SUDOKU VALIDATOR

Solving Solutions

TODAY'S AGENDA

- **Synthesize fundamental concepts learned so far**
  - Iteration (loops)
  - Data Types (Nested Arrays)
  - REACTO Problem Solving

R estate

E xamples

A pproach

C ode

T est

O ptimize

RESTATE

- **Rephrase in your own words (diagram if useful)**
- **Make sure you fully understand the problem**
- **Leads very naturally into…**

EXAMPLES

- **Representative input and output**
- **Consider edge cases**
- **Consider errors**
- **Write them down**

APPROACH

- **Come up with at least one *conceptual* solution**
- **Don’t code yet!**
- **Make some comments in your code file**

CODE

- **Translate your *Approach* into working JS**
- **FSA Admissions Team will even give partial credit for a solid approach (*even if the code isn't complete*)**
- **Make sure include all those edge cases!**

SUDOKU

						2		
	8				7		9	
6		2				5		
	7			6				
			9		1			
				2			4	
		5				6		3
	9		4				7	
		6						

(a) Sudoku Puzzle

9	5	7	6	1	3	2	8	4
4	8	3	2	5	7	1	9	6
6	1	2	8	4	9	5	3	7
1	7	8	3	6	4	9	5	2
5	2	4	9	7	1	3	6	8
3	6	9	5	2	8	7	4	1
8	4	5	7	9	2	6	1	3
2	9	1	4	3	6	8	7	5
7	3	6	1	8	5	4	2	9

(b) Solution

TEST

- **Use Examples in the test specs to hone your solution**
- **Ensure your Code works for all Examples**
- **Debug as necessary**

SUDOKU

- **Popular number puzzle from Japan**
- **Players fill in 9 x 9 Grid of Numbers**
- **A board is “solved” if:**
  - Numbers [1-9] are used ***only once*** per row *(no repeats!)*
  - Numbers [1-9] are used ***only once*** per column *(no repeats!)*
  - Numbers [1-9]s are used ***only once*** per 3x3 mini-grid *(no repeats!)*

SUDOKU SOLVER

- Create a function to check if a Sudoku board is valid
- Your function should return **true** if the board is valid, **false** if it isn't

SUDOKU SOLVER - RESTATE

- Create a function that takes an array of arrays of integers as an argument, representing a Sudoku Board
- My function will return a Boolean, based on whether the Sudoku solution is valid

SUDOKU SOLVER - EXAMPLES

```
var validPuzzle = [
  [ 8, 9, 5,   7, 4, 2,   1, 3, 6],
  [ 2, 7, 1,   9, 6, 3,   4, 8, 5],
  [ 4, 6, 3,   5, 8, 1,   7, 9, 2],

  [ 9, 3, 4,   6, 1, 7,   2, 5, 8],
  [ 5, 1, 7,   2, 3, 8,   9, 6, 4],
  [ 6, 8, 2,   4, 5, 9,   3, 7, 1],

  [ 1, 5, 9,   8, 7, 4,   6, 2, 3],
  [ 7, 4, 6,   3, 2, 5,   8, 1, 9],
  [ 3, 2, 8,   1, 9, 6,   5, 4, 7]
]
```

```
var invalidPuzzle = [
  [ 8, 9, 5,   7, 4, 2,   1, 3, 6],
  [ 2, 7, 1,   9, 6, 3,   4, 8, 5],
  [ 4, 6, 8,   5, 8, 1,   7, 9, 2],

  [ 9, 3, 4,   6, 1, 7,   2, 5, 8],
  [ 5, 1, 7,   2, 3, 8,   9, 6, 4],
  [ 6, 8, 2,   7, 5, 9,   3, 7, 1],

  [ 1, 5, 9,   8, 7, 4,   6, 2, 3],
  [ 7, 4, 6,   3, 2, 5,   8, 1, 9],
  [ 3, 2, 8,   1, 9, 6,   5, 2, 7]
]
```

SUDOKU SOLVER - APPROACH

- Write helper functions to get array for a specific row, column, or subsection
- Write a helper function to validate a specific row, column, or subsection
- Loop over each of the 9 rows, columns, subsections, and validate each.
  - If any isn’t valid, return **false**. Otherwise, return **true**

SUDOKU SOLVER - APPROACH

- Write helper functions to get array for a specific row, column, or subsection
- Write a helper function to validate a specific row, column, or subsection
- Loop over each of the 9 rows, columns, subsections, and validate each.
  - If any isn't valid, return **false**. Otherwise, return **true**

SUDOKU SOLVER - CODE

- When you feel confident in your approach, translate it to code

- Remember to break down complex logic into smaller helper functions

OPTIMIZE

- The final (and least important) step!
- Only if your code works and you have plenty of time
- Is there a more concise way to write this code?
- Are there built-in methods that can help?
- Did I document my code so it is easy to understand?

SUDOKU REVIEW

R estate

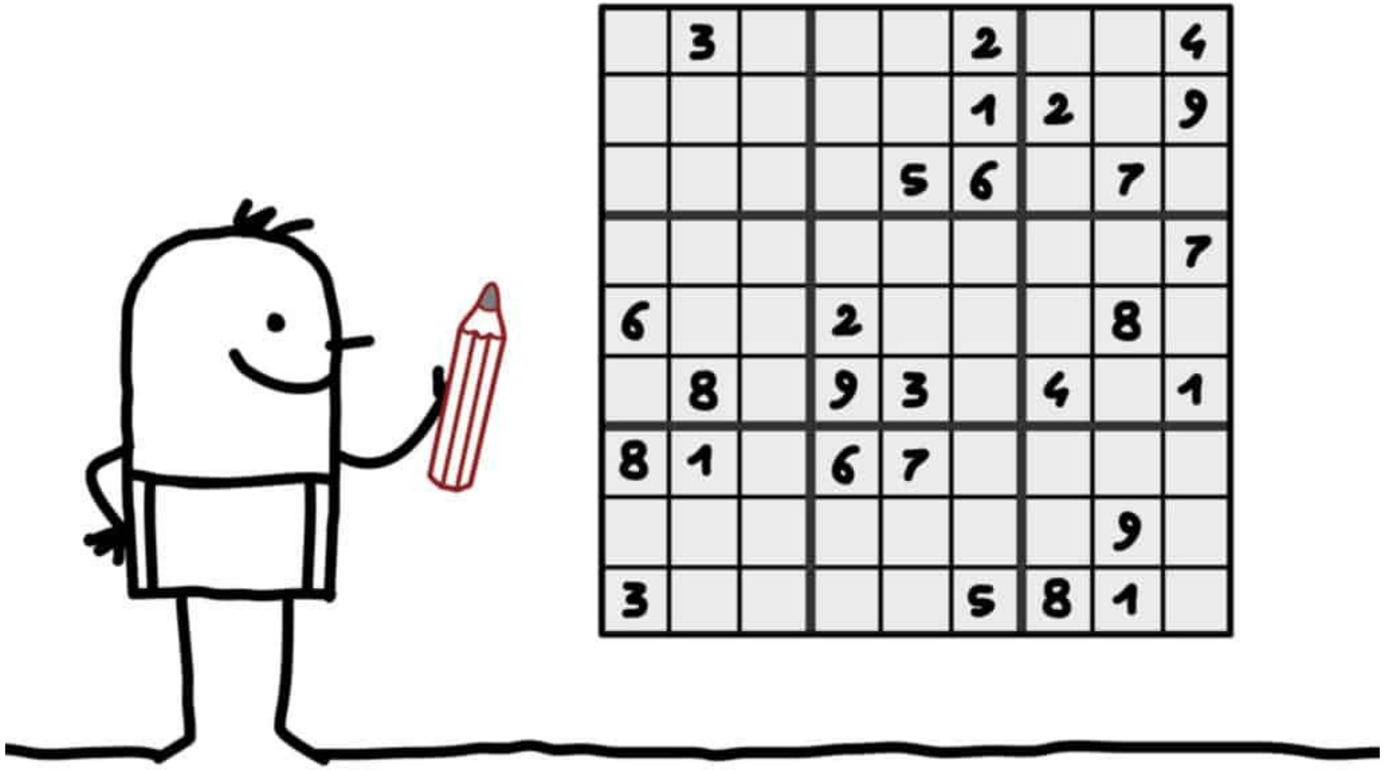
E xamples

A pproach

C ode

T est

O ptimize



	3				2			4
					1	2		9
				5	6		7	
								7
6			2				8	
	8		9	3		4		1
8	1		6	7				
							9	
3					5	8	1	