Lesson Content

VI Video Course - Lessons 6 - 7

OT Reading Material

Javascript: Boolean, If-Statements, and Functions

I. Introduction to Logical Programming

Explain: Programming involves making decisions based on conditions, enabling dynamic and interactive behavior. JavaScript achieves this with booleans, if-statements, and functions.

- Demo: Log examples of boolean values directly in the browser console:

```
console.log(true); // Output: true
console.log(false); // Output: false
```

II. Booleans: The Core of Logical Decisions Explain: Booleans represent true or false, often derived from comparisons or logic.

Comparison Operators

| Operator | Meaning | Example | Result | |---------|---------|---------|---------| | > | Greater than | `10 > 5` | `true` | | < | Less than | `2 < 8` | `true` | | >= | Greater than or equal to | `7 >= 7` | `true` | | <= | Less than or equal to | `4 <= 3` | `false` | | === | Equal (strict comparison) | `5 === 5` | `true` | | !== | Not equal | `5 !== "5"` | `true` |

Booleans: True or False Logic Explain: Booleans represent true or false values, often the result of comparisons. Demo: Use comparison operators to generate boolean values:

```
console.log(10 > 5); // Output: true
console.log(5 === "5"); // Output: false
console.log(7 !== 3); // Output: true
```

Practical Activity: Ask students to write comparisons using <, >=, and `!==` to determine boolean results.

Truthy and Falsy Values

Explain: In JavaScript, truthy and falsy values are evaluated in conditions:

Falsy: `false`, `0`, `""`, `null`, `undefined`, NaN Truthy: Anything not falsy (e.g., `"hello"`, `42`, `{}`, `[]`).

Demo: Log the boolean values of falsy and truthy values:

```
console.log(Boolean(0)); // Output: false
console.log(Boolean("")); // Output: false
console.log(Boolean("Hello")); // Output: true
```

Activity: Have students test other falsy values like null, undefined, and NaN.

III. If-Statements: Adding Logic Explain: If-statements allow conditional execution of code.

Demo: Write an example of an age verification:

```
if (condition) {
    // Code to execute if the condition is true
} else {
    // Code to execute if the condition is false
}
```

```
let age = 17;
if (age >= 18) {
    console.log("You are eligible to vote.");
} else {
    console.log("You are not old enough to vote.");
}
```

Combining Conditions with Logical Operators Explain: Use &&, ||, and ! to combine or negate conditions.

Demo: Combine conditions to check for a workday:

```
let isWeekend = true;
let isHoliday = false;

if (isWeekend || isHoliday) {
    console.log("You can relax today!"); // Output: "You can
} else {
    console.log("It's a workday.");
}
```

Activity: Ask students to write combined conditions using their own variables.

Nested If-Statements Explain: Nesting allows more specific checks.

Demo: Use a nested condition to verify a user's access:

```
const username = "admin";
const password = "password123";

if (username === "admin") {
    if (password === "password123") {
        console.log("Access granted!"); // Output: "Access g
    } else {
        console.log("Incorrect password.");
    }
} else {
    console.log("Unknown user.");
}
```

Activity: Have students create a nested if-statement for a login system.

> IV. Functions: Reusable Logic Explain: Functions group reusable blocks of code to improve readability and reduce repetition.

Explain: Functions group reusable blocks of code, making programs efficient and readable.

```
functionName(parameters) {
    // Code blockto execute
    return value; //Optional
}
```

Demo: Greet a user with a personalized message:

```
function greet(name) {
    console.log(`Hello, ${name}!`);
```

```
}
greet("Alice"); // Output: "Hello, Alice!"
```

Combining Functions with Conditions Explain: Functions can include if-statements to make decisions.

Demo: Check voting eligibility:

```
function checkEligibility(age) {
    if (age >= 18) {
        return "You are eligible to vote.";
    } else {
        return "You are not old enough to vote.";
    }
}

console.log(checkEligibility(21)); // Output: "You are eligi
```

also for the above a alternate simpler way is to use Ternary operator

```
function checkEligibility(age) {
    return age >= 18 ? "You are eligible to vote." : "You ar
}
console.log(checkEligibility(21)); // Output: "You are eligi
```

Default Parameters

Explain: Functions can have default values for parameters. Demo: Greet a user with a default name:

```
function greet(name = "Guest") {
    console.log(`Hello, ${name}!`);
}
greet(); // "Hello, Guest!"
greet("Bob"); // "Hello, Bob!"
```

Practical Example: Tax and Discount Calculator

Explain: Use functions to calculate a final price after applying a discount and tax. Demo: Calculate the final price of an item with a discount and tax:

```
function calculateFinalPrice(price, discountRate = 0.1, taxR
    const discount = price * discountRate;
    const discountedPrice = price - discount;
    const tax = discountedPrice * taxRate;
    const finalPrice = discountedPrice + tax;

    console.log(`Original Price: $${price}`);
    console.log(`Discount: $${discount}`);
    console.log(`Tax: $${tax}`);
    console.log(`Final Price: $${finalPrice.toFixed(2)}`);
}
calculateFinalPrice(200, 0.15, 0.1);
```

Exercises: Booleans, If-Statements, and Functions

1. Truthy and Falsy Check

Write a script to check if a value is truthy or falsy. Use the following values:

- `0`
- "" (empty string)
- `"Hello"`
- `42`
- `null`

Log `"Truthy"` or `"Falsy"` for each value.

2. Age Eligibility

Write a script to check if a user is eligible to vote based on their age.
If the user is 18 or older, log `"You are eligible to vote."`
Otherwise, log `"You are not old enough to vote."`

3. Calculate Final Price

Write a script to calculate the final price of an item after a 20% discount and an 8% tax.
Use the following:

- Initial price: $150
- Discount rate: 20%
- Tax rate: 8%

Log:

- `"Original Price: $<price>"`
- `"Discount: $<discount>"`
- `"Tax: $<tax>"`
- `"Final Price: $<finalPrice>"`

4. Nested If-Statements for Login

Write a script to simulate a login system:

I. Check if the `username` is `"admin"`.

II. If it is, check if the `password` is `"password123"`.

- If both are correct, log `"Access granted!"`.
- Otherwise, log `"Incorrect password."`

III. If the `username` is not `"admin"`, log `"Unknown user."`

5. Driving Eligibility

Write a script to check if a person is eligible to drive:

I. If the person is 18 or older:
- Check if they have a driver's license (`hasLicense` is `true` or `false`).
- If they do, log `"You can drive."`
- If not, log `"You need a license to drive."`

II. If the person is under 18, log `"You are not old enough to drive."`

6. Logical Operators Practice

Write a script to check multiple conditions using logical operators:

I. Declare `isWeekend` and `isHoliday` as boolean variables.

II. If it's either a weekend or a holiday, log `"You can relax today!"`.

III. If it's neither, log `"It's a workday."`.

7. Function: Greeting

Write a function `greet(name)` that takes a `name` as a parameter and logs a personalized greeting:
`"Hello, <name>! Welcome to JavaScript class."`

---

8. Function: Square a Number

Write a function `square(number)` that returns the square of the given number.
Use the function to calculate and log the square of `5`, `7`, and `10`.

---

9. Function with Default Parameters

Write a function `calculateDiscount(price, discountRate = 0.1)` that:

I. Calculates the discount on a given price.

II. Logs `"Discount: $<discount>"` and `"Final Price: $<finalPrice>"`.

Call the function with:

I. `price = 100` and no discount rate (use default).

II. `price = 200` and `discountRate = 0.2`.

---

10. Tax and Discount Calculator

Write a function `calculateFinalPrice(price, discountRate = 0.15, taxRate = 0.08)` that:

I. Calculates the discounted price.

II. Adds tax to the discounted price.

III. Logs:

- `"Original Price: $<price>"`
- `"Discount: $<discount>"`
- `"Tax: $<tax>"`
- `"Final Price: $<finalPrice>"`

Call the function with:

- `price = 100` (use default rates).
- `price = 250`, `discountRate = 0.2`, `taxRate = 0.1`.

Mark Lesson As Complete

Cape - Mark Complete