

## The Next Step

The screenshot shows a web browser with multiple tabs. The active tab is 'The next step | Python'. The browser address bar shows the URL: [campus.datacamp.com/courses/intermediate-python/case-study-hacker-statistics/ex-6](https://campus.datacamp.com/courses/intermediate-python/case-study-hacker-statistics/ex-6). The page content includes a sidebar with 'Exercise' and 'Instructions' sections. The main content area shows a code editor with a Python script and a 'Run Code' button. Below the code editor is a 'Python Shell' section.

**The next step**  
Before, you have already written Python code that determines the next step based on the previous step. Now it's time to put this code inside a `for` loop so that we can simulate a random walk.

`numpy` has been imported as `np`.

**Instructions** (100 XP)

- Make a list `random_walk` that contains the first step, which is the integer 0.
- Finish the `for` loop.
- The loop should run 100 times.
- On each iteration, set `step` equal to the last element in the `random_walk` list. You can use the index `-1` for this.
- Next, let the `if - elif - else` construct update `step` for you.
- The code that appends `step` to `random_walk` is already coded.
- Print out `random_walk`.

[Take Hint \(-30 XP\)](#)

```
1 # NumPy is imported, seed is set
2
3 # Initialize random_walk
4
5
6 # Complete the for loop
7 for x in ____:
8     # Set step: last element in random_walk
9
10
11     # Roll the dice
12     dice = np.random.randint(1,7)
13
14     # Determine next step
15     if dice <= 2:
16         step = step - 1
17     elif dice <= 5:
18         step = step + 1
19     else:
20         step = step + np.random.randint(1,7)
21
22 # append next_step to random_walk
23 random_walk.append(step)
```

[Run Code](#) [Submit Answer](#)

**Python Shell** Slides

In [1]:

Below is the exercise on 'The Next Step' from the Python course. The image includes the instructions, code, and task details.

Solution:

```
# NumPy is imported, seed is set
import numpy as np
np.random.seed(123)
```

```
# Initialize random_walk
random_walk = [0]
```

```
# Complete the for loop
for x in range(100): # Loop runs 100 times
    # Set step: last element in random_walk
    step = random_walk[-1]
```

```
# Roll the dice
dice = np.random.randint(1, 7)
```

```
# Determine next step
```

```

if dice <= 2:
    step = step - 1 # Move down
elif dice <= 5:
    step = step + 1 # Move up
else:
    step = step + np.random.randint(1, 7) # Move up by a random value

# Append next_step to random_walk
random_walk.append(step)

# Print out random_walk
print(random_walk)

```

Explanation:

1. Import numpy as np and set the random seed using np.random.seed(123) to ensure reproducibility.
2. Initialize random\_walk as a list containing the first step, 0.
3. Use a for loop that runs 100 times to simulate the steps of the random walk:
  - Get the current step as the last element of the random\_walk list using random\_walk[-1].
  - Roll the dice using np.random.randint(1, 7) to generate a random integer between 1 and 6.
  - Use an if-elif-else construct to determine the next step based on the dice roll:
    - If dice is 1 or 2, decrease step by 1.
    - If dice is 3, 4, or 5, increase step by 1.
    - If dice is 6, roll the dice again and add the new result to step.
4. Append the updated step to the random\_walk list.
5. Print the final random\_walk list after completing the loop.