Algorithm

Linear Search

Visual Walkthrough

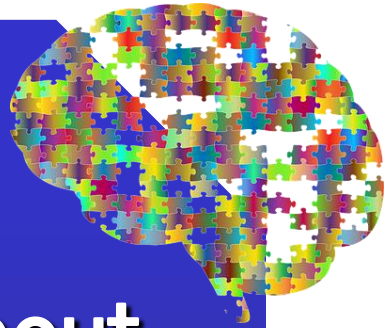
Linear Search

- **Search Problem:**

> “Find this item in this list.
> If the item is found, return the index.
> If it’s not found, return -1”.

Linear Search

- **Most intuitive** search algorithm.
- Easy to implement in Python.
- Also called “Sequential Search”.
- Use when the list contains a few elements.
- Very slow for medium and large lists.



Let's think about... Searching

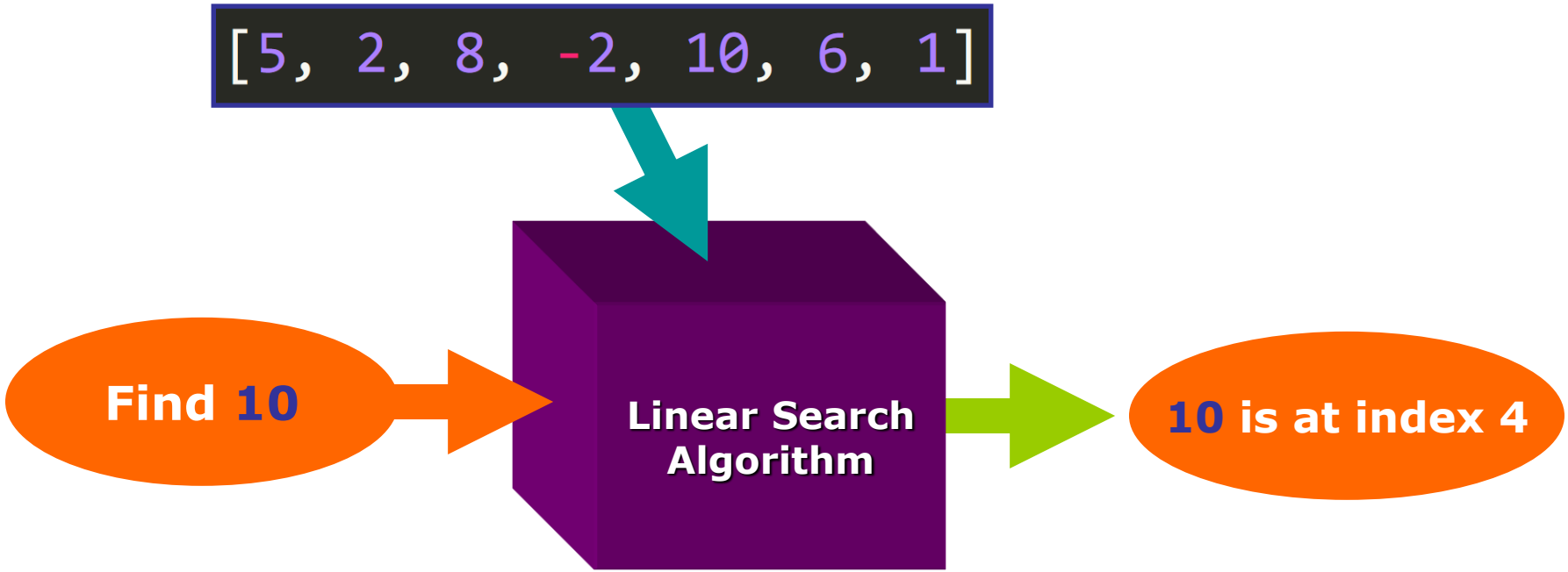
Linear Search

```
[5, 2, 8, -2, 10, 6, 1]
```

Linear Search

- Start from the first item (index 0).
- Repeat until you reach the end of the list:
  - Is this item the one that you are looking for?
  - If yes, return the current index.
  - If not, go to the next item.
- Return -1 if you reach the end of the list without finding the item.

Linear Search



Linear Search

10

```
[5, 2, 8, -2, 10, 6, 1]
```

```
Starting Linear Search Algorithm
============ Iteration #0 ============
Current list item: 5
Item searched: 10
Is the current item equal to the item searched? False
This is not the item
```

Linear Search

10

```
[5, 2, 8, -2, 10, 6, 1]
```

```
============ Iteration #1 ============
Current list item: 2
Item searched: 10
Is the current item equal to the item searched? False
This is not the item
```

Linear Search

10

```
[5, 2, 8, -2, 10, 6, 1]
```

```
============ Iteration #2 ============
Current list item: 8
Item searched: 10
Is the current item equal to the item searched? False
This is not the item
```

Linear Search

10

```
[5, 2, 8, -2, 10, 6, 1]
```

```
============ Iteration #3 ============
Current list item: -2
Item searched: 10
Is the current item equal to the item searched? False
This is not the item
```

Linear Search

10

```
[5, 2, 8, -2, 10, 6, 1]
```

```
============ Iteration #4 ============
Current list item: 10
Item searched: 10
Is the current item equal to the item searched? True
Item found! at index 4
4
```

To the Code!