Python Tricks

Extract string Value from array :

Description:

- ❖ To extract the values from an array initialization code, we can split the string at the equal sign (=) to isolate the portion after the equal sign. Then, we strip any leading or trailing whitespace and split the resulting string at the semicolon (;) to remove any additional information after the values.
- ❖ This allows us to obtain the string containing the values.

Example:

Consider the following array initialization code:

```
array_init_code = 'int data_arr[1000] = "some values" ;'
```

We want to extract the values string 'some values'.
We can achieve this using the following code:

```
values_string = array_init_code.split('=')[1].strip().split(";")[0]
print("Extracted values string:", values_string)
```

This will output:

```
Extracted values string: "some values"
```

We can make a function to encapsulate this logic :

```
def extract_string_from_array(line):
    """
    Extracts the values string from an array.

    Args:
    - line (str): The array line extracted from a file. Example: 'int data_arr[1000] = "some values" ;'

    Returns:
    - str: The extracted values string. Example: "some values"
    - None: If an error occurs during extraction.
    """
    try:
        # Check if the line contains '=' and ';' characters
        if '=' not in line or ';' not in line:
            raise ValueError("Invalid array initialization line")
        # Check if double quotes exist in the values string
        if line.count('"') != 2:
            raise ValueError("Double quotes not properly enclosed")

        # Extract the values string between '=' and ';' characters
        values_string = line.split('=')[1].strip().split(";")[0]
        # Extract the values string between double quotes
        values_string = values_string.split('"')[1]
        return values_string
    except Exception as e:
        print("Error:", e)
        return None

line = 'int data_arr[1000] = "hello World" ;'
string_value = extract_string_from_array(line)
print("Extracted string:", string_value)
```

`'"hello World"'`

Since the string contains double quotes, it will split the string into three parts:

1- an empty string before the first double quote
2- the text 'hello World'
3- and another empty string after the second double quote.

Therefore, values_string will be ['', 'hello World', ''].

Extract Values (numbers) from array :

Description:

- To extract the values from an array initialization code in the format {1, 2, 3, 4, 5}, we can split the string at the opening curly brace {, then take the portion after that. Afterward, we split this portion again at the closing curly brace } to remove any additional information after the values. This allows us to obtain the string containing the values.
- Then, we split the values string at commas , to obtain a list of individual values.
- Finally, we iterate over this list and convert each value to an integer after stripping any leading or trailing whitespace.

Example:

Consider the following array code:

```
array_init_code = 'int data_arr[1000] = {1, 2, 3, 4, 5};'
```

We want to extract the values [1, 2, 3, 4, 5].
We can achieve this using the following code:

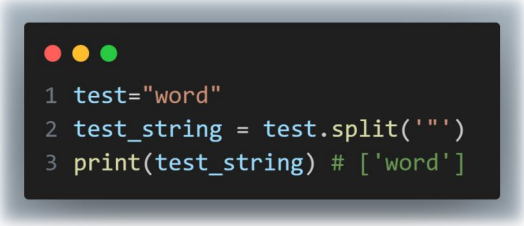
```
values_str = array_init_code.split('{', 1)[1].split('}')[0]
values = [int(val.strip()) for val in values_str.split(',')]
print("Extracted values:", values)
```

This will output:

```
Extracted values: [1, 2, 3, 4, 5]
```

We can make a function to encapsulate this logic :

```
def extract_numbers_from_array(array_init_code):
    """
    Extracts the values from an array code.

    Args:
    - line (str): The array line exctracted from a file ... EX : 'int data_arr[1000] = {1, 2, 3, 4, 5};'

    Returns:
    - list: The extracted values as a list of integers ... EX : [1, 2, 3, 4, 5]
    - None: If an error occurs during extraction.
    """
    try:
        # Check if the array initialization code contains opening and closing curly braces
        if '=' not in line or ';' not in line:
            raise ValueError("Invalid array initialization line")

        if '{' not in array_init_code or '}' not in array_init_code:
            raise ValueError("Array initialization code is missing curly braces")

        # Extract the values string between the curly braces
        values_str = array_init_code.split('{', 1)[1].split('}')[0]

        # Split the values string at commas and convert values to integers
        values = [int(val.strip()) for val in values_str.split(',')]

        return values
    except Exception as e:
        print("Error:", e)
        return None

line = 'int data_arr[1000] = {1, 2, 3, 4, 5};'
values = extract_numbers_from_array(line)
print("Extracted values:", values)
```



Explanation of Output:

The output is a list containing only one element: 'word'. This is because the original string "word" didn't have any double quotes within it to act as delimiters for splitting. Therefore, the split() method effectively returned the original string as a single element in the list.