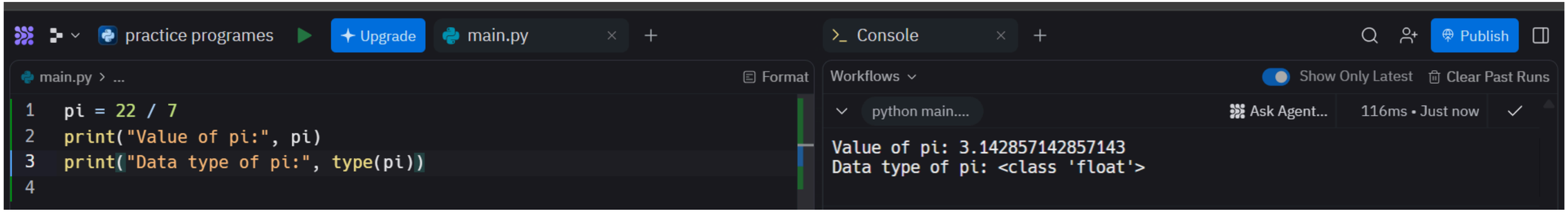
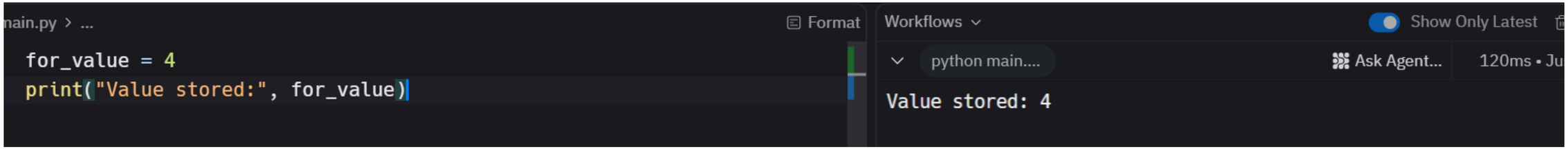
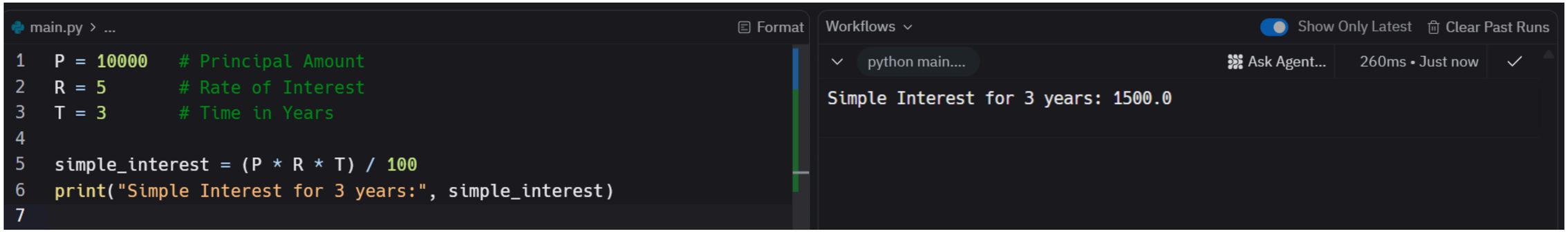
Task-1

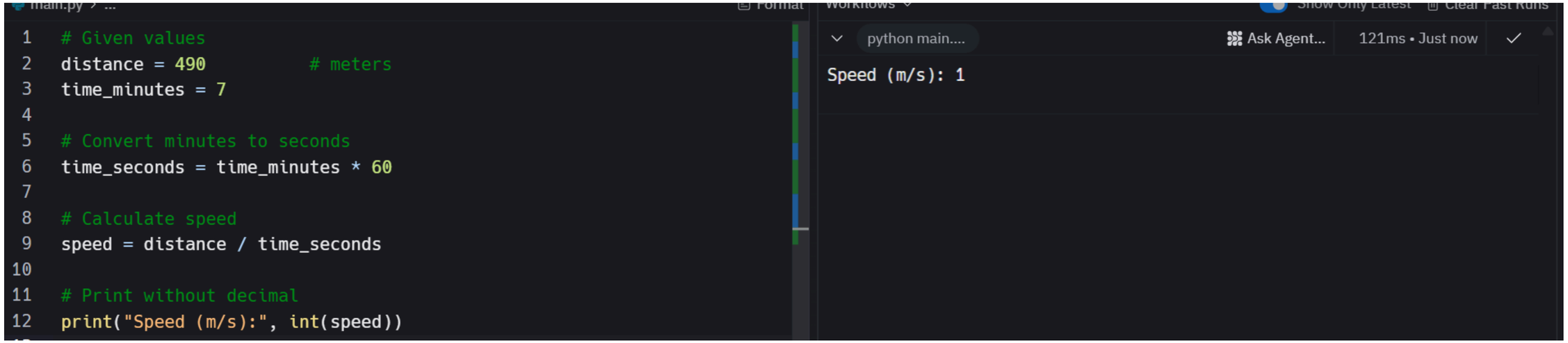
Variables


practice programes
Upgrade
main.py
Console
Publish
main.py > ...
Format
Workflows
Show Only Latest
Clear Past Runs
python main....
Ask Agent...
116ms • Just now
pi = 22 / 7
print("Value of pi:", pi)
print("Data type of pi:", type(pi))
Value of pi: 3.142857142857143
Data type of pi: <class 'float'>


main.py > ...
Format
Workflows
Show Only Latest
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 > ...
Format
# Given values
distance = 490 # meters
time_minutes = 7
# Convert minutes to seconds
time_seconds = time_minutes * 60
# Calculate speed
speed = distance / time_seconds
# Print without decimal
print("Speed (m/s):", int(speed))
python main....
Ask Agent...
121ms • Just now
Speed (m/s): 1


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
python main....
Ask Agent...
121ms • Just now
Speed (m/s): 1


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
python main....
Ask Agent...
121ms • Just now
Speed (m/s): 1


List

main.py > ... Format

```python
# Step 0: Initial list
justice_league = ["Superman", "Batman", "Wonder Woman", "Flash",
"Aquaman", "Green Lantern"]
print("Initial Justice League:", justice_league)
print("-" * 60)

# 1. Number of members
print("Total members:", len(justice_league))
print("-" * 60)

# 2. Add Batgirl and Nightwing
justice_league.extend(["Batgirl", "Nightwing"])
print("After adding Batgirl and Nightwing:")
print(justice_league)
print("-" * 60)

# 3. Make Wonder Woman the leader
justice_league.remove("Wonder Woman")
justice_league.insert(0, "Wonder Woman")
print("After making Wonder Woman the leader:")
print(justice_league)
print("-" * 60)

```

Workflows Show Only Latest Clear Past Runs

python main.... Ask Agent... 288ms • Just now

```
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
```

```python
# 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)
```

```python
# 5. Replace entire list with new team
justice_league = [
    "Cyborg",
    "Shazam",
    "Hawkgirl",
    "Martian Manhunter",
    "Green Arrow"
]

print("After forming new Justice League:")
print(justice_league)
print("-" * 60)

# 6. Sort alphabetically
justice_league.sort()
print("Sorted Justice League:")
print(justice_league)
print("-" * 60)

# New leader
print("New leader of Justice League:", justice_league[0])
```

```
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
```

IF Condition

```python
# BMI Calculator Program

height = float(input("Enter height in meters: "))
weight = float(input("Enter weight in kilograms: "))

bmi = weight / (height ** 2)
print("BMI:", round(bmi, 2))

if bmi >= 30:
    print("Obesity")
elif bmi >= 25:
    print("Overweight")
elif bmi >= 18.5:
    print("Normal")
else:
    print("Underweight")

```

python main.... Ask Agent... 11s • Just now

```
Enter height in meters: 25
Enter weight in kilograms: 12
BMI: 0.02
Underweight
```

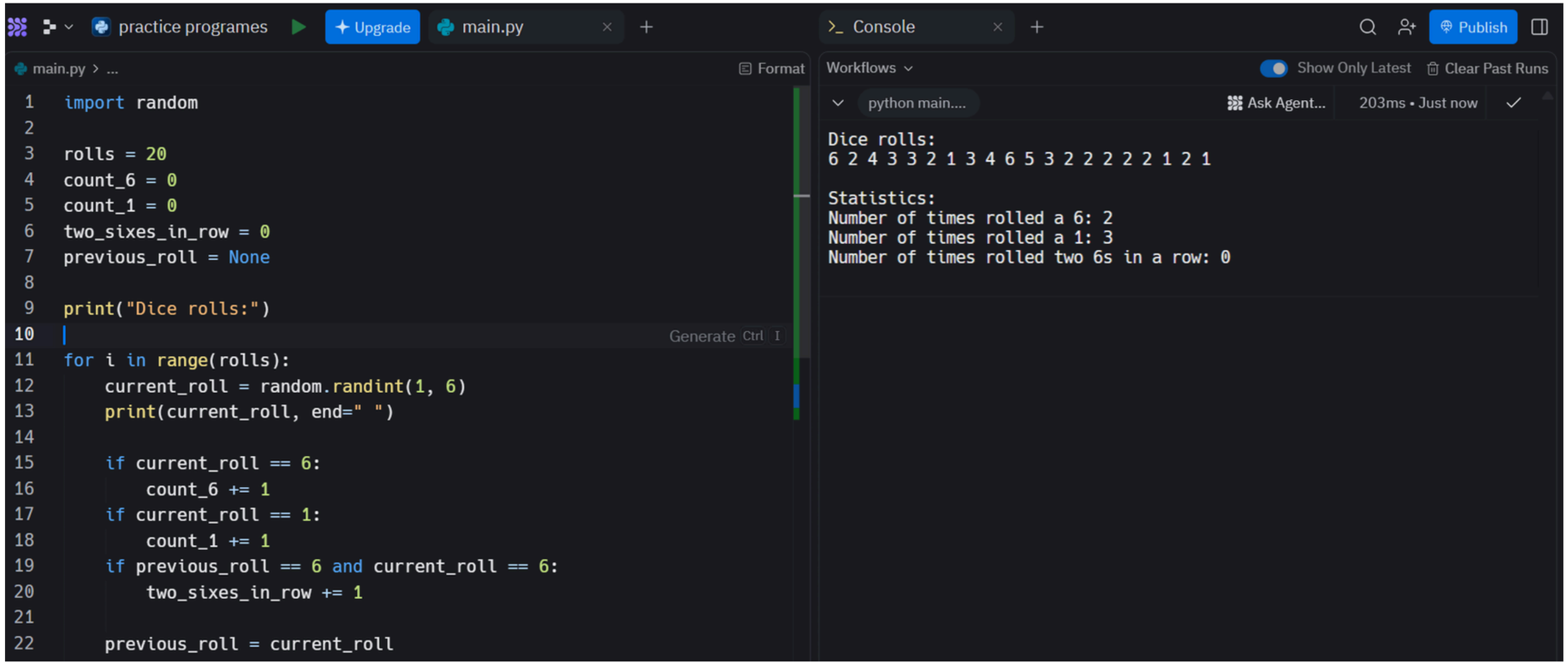
```python
# Lists of cities by country
australia = ["Sydney", "Melbourne", "Brisbane", "Perth"]
uae = ["Dubai", "Abu Dhabi", "Sharjah", "Ajman"]
india = ["Mumbai", "Bangalore", "Chennai", "Delhi"]

# User input
city1 = input("Enter the first city: ")
city2 = input("Enter the second city: ")

# Check if both cities belong to the same country
if city1 in india and city2 in india:
    print("Both cities are in India")

elif city1 in australia and city2 in australia:
    print("Both cities are in Australia")

elif city1 in uae and city2 in uae:
    print("Both cities are in UAE")

else:
    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
```

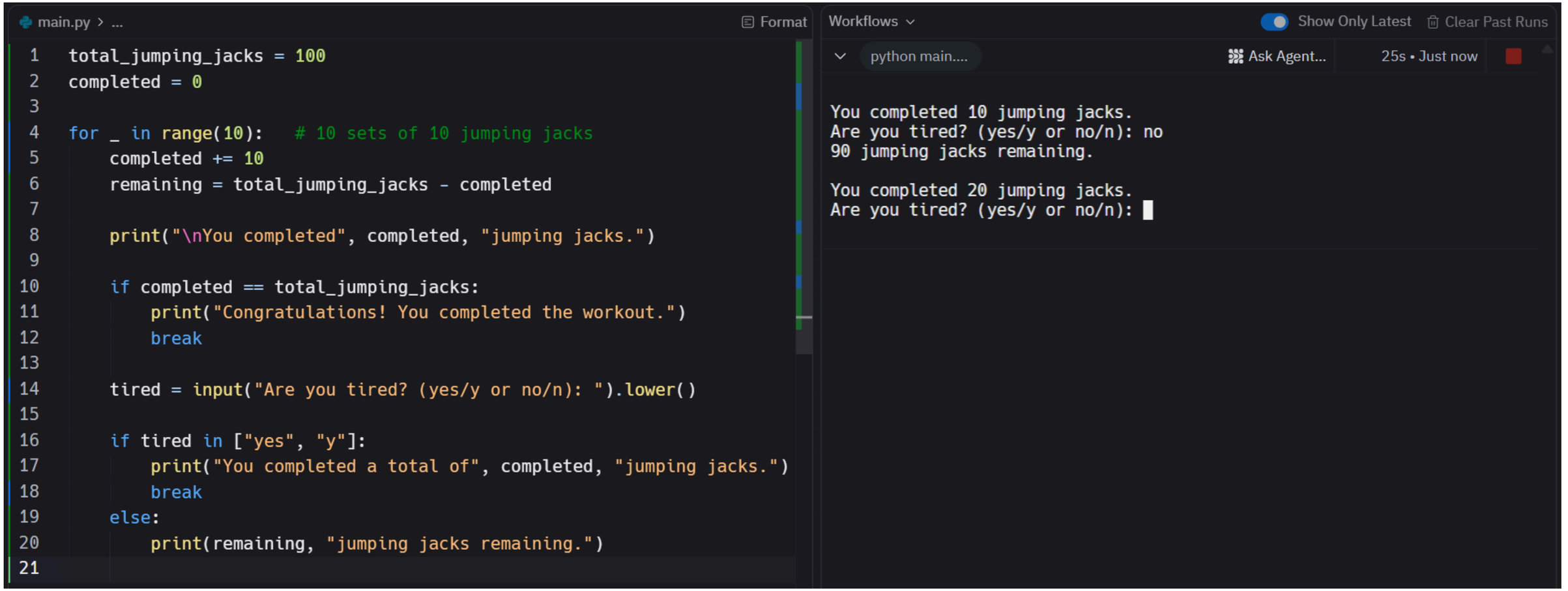
For Loop

(Dice Roll Simulation)




```
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)


main.py > ...
Format
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
Workflows
Show Only Latest
Clear Past Runs
python main....
Ask Agent...
25s • Just now
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):