**HTML**

**CANVAS**

<!DOCTYPE html>

<html lang="en">

<head>

    <title>Canvas API</title>

</head>

<body>

    <canvas id="myCanvas" width="200" height="200" style="border:1px solid #000000;">

    </canvas>

</body>

</html>

LINE

    var c = document.getElementById("myCanvas");

    var ctx = c.getContext("2d");

    ctx.moveTo(0, 0);

    ctx.lineTo(200, 100);

    ctx.stroke();

CIRCLE

    var c = document.getElementById("myCanvas");

    var ctx = c.getContext("2d");

    ctx.beginPath();

    ctx.arc(100, 100, 40, 0, 2 \* Math.PI);

    ctx.stroke();

RECTANGLE

    var c = document.getElementById("myCanvas");

    var ctx = c.getContext("2d");

    ctx.beginPath();

    ctx.rect(20, 20, 150, 100);

    ctx.stroke();

ELLIPSE

    ctx.ellipse(x - 25, y + 150, 20, 40, -100, 0, 2 \* Math.PI);

TEXT

    var c = document.getElementById("myCanvas");

    var ctx = c.getContext("2d");

    ctx.font = "30px Arial";

    ctx.fillText("Hello World", 10, 50);

IMAGE

    var c = document.getElementById("myCanvas");

    var ctx = c.getContext("2d");

    var img = new Image();

    img.src = '/images/backdrop.jpg';

    ctx.drawImage(img, 0, 0);

OTHER PROPERTIES

    var c = document.getElementById("myCanvas");

    var ctx = c.getContext("2d");

    ctx.beginPath();

    ctx.arc(95, 50, 40, 0, 360);

    ctx.lineWidth = 2;

ctx.strokeStyle = 'black';

    ctx.stroke();

    ctx.fillStyle = "red";

    ctx.fill();

    ctx.closePath();

*// To clear the canvas*

    ctx.clearRect(0, 0, c.width, c.height);

**SVG**

<svg width="100" height="100">

    <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />

</svg>

<svg width="200" height="100">

    <rect width="200" height="100" stroke="yellow" stroke-width="4" fill="red" />

</svg>

**GEOLOCATION**

<button onclick="getLocation()">Get Location</button>

<p id="demo"></p>

<script>

    var x = document.getElementById("demo");

    function getLocation() {

        if (navigator.geolocation) {

            navigator.geolocation.getCurrentPosition(showPosition);

        } else {

            x.innerHTML = "Geolocation is not supported by this browser.";

        }

    }

    function showPosition(*position*) {

        x.innerHTML = "Latitude: " + position.coords.latitude + "<br>Longitude: " + position.coords.longitude;

    }

</script>

HTML 5 Tags

<!DOCTYPE html>

<html lang="en">

<head>

    <title>HTML 5 & CSS 3</title>

    <link rel="stylesheet" href="style.css" />

    <script src="script.js"></script>

</head>

<body>

    <nav></nav>

    <header></header>

    <section></section>

    <footer></footer>

    <article></article>

    <aside></aside>

    <figure></figure>

    <progress value="60" max="100">60%</progress>

    <meter min="0" max="10" value="8">8 of 10</meter>

    <audio controls autoplay>

        <source src="/html5/audio.mp3" type="audio/mp3" />

    </audio>

    <video width="300" height="200" controls autoplay>

        <source src="/html5/foo.mp4" type="video/mp4" />

    </video>

    <form action="insert.php">

        <fieldset>

            <legend>Personal Details:</legend>

            <label for="fname">First name:</label><br>

            <input type="text" id="fname" name="fname" value="John"><br>

            <label for="lname">Last name:</label><br>

            <input type="text" id="lname" name="lname" value="Doe"><br><br>

            <label for="email">Email:</label><br>

            <input type="email" id="email" name="email" placeholder="Enter Email">

            <input type="submit" value="Submit">

            <button type="submit">Submit</button>

        </fieldset>

    </form>

</body>

</html>

A screenshot of a video player

Description automatically generated with medium confidence

**CSS**

audio {

*border-radius*: 10px;

*border-width*: 5px;

*border-style*: solid;

*border-color*: black;

}

form {

*width*: 50%;

*box-shadow*: 0px 5px 15px rgba(0, 0, 0, 0.3);

*transform*: scale(1.1);

*transform*: translate(50%, 50%);

*transform*: rotate(45deg);

*transform*: skew(10deg, 10deg);

*/\* All those has 3d too \*/*

*transform*: scale3d(), translate3d(), rotate3d();

*/\* All those has X, Y, Z separately \*/*

*transform*: translateX(), translateY(), translateZ();

}

.transition {

*transition-property*: width 2s, height 2s, transform 2s;

*transition-duration*: 2s;

*transition-timing-function*: linear, ease-in, ease-out, ease-in-out;

*transition-delay*: 1s;

*transition*: all 2s linear 1s;

}

.animation {

*animation-name*: example;

*animation-duration*: 5s;

*animation-timing-function*: linear;

*animation-delay*: 2s;

*animation-iteration-count*: infinite;

*animation-direction*: alternate;

*animation*: example 5s linear 2s infinite alternate;

}

@keyframes *example* {

    0% {

*background-color*: red;

    }

    25% {

*background-color*: yellow;

    }

    50% {

*background-color*: blue;

    }

    100% {

*background-color*: green;

    }

}

*/\* Media Queries \*/*

@media (max-width: 600px) {

    body {

*background-color*: red;

    }

}

*/\* Universal Selector \*/*

    \* {

*font-family*: Arial, Helvetica, sans-serif;

    }

*/\* Element Selector \*/*

    h1 {

*color*: blue;

    }

*/\* Class Selector \*/*

    .highlight {

*background-color*: yellow;

    }

*/\* ID Selector \*/*

    #header {

*font-size*: 24px;

    }

*/\* Attribute Selector \*/*

    input[type='text'] {

*border*: 1px solid gray;

    }

*/\* Descendant Selector \*/*

    ul li {

*color*: green;

    }

*/\* Child Selector \*/*

    ul>li {

*font-weight*: bold;

    }

*/\* Adjacent Sibling Selector \*/*

    h2+p {

*margin-top*: 20px;

    }

*/\* General Sibling Selector \*/*

    h2~p {

*color*: red;

    }

*/\* Pseudo-class Selector \*/*

    a:hover {

*color*: purple;

    }

*/\* Pseudo-element Selector \*/*

    p::first-line {

*font-weight*: bold;

    }

*/\* :not() Selector \*/*

    input:not([type='submit']) {

*background-color*: lightgray;

    }

**AJAX**

<!DOCTYPE html>

<html>

<head>

    <title>Song Information</title>

    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>

</head>

<body ng-app="myApp" ng-controller="songController">

    <div class="container">

        <div ng-repeat="song in songs">

            <h2>{{song.songName}}</h2>

            <p>{{song.lyrics}}</p>

        </div>

    </div>

    <script>

        var app = angular.module('myApp', []);

        app.controller('songController', function (*$scope*) {

*// Using XMLHttpRequest*

            var xhr = new XMLHttpRequest();

            var filePath = 'musics.json';

            xhr.open('GET', filePath, true);

            xhr.responseType = 'json';

            xhr.onload = function () {

                if (xhr.status === 200) {

                    var musicData = xhr.response;

                    $scope.songs = musicData;

                    $scope.$apply();  *// Trigger the digest cycle to update the view*

                }

            };

            xhr.send();

        })

*OR*

*// Using angular JS*

       app.controller('SongController', function (*$scope*, *$http*) {

        var filePath = 'musics.txt';

        $http.get(filePath)

            .then(*response* => {

                var musicData = response.data;

                var lines = musicData.split('\n');

                $scope.songName = lines[0];

                $scope.lyrics = lines.slice(1).join('\n');

            })

            .catch(*error* => {

                console.error('Error retrieving song information:', error);

            });

    });

    </script>

</body>

</html>

**JS**

Scroll to a div

    function scrollFunction() {

        document.getElementById("try").scrollIntoView({

            behavior: 'smooth',

        });

    }

Animate div on scroll

    window.addEventListener('scroll', function () {

        var movingElement = document.querySelector('.moving-element');

        var value = window.scrollY;

*// Update the left position based on scrollY value*

        movingElement.style.left = value + 'px';

    });

FORM VALIDATION (REGULAR EXPRESSION)

<!DOCTYPE html>

<html lang="en">

<head>

    <title>Form Validation</title>

</head>

<body>

    <form onsubmit="return validate()">

        <fieldset>

            <legend>Form Validation</legend>

            <p><input type="text" id="phone-number" placeholder="phone number" /></p>

            <p><input type="text" id="pin-code" placeholder="pin code" /></p>

            <p><input type="text" id="email-id" placeholder="email id" /></p>

            <p><input type="submit" /></p>

        </fieldset>

    </form>

</body>

<script>

    function validate() {

        var phoneNumber = document.getElementById('phone-number').value;

        var postalCode = document.getElementById('pin-code').value;

        var emailId = document.getElementById('email-id').value;

        var phoneRGEX = /^\d{10}$/;

        var postalRGEX = /^\d{6}$/;

        var emailRGEX = /^\w+([\.-]?\w+)\*@\w+([\.-]?\w+)\*(\.\w{2,3})+$/;

        var phoneResult = phoneRGEX.test(phoneNumber);

        var postalResult = postalRGEX.test(postalCode);

        var emailResult = emailRGEX.test(emailId);

        if (phoneResult == false) {

            alert('Please enter a valid phone number');

            return false;

        }

        if (postalResult == false) {

            alert('Please enter a valid pin code');

            return false;

        }

        if (emailResult == false) {

            alert('Please enter a valid email id');

            return false;

        }

        alert("Form Submitted Successfully");

        return true;

    }

</script>

</html>

DOM MANIPLUATION

const btn = document.getElementById('btn');

        btn.addEventListener('click', function master() {

            var master = document.getElementsByClassName("master2");

            master[2].innerHTML = 'To earn money';

            master[3].innerHTML = 'I need a job';

        });

SIMPLE MESSAGE SERVICE (CREATE & APPEND NEW DIV)

    var chat = document.getElementById("chat");

    var messageInput = document.getElementById("message");

    var sendButton = document.getElementById("send-button");

    sendButton.addEventListener("click", function () {

        var message = document.createElement("div");

        message.innerText = messageInput.value;

        chat.appendChild(message);

        messageInput.value = "";

        chat.scrollTop = chat.scrollHeight;

    });

FORM VALIDATION

<!DOCTYPE html>

<html>

<head>

    <style>

        .error-message {

*color*: red;

*font-size*: 12px;

*margin-top*: 5px;

*display*: none;

        }

        input:invalid {

*border-color*: red;

        }

    </style>

</head>

<body>

    <form>

        <label for="username">Username:</label>

        <input type="text" id="username" name="username" pattern="[a-z\_]+" required>

        <p id="username-error" class="error-message"></p>

        <label for="password">Password:</label>

        <input type="password" id="password" name="password" required>

        <p id="password-error" class="error-message"></p>

        <button type="submit">Submit</button>

    </form>

    <script>

        var usernameInput = document.getElementById('username');

        var usernameError = document.getElementById('username-error');

        var passwordInput = document.getElementById('password');

        var passwordError = document.getElementById('password-error');

        usernameInput.addEventListener('input', function () {

            var username = usernameInput.value;

            if (validateUsername(username)) {

                usernameError.style.display = 'none';

                usernameInput.classList.remove('invalid');

            } else {

                usernameError.style.display = 'block';

                usernameError.textContent = 'Username must have lowercase letters and underscores only.';

                usernameInput.classList.add('invalid');

            }

        });

        passwordInput.addEventListener('input', function () {

            var password = passwordInput.value;

            if (validatePassword(password)) {

                passwordError.style.display = 'none';

                passwordInput.classList.remove('invalid');

            } else {

                passwordError.style.display = 'block';

                passwordError.textContent = 'Password must have at least one uppercase letter, one lowercase letter, one digit, and one special character.';

                passwordInput.classList.add('invalid');

            }

        });

        function validateUsername(*username*) {

            return /^[a-z\_]+$/.test(username);

        }

        function validatePassword(*password*) {

*// Check for at least one uppercase letter*

            if (!/[A-Z]/.test(password)) {

                return false;

            }

*// Check for at least one lowercase letter*

            if (!/[a-z]/.test(password)) {

                return false;

            }

*// Check for at least one digit*

            if (!/\d/.test(password)) {

                return false;

            }

*// Check for at least one special character*

            if (!/[!@#$%^&\*]/.test(password)) {

                return false;

            }

*// All criteria are met*

            return true;

        }

    </script>

</body>

</html>