Background Story

One afternoon, after dismissals, Odessa found herself sitting in the school's computer lab with a tedious task at hand. She needed to create lists of section names to be inputted into the school's database system. Normally, this wouldn't be such a problem, but her school, based on the outskirts of Manila, had been abruptly split into multiple sections due to an increase in enrollment. She was dealing with hundreds of section names!

With her fingers getting tired from the monotony, and her mind numb from the sheer repetition, an idea sparked in Odessa's mind. She remembered her ongoing JavaScript lectures and thought, "Why not use what I've learned so far to make this task more manageable?" She decided to create a name-list generator using JavaScript 'loops'.

As she began to automate these repetitive tasks, she soon realized that the code she wrote was saving her an immense amount of time and energy. She was amazed at the power she held in her hands. Before long, her classmates started noticing her efficiency and began to approach her for help. Through this experience, Odessa gained confidence in her coding skills and started to believe that she might actually have a talent for programming.

Theory & Lecture Content

Loops in JavaScript are powerful programming structures that allow us to execute a block of code multiple times until a certain condition is met. Let's understand each kind with some examples:

1. For Loop:

```
for (initialization; condition; increment/decrement) {
    // code to be executed
}
```

```
//Example:
for (let i = 0; i < 5; i++) {
    console.log(i); // 0, 1, 2, 3, 4
}
```

2. While Loop:

```
while (condition) {
    // code to be executed
}
```

```
//Example:
let i = 0;
```

```
while (i < 5) {
    console.log(i); // 0, 1, 2, 3, 4
    i++;
}
```

3. Do...While Loop:

```
do {
    // code to be executed
} while (condition);
```

```
//Example:
let i = 0;
do {
    console.log(i); // 0, 1, 2, 3, 4
    i++;
} while (i < 5);
```

4. Nested Loops: We can put one loop inside another to iterate over multiple dimensions of data.

For additional details, please check the Mozilla Documentation.

Exercises

Exercise 1:

- Create a function that prints even numbers from 0 to 10 using a for loop.

Starter Code:

```
function printEvenNumbers() {
    //Your code here
}
```

Solution Code:

```
function printEvenNumbers() {
    for (let i = 0; i <= 10; i++) {
        if(i % 2 === 0) {
            console.log(i);
        }
    }
}
printEvenNumbers();
```

Test Cases

We can write a jest test to validate the correct implementation of the function.

```
test('print even numbers from 0 to 10', () => {
    console.log = jest.fn();
    printEvenNumbers();
    expect(console.log.mock.calls[0][0]).toBe(0);
    expect(console.log.mock.calls[1][0]).toBe(2);
    expect(console.log.mock.calls[2][0]).toBe(4);
    //... complete it till 10
});
```

Closing Story

After using loops to automate her work, Odessa was recognized for her efficiency, which boosted her confidence and motivated her to dive deeper into programming. She realized that coding wasn't just about working on abstract stuff, but also about solving real-world problems effectively, which made it exciting. 🎉🚀 What's more, she discovered a new purpose – to help her classmates with their tasks using her newfound knowledge.

Having mastered loops, Odessa is now ready to learn about a new topic which is more fun and equally important – `Objects in JavaScript`.