Practical 6 (Writeup)

Title: Create student registration form using .

Aim: Create student registration form using .

Theory:

- Operators

1. Arithmetic Operators:

- Addition (+):

```
let sum = 5 + 3;
console.log(sum); // Output: 8
```

- Subtraction (-):

```
let difference = 10 - 5;
console.log(difference); // Output: 5
```

- Multiplication (*):

```
let product = 3 * 4;
console.log(product); // Output: 12
```

- Division (/):

```
let quotient = 20 / 5;
console.log(quotient); // Output: 4
```

- Modulus (%):

```
let remainder = 10 % 3;
console.log(remainder); // Output: 1
```

- Exponentiation (**):

```
let result = 2 ** 3;
console.log(result); // Output: 8
```

2. Assignment Operators:

- Assignment (=):

```
let x = 10;
```

- Addition Assignment (+=):

```
let a = 5;
```

```
a += 3; // Equivalent to a = a + 3;
console.log(a); // Output: 8
```

- Subtraction Assignment (-=):

```
let b = 10;
b -= 5; // Equivalent to b = b - 5;
console.log(b); // Output: 5
```

- Multiplication Assignment (*=):

```
let c = 3;
c *= 4; // Equivalent to c = c * 4;
console.log(c); // Output: 12
```

- Division Assignment (/=):

```
let d = 20;
d /= 5; // Equivalent to d = d / 5;
console.log(d); // Output: 4
```

3. Comparison Operators:

- Equal to (==):

```
let x = 5;
let y = 5;
console.log(x == y); // Output: true
```

- Not equal to (!=):

```
let a = 5;
let b = 3;
console.log(a != b); // Output: true
```

- Strict equal to (===):

```
let p = 5;
let q = '5';
console.log(p === q); // Output: false
```

- Strict not equal to (!==):

```
let m = 5;
```

```
let n = '5';
console.log(m !== n); // Output: true
```

- Greater than (>):

```
let num1 = 10;
let num2 = 5;
console.log(num1 > num2); // Output: true
```

- Less than (<):

```
let num3 = 8;
let num4 = 12;
console.log(num3 < num4); // Output: true
```

- Greater than or equal to (>=):

```
let val1 = 7;
let val2 = 7;
console.log(val1 >= val2); // Output: true
```

- Less than or equal to (<=):

```
let val3 = 4;
let val4 = 6;
console.log(val3 <= val4); // Output: true
```

4. Logical Operators:

- Logical AND (&&):

```
let x = 5;
let y = 10;
console.log((x < 10) && (y > 5)); // Output: true
```

- Logical OR (||):

```
let a = 3;
let b = 7;
console.log((a > 5) || (b > 5)); // Output: true
```

- Logical NOT (!):

```
let isValid = false;
```

```
console.log(!isValid); // Output: true
```

5. Unary Operators:

- Increment (++):

```
let count = 5;
count++;
console.log(count); // Output: 6
```

- Decrement (--):

```
let num = 10;
num--;
console.log(num); // Output: 9
```

- Unary Plus (+):

```
let num1 = 5;
let num2 = +num1;
console.log(num2); // Output: 5
```

- Unary Negation (-):

```
let num3 = 10;
let num4 = -num3;
console.log(num4); // Output: -10
```

- Logical NOT (!):

```
let isTrue = false;
console.log(!isTrue); // Output: true
```

- typeof:

```
let name = 'John';
console.log(typeof name); // Output: string
```

6. Conditional (Ternary) Operator:

- Ternary Operator (?:)

```
let age = 18;
let isAdult = (age >= 18) ? 'Adult' : 'Not Adult';
console.log(isAdult); // Output: Adult
```

- Alert Box

An alert box is a built-in function that displays a small dialog box with a message to the user. It is often used to provide important information or prompt the user for confirmation. When the alert box appears, it interrupts the execution of the script until the user dismisses it by clicking the "OK" button.

The syntax to display an alert box is: alert("Your message goes here");

- Confirm Box

A confirm box is another built-in function that displays a dialog box with a message and provides the user with a choice to confirm or cancel an action. It is commonly used to ask for user confirmation before proceeding with a potentially irreversible action.

The syntax to display a confirm box is: confirm("Your message goes here");

- Prompt Box

A prompt box is a built-in function that displays a dialog box with a message and allows the user to enter input. It is commonly used to prompt the user for data or information.

The syntax to display a prompt box is: prompt("Message", "Default Value");

- Events in JS

1. Mouse Events:

- click: Occurs when the user clicks the mouse button.

- dblclick: Occurs when the user double-clicks the mouse button.

- mouseover: Occurs when the mouse pointer enters an element.

- mouseout: Occurs when the mouse pointer leaves an element.

- mousemove: Occurs when the mouse pointer moves within an element.

2. Keyboard Events:

- keydown: Occurs when a key on the keyboard is pressed down.

- keyup: Occurs when a key on the keyboard is released.

- keypress: Occurs when a key on the keyboard is pressed and released.

3. Form Events:

- submit: Occurs when a form is submitted.

- focus: Occurs when an element receives focus.

- blur: Occurs when an element loses focus.

- change: Occurs when the value of an input element changes.

4. Window Events:

- load: Occurs when the web page finishes loading.

- resize: Occurs when the browser window is resized.

- scroll: Occurs when the user scrolls within the web page.

5. Touch Events:

- touchstart: Occurs when a finger touches the screen.

- touchend: Occurs when a finger is lifted from the screen.

- touchmove: Occurs when a finger moves on the screen.

6. Media Events:

- play: Occurs when media (audio/video) starts playing.

- pause: Occurs when media (audio/video) is paused.

- ended: Occurs when media (audio/video) playback has ended.

7. Event Handling:

- addEventListener: Method used to attach an event handler to an element.

- removeEventListener: Method used to remove an event handler from an element.

- Form Validation

Form Validation:

- Form validation is often performed to ensure that user-submitted form data is valid before it is sent to the server for processing.

- can be used to validate form fields such as text inputs, checkboxes, radio buttons, dropdowns, etc.

- Common validation checks include checking for required fields, validating email addresses, verifying passwords, enforcing data format (e.g., numbers, dates), and checking input length or range.

- Validation can be performed on form submission or interactively as the user fills in the form using event listeners or inline event attributes.

Conclusion: This write-up covered the concepts of operators, alert boxes, confirm boxes, prompt boxes, events, and form validation in JavaScript, providing examples and explanations for each. By understanding and applying these concepts, developers can create more interactive and robust web applications.

****code and output not available in this file for practical***