```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Document Object Model - DOM</title>
</head>

<body>
    <!-- Primer ejemplo -->
    <div id="caja" style="width:200px; height:200px;
background:lightgray;"></div>
    <div id="areaTouch" style="width:200px; height:200px;
background:skyblue;"></div>

    <!-- Segundo ejemplo -->
    <div id="container" style="position:relative; width:100%;
height:400px; background:whitesmoke; margin-top:20px;">
        <div id="circulo" style="width:80px; height:80px;
background:red; border-radius:50%; position:absolute; top:100px;
left:100px;"></div>
    </div>
    <!-- Tercer ejemplo -->
    <div style="margin-top:20px;">
        <button id="botonAnimado" style="padding:10px 20px;
font-size:18px; transition: all 0.3s;">¡Tócame o pasa el
mouse!</button>
    </div>

<!-- Cuarto ejemplo -->
<div id="contenedorTriangulo" style="position:relative; width:100%;
height:300px; margin-top:20px; background:whitesmoke;">
    <div id="triangulo" style="
        width: 0;
        height: 0;
        border-left: 50px solid transparent;
        border-right: 50px solid transparent;
        border-bottom: 100px solid crimson;
        position: absolute;
        top: 100px;
        left: calc(50% - 50px);
```

```
        transition: transform 0.2s ease;">
    </div>
</div>
<script src="dom06-mouse-touch.js"></script>


</body>

</html>

const caja = document.getElementById('caja');
caja.addEventListener('mouseover', () => caja.style.background =
'yellow');
caja.addEventListener('mouseout', () => caja.style.background =
'lightgray');
caja.addEventListener('click', () => alert('¡Click!'));

const area = document.getElementById('areaTouch');
area.addEventListener('touchstart', () => {
    area.style.background = 'deepskyblue';
    console.log('Touch iniciado');
});
area.addEventListener('touchend', () => {
    area.style.background = 'skyblue';
    console.log('Touch finalizado');
});
const circulo = document.getElementById('circulo');
let colores = ['red', 'green', 'blue', 'orange', 'purple', 'pink'];
let colorActual = 0;

let arrastrando = false;

function cambiarColor() {
    colorActual = (colorActual + 1) % colores.length;
    circulo.style.background = colores[colorActual];
}

// Mover el círculo al punto (centrado)
function moverCirculo(x, y) {
    const container = document.getElementById('container');
    const containerRect = container.getBoundingClientRect();
    const circuloRect = circulo.getBoundingClientRect();
    const mitadAncho = circuloRect.width / 2;
```

```
    const mitadAlto = circuloRect.height / 2;

    const offsetTop = containerRect.top + window.scrollY;
    const offsetLeft = containerRect.left + window.scrollX;

    circulo.style.left = (x - offsetLeft - mitadAncho) + 'px';
    circulo.style.top = (y - offsetTop - mitadAlto) + 'px';
}


// Touch Events
circulo.addEventListener('touchstart', (e) => {
    e.preventDefault();
    cambiarColor();
    arrastrando = true;
});

document.addEventListener('touchmove', (e) => {
    if (!arrastrando) return;
    const touch = e.touches[0];
    moverCirculo(touch.pageX, touch.pageY);
});

document.addEventListener('touchend', () => {
    arrastrando = false;
});

// Mouse Events
circulo.addEventListener('mousedown', (e) => {
    e.preventDefault();
    cambiarColor();
    arrastrando = true;
});

document.addEventListener('mousemove', (e) => {
    if (!arrastrando) return;
    moverCirculo(e.clientX, e.clientY);
});

document.addEventListener('mouseup', () => {
    arrastrando = false;
});
```

```
// Cuarto ejemplo: Botón que crece con eventos

const boton = document.getElementById('botonAnimado');

// Aumentar tamaño
function agrandar() {
    boton.style.transform = 'scale(1.5)';
}

// Volver a tamaño normal
function normalizar() {
    boton.style.transform = 'scale(1)';
}

// Mouse events
boton.addEventListener('mouseover', agrandar);
boton.addEventListener('mouseout', normalizar);

// Touch events
boton.addEventListener('touchstart', (e) => {
    e.preventDefault();
    agrandar();
});
boton.addEventListener('touchend', normalizar);


// Cuarto ejemplo: Espiral que gira al arrastrar y se detiene a los 6
segundos
const espiral = document.getElementById('espiral');
let rotacion = 0;
let arrastrandoEspiral = false;
let intervaloGiro = null;

// Función para girar
function girar(xInicial, xActual) {
    const diferencia = xActual - xInicial;
    rotacion += diferencia * 0.5; // Controla velocidad de giro
    espiral.style.transform = `rotate(${rotacion}deg)`;
}

// Evento Touch
```

```
let xInicioTouch = 0;

espiral.addEventListener('touchstart', (e) => {
    e.preventDefault();
    const touch = e.touches[0];
    xInicioTouch = touch.pageX;
    arrastrandoEspiral = true;

    if (intervaloGiro) clearTimeout(intervaloGiro);
    intervaloGiro = setTimeout(() => arrastrandoEspiral = false, 6000);
// Detener a los 6 segundos
});

document.addEventListener('touchmove', (e) => {
    if (!arrastrandoEspiral) return;
    const touch = e.touches[0];
    girar(xInicioTouch, touch.pageX);
    xInicioTouch = touch.pageX;
});

document.addEventListener('touchend', () => {
    arrastrandoEspiral = false;
});



// Evento Mouse
const triangulo = document.getElementById('triangulo');
let isDragging = false;
let startX = 0;
let scaleValue = 1;

// Función para actualizar tamaño
function updateScale(newX) {
    const diffX = newX - startX;
    scaleValue += diffX * 0.005; // Ajusta sensibilidad aquí
    scaleValue = Math.max(0.5, Math.min(2, scaleValue)); // Limitar
entre 0.5x y 2x
    triangulo.style.transform = `scale(${scaleValue})`;
    startX = newX; // actualizar la posición para el siguiente
movimiento
}
```

```
// MOUSE Events
triangulo.addEventListener('mousedown', (e) => {
    e.preventDefault();
    isDragging = true;
    startX = e.clientX;
});

document.addEventListener('mousemove', (e) => {
    if (!isDragging) return;
    updateScale(e.clientX);
});

document.addEventListener('mouseup', () => {
    isDragging = false;
});

// TOUCH Events
triangulo.addEventListener('touchstart', (e) => {
    e.preventDefault();
    isDragging = true;
    startX = e.touches[0].clientX;
});

document.addEventListener('touchmove', (e) => {
    if (!isDragging) return;
    updateScale(e.touches[0].clientX);
});

document.addEventListener('touchend', () => {
    isDragging = false;
});
```