Input, Conditions, Loops, and Switch in JS

These notes cover JavaScript input methods, conditional statements, loops, and switch statements with explanations and real-world examples.

1. Input Statements

JavaScript can take input from users using prompt() in browsers or readline-sync in Node.js. Inputs are often strings, so they need to be converted to numbers using Number() or parseInt().

Example Code:

```
let name = prompt("Enter your name:");
console.log("Hello, " + name);
```

Real-world Example:

When creating a login form or signup page, JavaScript takes user input such as username and password from text fields to validate or store data.

2. Conditional Statements

Conditional statements allow the program to make decisions based on conditions. They help execute certain blocks of code when a condition evaluates to true.

Example Code:

```
let age = 18;
if (age >= 18) {
  console.log("You are eligible to vote.");
}
```

Real-world Example:

Online age verification systems use conditionals to check if users meet age requirements for services such as YouTube or social media accounts.

3. if Statement

The 'if' statement runs code only when a condition is true. It’s the simplest form of decision-making in JavaScript.

Example Code:

```
let temperature = 30;
if (temperature > 25) {
```

```
  console.log("It’s a hot day!");
}
```

Real-world Example:

Weather apps use if conditions to determine which message to show (e.g., 'It’s hot today!' or 'It’s cold today!').

4. if...else Statement

if...else provides an alternative path if the condition is false.

Example Code:

```
let isLoggedIn = false;
if (isLoggedIn) {
  console.log("Welcome back!");
} else {
  console.log("Please log in first.");
}
```

Real-world Example:

Used in login systems where the website either welcomes the user or redirects them to a login page.

5. else if Statement

Used to check multiple conditions one after another. JavaScript executes the first true condition and skips the rest.

Example Code:

```
let marks = 85;
if (marks >= 90) {
  console.log("A Grade");
} else if (marks >= 75) {
  console.log("B Grade");
} else {
  console.log("C Grade");
}
```

Real-world Example:

Used in grading systems to determine grades based on student marks.

6. Ternary Operator

A shorthand form of if...else for simple conditions. Returns a value based on the condition.

Example Code:

```
let age = 20;
let message = (age >= 18) ? "Adult" : "Minor";
console.log(message);
```

Real-world Example:

Used to show quick decisions like showing 'Available' or 'Out of stock' for products.

7. Loops Overview

Loops allow repeated execution of code blocks until a condition is false. They reduce redundancy and simplify repetitive tasks.

Real-world Example:

Used in dashboards or reports where multiple data entries (like products or users) need to be displayed repeatedly.

8. for Loop

Used when the number of iterations is known. Commonly used to iterate through arrays or numeric ranges.

Example Code:

```
for (let i = 1; i <= 5; i++) {
  console.log(i);
}
```

Real-world Example:

Used to loop through a product list to display each product name on an e-commerce website.

9. while Loop

Executes code as long as the condition remains true.

Example Code:

```
let i = 1;
while (i <= 5) {
  console.log(i);
```

```
  i++;
}
```

10. do...while Loop

Executes the block once before checking the condition. Ensures at least one execution.

Example Code:

```
let i = 1;
do {
  console.log(i);
  i++;
} while (i <= 5);
```

11. Nested Loops

A loop inside another loop. Useful for multidimensional data like grids or matrices.

Example Code:

```
for (let i = 1; i <= 3; i++) {
  for (let j = 1; j <= 2; j++) {
    console.log(`i=${i}, j=${j}`);
  }
}
```

Real-world Example:

Used for rendering tables, calendars, or 2D game maps.

12. Loop Control Statements

Used to alter loop behavior. 'break' stops the loop, 'continue' skips the current iteration.

Example Code:

```
for (let i = 1; i <= 5; i++) {
  if (i === 3) continue;
  console.log(i);
}
```

Real-world Example:

Used to skip invalid entries or stop execution once a condition is met, such as finding a specific product in inventory.

13. switch Statement

The switch statement executes different code blocks based on specific case matches.

Example Code:

```
let day = 3;
switch(day) {
  case 1: console.log("Monday"); break;
  case 2: console.log("Tuesday"); break;
  case 3: console.log("Wednesday"); break;
  default: console.log("Invalid day");
}
```

Real-world Example:

Used to display different content based on the day of the week.

14. Practice Challenge

- Print even numbers between 1 and 20 using loops.
- Create a grading system using if...else if.
- Use switch to map numbers to month names.
- Ask for user input using prompt() and display personalized messages.