

The enumerate() function in Python is a built-in function that adds a counter to an iterable and returns it as an enumerate object (which is an iterator of tuples).

Syntax:

python

enumerate(iterable, start=0)

- iterable: Any object that supports iteration (e.g., list, tuple, string).
 - start (optional): The value from which to start counting (default is 0).
-

How it works:

It returns pairs of (index, value) from the iterable.

Example 1: Basic usage

python

```
fruits = ['apple', 'banana', 'cherry']
```

```
for index, fruit in enumerate(fruits):
```

```
    print(index, fruit)
```

Output:

text

0 apple

1 banana

2 cherry

Example 2: Custom start index

python

```
for i, fruit in enumerate(fruits, start=1):
```

```
    print(i, fruit)
```

Output:

text

1 apple

2 banana

3 cherry

Example 3: Convert to list

python

```
list(enumerate(fruits))
```

```
# Output: [(0, 'apple'), (1, 'banana'), (2, 'cherry')]
```

Common use case:

Instead of:

python

```
i = 0
```

```
for item in items:
```

```
    print(i, item)
```

```
i += 1
```

Use enumerate():

```
python
```

```
for i, item in enumerate(items):
```

```
    print(i, item)
```

Much cleaner and less error-prone!

Summary:

enumerate() is a convenient way to loop over something and get both the index and the value at the same time.

You're getting this error:

```
json.decoder.JSONDecodeError: Expecting value: line 1 column 1 (char 0)
```

This error happens when Python tries to `json.load()` an EMPTY or INVALID JSON file.

In your case, the file is:

```
youtube.txt
```

and it is either empty or not valid JSON.



Why This Happens

`json.load()` expects the file to contain valid JSON like:

- `[]`
- `{}`

- [{"name": "x", "time": "1:00"}]

But your file is probably **empty**, so JSON sees:

```
" "
```

and fails at character 0.

How to Fix It

Fix 1 (Recommended): Handle Empty File

Modify your `load_data()` like this:

```
def load_data():
    try:
        with open('youtube.txt', 'r') as file:
            content = file.read().strip()
            if not content:  # file is empty
                return []
            return json.loads(content)
    except FileNotFoundError:
        return []
    except json.JSONDecodeError:
        print("File contains invalid JSON. Resetting file.")
        return []
```

This fixes:

- ✓ empty file
 - ✓ corrupted JSON
 - ✓ missing file
-

Fix 2: Manually Fix the File

Open your `youtube.txt` and put:

```
[]
```

Save it.

This ensures `json.load()` works.

Why Your Program Created an Empty File

Because when you first ran `save_data_helper` with an empty list:

```
json.dump(videos, file)
```

if `videos` was empty, JSON writer wrote:

```
[]
```

...but if the file was created manually or became corrupted, it might be empty.

Final Tip

Always add this safety check when working with JSON files, especially in CLI apps:

```
if os.path.getsize("youtube.txt") == 0:  
    return []
```