7.3 JS Arithmetic

mathematician - answered by April Joy Ipac

- As with algebra, you can do arithmetic with JavaScript variables, using operators like `=` and `+`:

  ```
  let x = 5 + 2 + 3;
  ```

- Assign values to variables and add them together:

  ```
  let x = 5; // assign the value 5 to x
  let y = 2; // assign the value 2 to y
  let z = x + y; // assign the value 7 to z (5 + 2)
  ```

Arithmetic operators

- **Arithmetic operators** are used to perform arithmetic on numbers:

Operator	Description
`+`	Addition
`-`	Subtraction
`*`	Multiplication
`**`	Exponentiation
`/`	Division
`%`	Modulus (Division Remainder)
`++`	Increment

`--`	Decrement

Adding JavaScript strings

- The + operator can also be used to add (concatenate) strings.

```
let text1 = "John";
let text2 = "Doe";
let text3 = text1 + " " + text2;
```

  - The result of `text3` will be `John Doe`

- You can also add strings and numbers together, but be careful of the results!

```
let x = 16 + "Volvo";
```

  - Does it make any sense to add `"Volvo"` to 16? Will it produce an error or will it produce a result?
  - JavaScript will treat the example above as:

```
let x = "16" + "Volvo";
```

  - When adding a number and a string, JavaScript will treat the number as a string.

```
let x = 16 + "Volvo";
let x = "Volvo" + 16;
```

  - JavaScript evaluates expressions from left to right. Different sequences can produce different results:

```
let x = 16 + 4 + "Volvo";
```

  - The result of this would be:

```
20Volvo
```

  - On the other hand, if we run the following code:

```
let x = "Volvo" + 16 + 4;
```

  - The result of this would be:

```
Volvo164
```

  - In the first example, JavaScript treats 16 and 4 as numbers, until it reaches `"Volvo"`.
  - In the second example, since the first operand is a string, all operands are treated as strings.

Assignment Operators

- **Assignment operators** assign values to JavaScript variables.

Operator	Example	Same As
`=`	`x = y`	x = y
`+=`	`x += y`	x = x + y
`-=`	`x -= y`	x = x - y
`*=`	`x *= y`	x = x * y
`/=`	`x /= y`	x = x / y
`%=`	`x %= y`	x = x % y
`**=`	`x **= y`	x = x ** y

- Assignment operators work by updating the value of a variable, rather than performing an operation first like in arithmetic operators.
- For example. the **addition assignment** operator (`+=`) adds a value directly to a variable.

```
let x = 10;
x += 5; //the value of x is now 15.
```

Operator Precedence

- Operator precedence describes the order in which operations are performed in an arithmetic expression.
- Multiplication (`*`) and division (`/`) have higher precedence than addition (`+`) and subtraction (`-`).
- As in traditional mathematics, multiplication is done first:

```
let x = 100 + 50 * 3;
```

- As in traditional mathematics, the precedence can be changed by parentheses:

```
let x = (100 + 50) * 3;
```

- When using parentheses, the operations inside the parentheses are computed first.
- When many operations have the same precedence (like addition and subtraction), they are computed from left to right:

```
let x = 100 + 50 - 3;
```

- Check out **this reference material (https://kodego.instructure.com/courses/379/pages/ref-number-1-operator-precendence-list)** to view the complete JavaScript operator precedence list!

Additional Material

- **Learn more**
  - W3Schools