

New AutoGrader Features

Three powerful new checking methods have been added to the AutoGrader!

1. Check List Equals

Check if a variable contains a specific list, with optional order checking.

Method: `check_list_equals()`

```
python

grader.check_list_equals(
    var_name='my_list',
    expected_list=[1, 2, 3, 4, 5],
    order_matters=True,    # Default: True
    tolerance=1e-6        # For numeric comparisons
)
```

Parameters:

- **var_name:** Name of the variable to check
- **expected_list:** The list you expect
- **order_matters:**
 - `True` (default): List must match exactly including order
 - `False`: Only elements need to match, order doesn't matter
- **tolerance:** Tolerance for floating-point comparisons

Examples:

```
python

# Exact match with order
grader.check_list_equals('numbers', [10, 20, 30, 40])

# Elements match, order doesn't matter
grader.check_list_equals('unsorted', [5, 3, 1, 4, 2], order_matters=False)

# Floating point list with tolerance
grader.check_list_equals('results', [1.5, 2.5, 3.5], tolerance=0.01)
```

Excel Configuration:

test_type	variable_name	expected_list	order_matters	tolerance	description
list_equals	my_list	[1, 2, 3, 4, 5]	true	0.0	List must equal [1,2,3,4,5]
list_equals	sorted_nums	[10, 20, 30]	false	0.0	Contains 10,20,30 any order

2. Check Array Equals

Check if a variable contains a NumPy array equal to the expected array.

Method: `check_array_equals()`

```
python
```

```
import numpy as np

grader.check_array_equals(
    var_name='data_array',
    expected_array=np.array([1.5, 2.5, 3.5]),
    tolerance=1e-6
)
```

Parameters:

- **var_name**: Name of the variable
- **expected_array**: Expected array (can be list or numpy array)
- **tolerance**: Tolerance for numeric comparisons

Features:

- ☒ Automatically converts lists to arrays
- ☒ Checks array shape (dimensions must match)
- ☒ Element-wise comparison with tolerance
- ☒ Works with multi-dimensional arrays

Examples:

```
python
```

```
# 1D array
grader.check_array_equals('vector', [1, 2, 3, 4, 5])

# 2D array
grader.check_array_equals('matrix', [[1, 2], [3, 4], [5, 6]])

# With tolerance for floating point
grader.check_array_equals('measurements', [1.0, 2.0, 3.0], tolerance=0.01)
```

Excel Configuration:

test_type	variable_name	expected_array	tolerance	description
array_equals	data_array	[1.5, 2.5, 3.5, 4.5]	0.01	Array should match values
array_equals	matrix	[[1, 2], [3, 4]]	0.0	2D array check

3. Compare with Solution File

Execute a solution file and compare student's variables with the solution's variables.

Method: `compare_with_solution()`

```
python

grader.compare_with_solution(
    solution_file
```