



DEV ACADEMY  
TE KURA HANGARAU  
O AOTEAROA

# Data Structures & Testing w/ Vitest

# Agenda

- Navigating nested data structures
  - Nested objects
  - Nested arrays
- Unit Testing with Vitest
  - Reading tests
  - Vitest
  - Anatomy of a test (*Arrange, Act, Assert*)

# Nested data structures

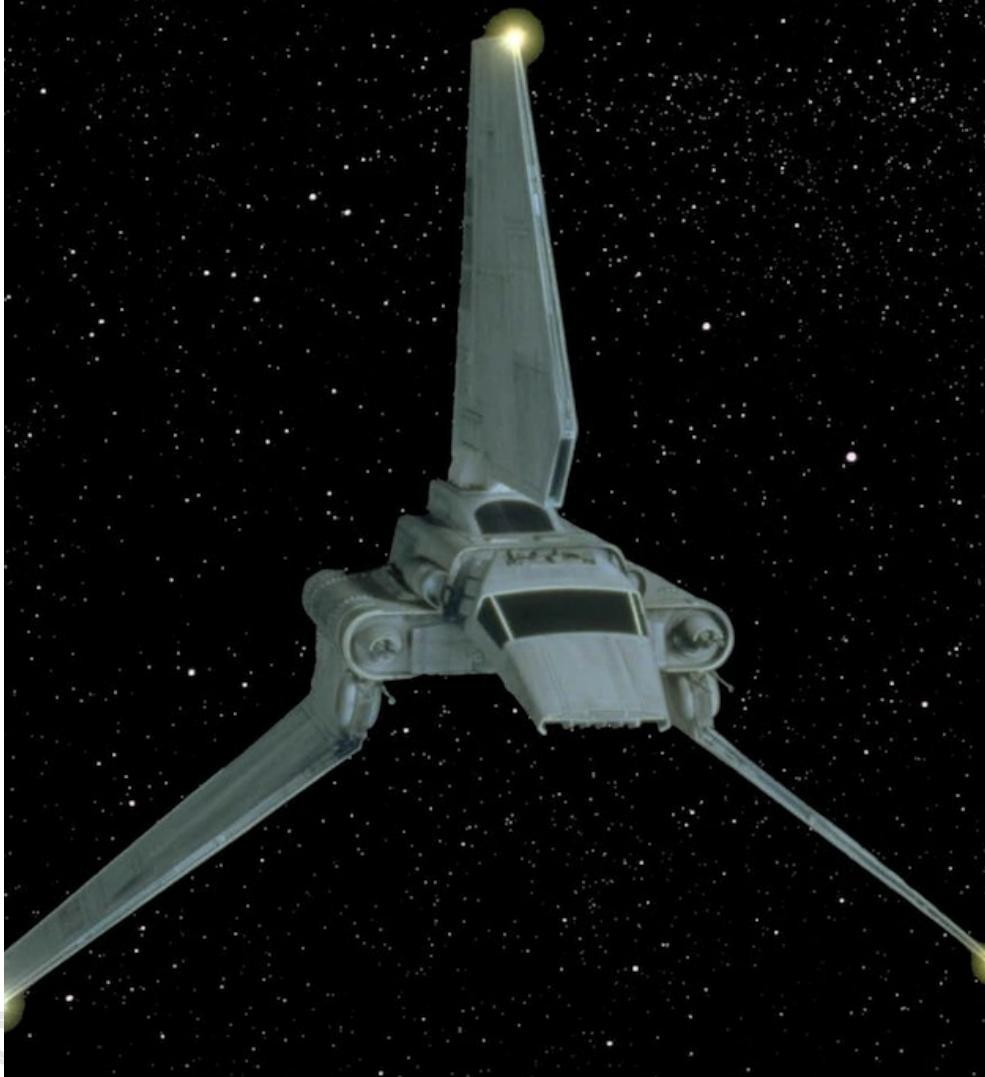
- Objects inside objects
  - ``{ a: { b: 'c' } }``
- Objects inside arrays
  - ``[ { a: 'b' }, { a: 'c' } ]``
- Arrays inside objects
  - ``{ a: [ 1, 2, 3, 4 ] }``
- Arrays inside arrays (matrices)
  - ``[ [1, 2], [3, 4] ]``



# Model data: Pilot

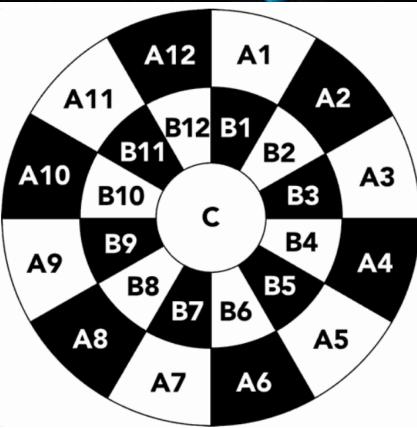
Demo

```
1  {
2    name: 'Pilot Name',
3    height: 172,          // cm
4    mass: 77,            // kg
5    birthYear: '19BBY',
6    homeworld: {
7      // ... nested object
8    },
9    vehicles: [
10   // ... nested array of objects
11 ]
12 }
```



# Model data: Dejarik Space Chess

Demo



```
1   [
2     // A1, A2, A3, A4, A5, A6, A7, A8,
3     [ null, null, 'MS', null, 'GT', null, 'MN', null,
4       // B1, B2, B3, B4, B5, B6, B7, B8,
5       [ null, 'NO', null, null, null, null, null, null, 'HJ',
6         // C
7         [ null ],
8       ]
9
10 /*
11  MS = Mantellian Savrip
12  GT = Grimtaash the Molator
13  GH = Ghhhk
14  HJ = Houjix
15  NO = Ng'ok
16  KS = Kintan strider
17  KL = K'lor'slug
18  MN = M'onnok
19 */
```



DEV ACADEMY  
TE KURA HANGARAU  
O AOTEAROA

# Testing w/ Vitest

A primer on unit testing



# Anatomy of a Test

Primer

```
1 import { getVehicleNames } from './queries.js'
2 import { describe, it, expect } from 'vitest'
3
4 describe('getVehicleNames', () => {
5     it('should return an array of vehicle names (strings)', () => {
6         // Arrange
7         const pilot = {
8             name: 'Pilot Name',
9             vehicles: [
10                 { name: 'vehicle 1' },
11                 { name: 'vehicle 2' },
12                 { name: 'vehicle 3' },
13             ]
14         }
15         const expectedOutput = ['vehicle 1', 'vehicle 2', 'vehicle 3']
16
17         // Act
18         const actualOutput = getVehicleNames(pilot)
19
20         // Assert
21         expect(actualOutput).toEqual(expectedOutput)
22     })
23 })
```

- Importing the function under test (`getVehicleNames`)
- Importing the necessary methods from the `'vitest'` library
- `'describe'` block to group tests (by name or by feature)
- `'it'` block to describe a single test
- `// Arrange`
  - Setup the test data
  - Setup the expected output
- `// Act`
  - Call the function under test to get the actual output
- `// Assert`
  - Compare the actual output with the expected output