



Programming & Artificial Intelligence

KERIO PLATFORM GUIDE • TERM 1 • 2026/2027

 Learn to code today ... build the future tomorrow



Simple, creative explanation of every lesson, supported by examples and illustrations



Interactive, hands-on coding examples in an engaging, student-friendly style



Smart coding drills and challenges that sharpen your skills step by step



Builds logical & algorithmic thinking and real-life problem-solving skills

Prepared & supervised by our team of educators and developers:

Eng. Sayed Samy Afandy

Sayed Afandy Educational Platform | Sayed Afandy Academy

 01030745596

Scan the QR code

to visit our learning platform
sayedafandy.com





Writing Code in the Editor (Console Output)

Lesson 1

The simplest example: printing text in the editor and the console

example1.js

```
console.log("Hello");
```

The `console.log()` command is responsible for outputting text and displaying messages inside the console.

Example 1: How do we print Good Morning?

greet.js

```
console.log("GoodMorning");
```

 **Output:** GoodMorning

Example 2: Code Comments

If you want a word or sentence **not** to run and only act as an explanation, put `//` at the start of the line:

comment.js

```
// Greetings for the day.
```

 No output appears, because the editor completely ignores comments.

Second: Math Problems and Arithmetic Operations

To write a math problem and show its result directly in the console, we write the operation **without** quotation marks:

math.js

```
console.log(2 + 2);
```

 **The output is:** 4

Exam Trap: The Quotation Marks Trap

Don't fall for it!

 **Wrong:** `console.log("2 + 2");` → prints "2 + 2"

 **Correct:** `console.log(2 + 2);` → prints 4

 **Coding secret:** Quotation marks " " make the browser print the text literally; without them, it calculates the math operation instantly!

The Four Basic Operations

Addition, subtraction, multiplication and division are called **the four basic arithmetic operations**.

Addition (+)

Subtraction (-)

Multiplication (*)

Division (/)



Calculation Methods (1) – The Four Arithmetic Operations

Lesson 2

1 2 3 4 The Four Arithmetic Operations (The Four Operations):



The four basic arithmetic operations are: **addition**, **subtraction**, **multiplication** and **division**.

Arithmetic operators (Operators): the symbols used in programming to carry out mathematical calculations:

+ Addition**- Subtraction***** Multiplication****/ Division**

+ First: Addition

+ sign

We use the plus sign (+) to do addition. Let's make the computer calculate $432 + 345$:

```
addition.js
console.log(432 + 345);
```

⚡ Result in the console: 777

— Second: Subtraction

- sign

We use the minus sign (-) to do subtraction. Let's make the computer calculate $100 - 50$:

```
subtraction.js
console.log(100 - 50);
```

⚡ Result in the console: 50

🚨 Exam Trap: Forbidden Calculator Symbols

Don't fall for it!

✗ **Wrong:** `console.log(100 × 5 ÷ 2);` → `SyntaxError`

✓ **Correct:** `console.log(100 * 5 / 2);` → prints 250

💡 **Coding secret:** In programming we use the asterisk * for multiplication and the slash / for division. The paper math symbols (× and ÷) are rejected by the code engine!

⚠ Important Warning – A Golden Note for the Young Programmer:



Multiplication and division in programming use symbols that are different from the usual math symbols in your notebook:

- We use the asterisk symbol (*) for multiplication (not the × sign).
- We use the slash (/) for division (not the ÷ sign).



Concatenation (1) – Joining Text with Arithmetic Operations

Lesson 3

💡 Look at these three separate lines (the traditional way):

separate.js

```
console.log("Set of 2 coffees");  
console.log(250 * 2);  
console.log("codycoins");
```

Output on 3 separate lines:

```
Set of 2 coffees  
500  
codycoins
```

✨ Combining Text with Arithmetic in One Line (Concatenation)



We use the plus sign (+) to join text with math formulas in one neat, short line:

📌 The general rule for concatenation:

syntax.js

```
console.log("character " + (formula) + " character");
```

★ The same result as before, but in just one line of code:

combined.js

```
console.log("Set of 2 coffees " + (250 * 2) + " codycoins");
```

⚡ Result in one line: Set of 2 coffees 500 codycoins

🚨 Exam Trap: The Number + Text Concatenation Trap

Don't fall for it!

❌ Wrong: `console.log("Total: " + 10 + 20);` → "Total: 1020"

✅ Correct: `console.log("Total: " + (10 + 20));` → "Total: 30"

💡 Coding secret: The + sign next to text acts like a magnet that glues words together. To do the math addition first, always put the operation inside its own brackets ()!

🎉 A Smart Coding Benefit:



Instead of writing 3 separate print commands that waste lines and space, we joined the sentence with the result in just one `console.log()` command! This makes your code faster, cleaner and more professional.



Concatenation (2) – Remainder and Order of Operations

Lesson 4

Division Remainder (Modulus Operator %):



In programming languages the (%) sign does **not** mean percentage; it means **the remainder of whole-number division**:

```
modulus.js  
  
console.log(10 % 3);
```

 **Remainder: 1**

 Explanation: $10 \div 3 = 3$ with a remainder of 1 (because $3 \times 3 = 9$, and $10 - 9 = 1$).



The Order of Operations Pyramid (Order of Precedence):

 Priority 1**Brackets ()** Priority 2**Multiplication * and
Division /** Priority 3**Addition + and
Subtraction -**

Case 1: Without brackets

```
p1.js  
  
console.log(2 + 3 * 4);
```

 **Output: 14**

Multiply $3*4 = 12$, then $12 + 2 = 14$

Case 2: With brackets

```
p2.js  
  
console.log((2 + 3) * 4);
```

 **Output: 20**

Brackets first: $(2+3) = 5$, then $5*4 = 20$

Equal Priorities (Left to Right)

When multiplication meets division, or addition meets subtraction, the computer carries out the operation that comes first from the left:

$10 / 2 * 3 = 15$

The % Secret for the Smaller Number

If the first number is smaller than the second, the result is always the first number itself: $3 \% 10 = 3$

and $5 \% 8 = 5$!

The Smart Programmer's Challenge:



What is the expected output of running the following command in JavaScript? Explain why.

```
console.log(10 + 5 * 2 - (4 / 2));
```

Model solution: Brackets first ($4/2 = 2$), then multiplication ($5*2 = 10$), so it becomes: $10 + 10 - 2 = 18$.