Strings, Numbers & the Math Object

AiCE Software Engineering Foundations — JavaScript & Node.js

You understand how JavaScript converts types and compares values. Now let's get comfortable with the two data types you'll use constantly in every program: strings and numbers — including JavaScript's built-in toolkit for working with each.

1. Creating & Measuring Strings

Strings in JavaScript can be written in single quotes, double quotes, or backticks. All three work the same way, but the convention is to use single quotes for plain text, and backticks whenever a variable needs to be inserted.

Create a file named strings.js:

```
const name = 'Chiamaka';

console.log(name.length);
console.log(name[0]);
console.log(name[name.length - 1]);

name[0] = 'B';
console.log(name);
```

Save and run:

Line	Result	Why
name.length	8	A property, not a function — no brackets after it, unlike Python's len(name).
name[0]	'C'	JavaScript uses zero-based indexing, just like Python, so the first character is at position 0.

Line	Result	Why
name[name.length - 1]	'a'	The last character, found by counting back one from the string's length.
name[0] = 'B'; then name	'Chiamaka' (unchanged)	Strings in JavaScript are immutable — once created, they cannot be changed directly.

2. String Methods

Since a string can't be changed directly, JavaScript gives methods that return a brand new string instead. Create a file named string_methods.js:

```
const bio = '  JavaScript developer from Lagos, Nigeria  ';
const email = 'adesina@aice.africa';

console.log(bio.trim());
console.log(bio.trim().toUpperCase());
console.log(bio.includes('Lagos'));
console.log(email.startsWith('adesina'));
console.log(email.endsWith('.africa'));
console.log(bio.indexOf('Lagos'));
console.log(bio.slice(2, 12));
console.log(bio.replace('Lagos', 'Abuja'));
console.log('Nigeria,Ghana,Kenya'.split(','));
```

Notice bio.trim().toUpperCase() — calling toUpperCase() right after trim(), on the same line. This is called method chaining, and it works because every method returns a new string.

Method	What It Does
trim()	Removes extra spaces from around the text.

Method	What It Does
includes(text)	Checks if one piece of text exists inside another — returns true or false.
startsWith(text) / endsWith(text)	Like includes(), but checks specifically the beginning or the end.
indexOf(text)	Returns the position where a piece of text begins, or -1 if not found.
slice(start, end)	Cuts out a portion of a string between two positions.
replace(old, new)	Swaps one piece of text for another.
split(separator)	Breaks a string apart into an array, wherever the separator is found.

3. Numbers & Math Operators

Unlike Python, which has separate int and float types, JavaScript has only one Number type for all numbers, whole or decimal. The usual math operators work the same way you already know — plus, minus, multiply, divide, remainder with %, and power with **. The one difference: Python's // for floor division doesn't exist in JavaScript — Math.floor() is used instead.

JavaScript also gives two operators Python doesn't have — ++ to increase a number by one, and -- to decrease it by one. Create a file named numbers.js:

```
let count = 0;
count++;
console.log(count);
count--;
console.log(count);

const a = 50, b = 8;
console.log(a % b);
console.log(a ** 2);
```

Save and run: count++ increases count from 0 to 1, and count-- brings it back down to 0. a % b gives 2, the remainder when 50 is divided by 8, while a ** 2 gives 2500 — 50 raised to the power of 2.

4. The Math Object

JavaScript also gives a built-in Math object, which does the same job as Python's math module — except it never needs importing; it's always available. Create a file named math_object.js:

```
console.log(Math.round(4.5));
console.log(Math.ceil(4.1));
console.log(Math.floor(4.9));
console.log(Math.abs(-50));
console.log(Math.sqrt(144));
console.log(Math.min(10, 5, 8));
console.log(Math.max(10, 5, 8));
console.log(Math.random());
```

Method	What It Does
Math.round()	Rounds to the nearest whole number.
Math.ceil()	Always rounds up.
Math.floor()	Always rounds down.
Math.abs()	Removes the negative sign from a number.
Math.sqrt()	Gives the square root.
Math.min() / Math.max()	Gives the smallest or largest number out of a list.
Math.random()	Gives a random decimal number between 0 and 1.

A common trick combines Math.random() with Math.floor() to get a random whole number in a range:

```
const randomNum = Math.floor(Math.random() * 10) + 1;
console.log(randomNum);
```

This gives a random whole number between 1 and 10.

Two more methods are useful for formatting numbers nicely for display:

```
const fee = 45000.789;
console.log(fee.toFixed(2));
console.log(fee.toLocaleString());
```

toFixed(2) rounds a number to two decimal places and gives it back as a string. toLocaleString() adds comma formatting — very useful when displaying money.

5. Practice: A Fee Calculator

Let's build a small project combining strings and numbers. Create a file named fee_calculator.js:

```
// AiCE Course Fee Calculator
const courses = [
    { name: 'Python Programming', fee: 45000 },
    { name: 'JavaScript & Node.js', fee: 60000 },
];
const discountRate = 0.10;

console.log('AiCE COURSE FEES');
courses.forEach(course => {
    const discounted = course.fee * (1 - discountRate);
    console.log(`${course.name} -
₦${course.fee.toLocaleString()} ->
₦${discounted.toLocaleString()}`);
});
```

Save and run — this calculates and displays each course fee, along with the discounted price, nicely formatted with commas. Push everything built in this lesson to GitHub.

Key Takeaways

Strings are immutable in JavaScript, and every string method — trim(), includes(), slice(), replace(), split(), and more — returns a brand new string rather than modifying the original. JavaScript merges Python's int and float into a single Number type, and the built-in Math object handles rounding, absolute value, square roots, min/max, and random numbers without ever needing an import. The next lesson moves into JavaScript's two core data structures — Arrays and Objects — starting with everything an array can hold and every method used to add, remove, slice, and sort what's inside it.