Binary Search: Takeaways

Syntax

- Implementing logic for binary search:

```
def player_age(name):
    name = format_name(name)
    first_guess_index = math.floor(length/2)
    first_guess = format_name(nba[first_guess_index][0])
    if name < first_guess:
        return "earlier"
    elif name > first_guess:
        return "later"
    else:
        return "found"
```

- Implementing binary search until our answer is found:

```
def player_age(name):
    name = format_name(name)
    upper_bound = length - 1
    lower_bound = 0
    index = math.floor((lower_bound + upper_bound) / 2)
    guess = format_name(nba[index][0])
    while name != guess:
        if name < guess:
            upper_bound = index - 1
        else:
            lower_bound = index + 1
        index = math.floor((lower_bound + upper_bound) / 2)
        guess = format_name(nba[index][0])
    return "found"
```

Concepts

- Binary search helps us find an item efficiently if we know the list is ordered. Binary search works by checking the middle element of the list, comparing it to the item we're looking for and repeating the process.
- Pseudo-code is a powerful, easy-to-use tool that will help you train your ability to develop and visualize algorithms. Pseudo-code comments reflect the code we want to write and describes in high-level human language.
- Pseudo-code for binary search:

```
If the name comes before our guess
    Adjust the bounds as needed
Else if the name comes after our guess
    Adjust the bounds as needed
Else
    Player found, so return first guess
```

- Binary search runs in logarithmic time, which we denote as $O(\log n)$.

Resources

- Binary search algorithm