

# Capítulo 1: Introducción a JavaScript Creativo

---

## Introducción a la programación creativa

### ¿Qué es la programación creativa?

La programación creativa es una disciplina que combina la programación informática con la creatividad artística. Su objetivo es utilizar el código como una herramienta para crear obras de arte digitales, animaciones interactivas, juegos y experiencias multimedia. La programación creativa se basa en la premisa de que la programación no es solo una herramienta para resolver problemas, sino también una forma de expresión artística.

La programación creativa se ha convertido en una disciplina muy popular en los últimos años, gracias a la creciente accesibilidad de las herramientas y tecnologías necesarias para desarrollar proyectos creativos. Con la programación creativa, puedes dar vida a tus ideas y crear proyectos interactivos que cautiven a tu audiencia.

En nuestro caso, tomaremos la programación creativa para visualizar y hacer tangible el aprendizaje de la programación. A través de ejemplos prácticos y proyectos creativos, aprenderás a programar con JavaScript de una manera divertida y estimulante.

### ¿Por qué aprender programación creativa?

La programación creativa tiene muchos beneficios, tanto para los programadores experimentados como para los principiantes. Algunas de las razones por las que deberías aprender programación creativa son:

- **Desarrollo de habilidades técnicas:** La programación creativa te permite desarrollar habilidades técnicas en programación, diseño y animación. Aprenderás a utilizar herramientas y tecnologías modernas para crear proyectos creativos.
- **Expresión artística:** La programación creativa te brinda una forma de expresión artística única. Puedes utilizar el código como un medio para crear obras de arte digitales, animaciones interactivas y experiencias multimedia.

- **Creatividad y pensamiento crítico:** La programación creativa fomenta la creatividad y el pensamiento crítico. Te desafía a pensar de forma creativa para resolver problemas y crear proyectos innovadores.
- **Oportunidades laborales:** La programación creativa es una habilidad muy demandada en la industria del diseño y la tecnología. Aprender programación creativa puede abrirte nuevas oportunidades laborales y profesionales o hacer tu perfil más atractivo para los empleadores.

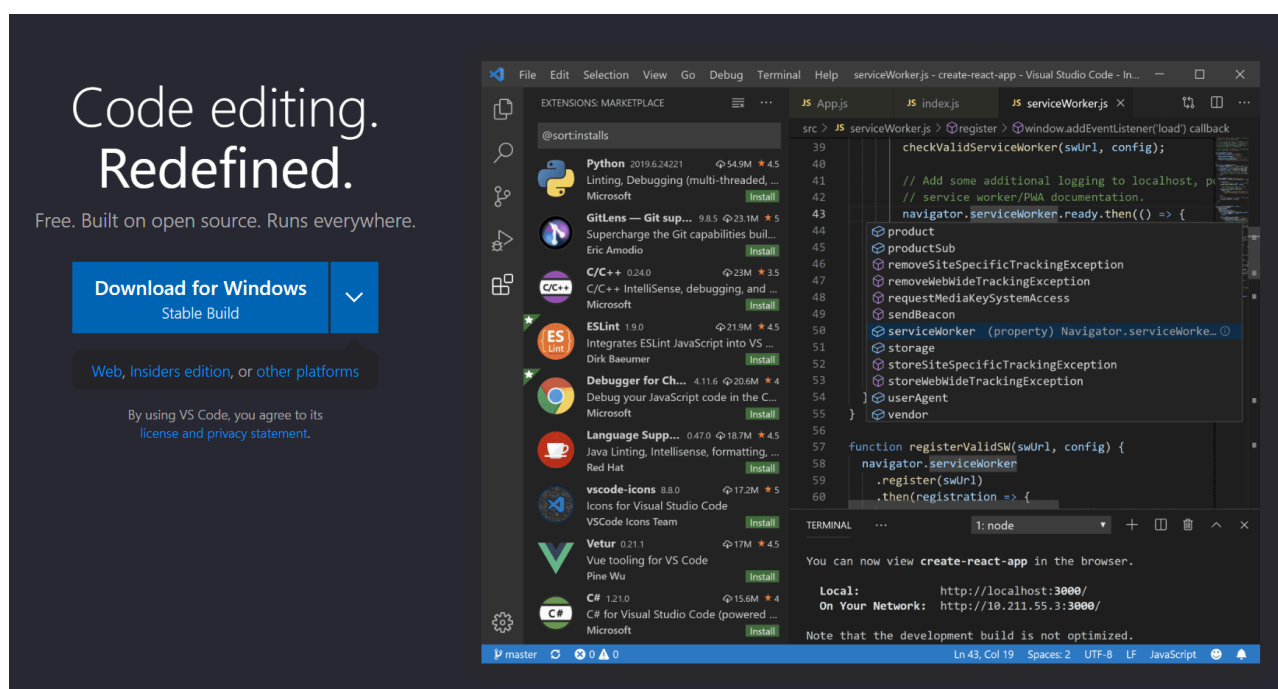
## Configuración del entorno de desarrollo

Para comenzar a programar con JavaScript, necesitarás configurar un entorno de desarrollo en tu computadora. A continuación, te explicaremos cómo configurar un entorno de desarrollo básico para programar con JavaScript.

### Editor de código

El primer paso para configurar tu entorno de desarrollo es elegir un editor de código. Un editor de código es una herramienta que te permite escribir y editar código fuente. Voy a recomendar el uso de Visual Studio Code, un editor de código gratuito y de código abierto desarrollado por Microsoft. Visual Studio Code es uno de los editores de código más populares y ampliamente utilizados en la comunidad de programadores.

Puedes descargar Visual Studio Code desde su [sitio web oficial](#). Una vez que hayas descargado e instalado Visual Studio Code, estarás listo para comenzar a programar con JavaScript.



## Extensiones de Visual Studio Code (opcional)

Visual Studio Code cuenta con una amplia variedad de extensiones que te permiten personalizar y ampliar sus funcionalidades. A continuación, te recomendaré algunas extensiones útiles para programar con JavaScript:

- **Prettier:** Prettier es una extensión que te ayuda a formatear tu código de forma automática y consistente. Puedes instalar Prettier desde la tienda de extensiones de Visual Studio Code. Una vez instalada, Prettier formateará tu código cada vez que guardes un archivo.
- **Auto Close Tag:** Auto Close Tag es una extensión que te permite cerrar automáticamente las etiquetas HTML al escribir el código. Esta extensión es muy útil para escribir código HTML de forma más rápida y eficiente.
- **Live Server:** Live Server es una extensión que te permite visualizar tu página web en tiempo real mientras escribes el código. Con Live Server, puedes ver los cambios que haces en tu página web de forma inmediata, sin necesidad de recargar el navegador.
- **Auto Rename Tag:** Auto Rename Tag es una extensión que te permite renombrar automáticamente las etiquetas HTML al cambiar el nombre de una etiqueta de apertura. Esta extensión es muy útil para mantener la coherencia en tu código HTML.
- **Material Icon Theme:** Material Icon Theme es una extensión que te permite personalizar los iconos de los archivos y carpetas en Visual Studio Code. Esta extensión es útil para identificar rápidamente los tipos de archivos en tu proyecto.

## Navegador web

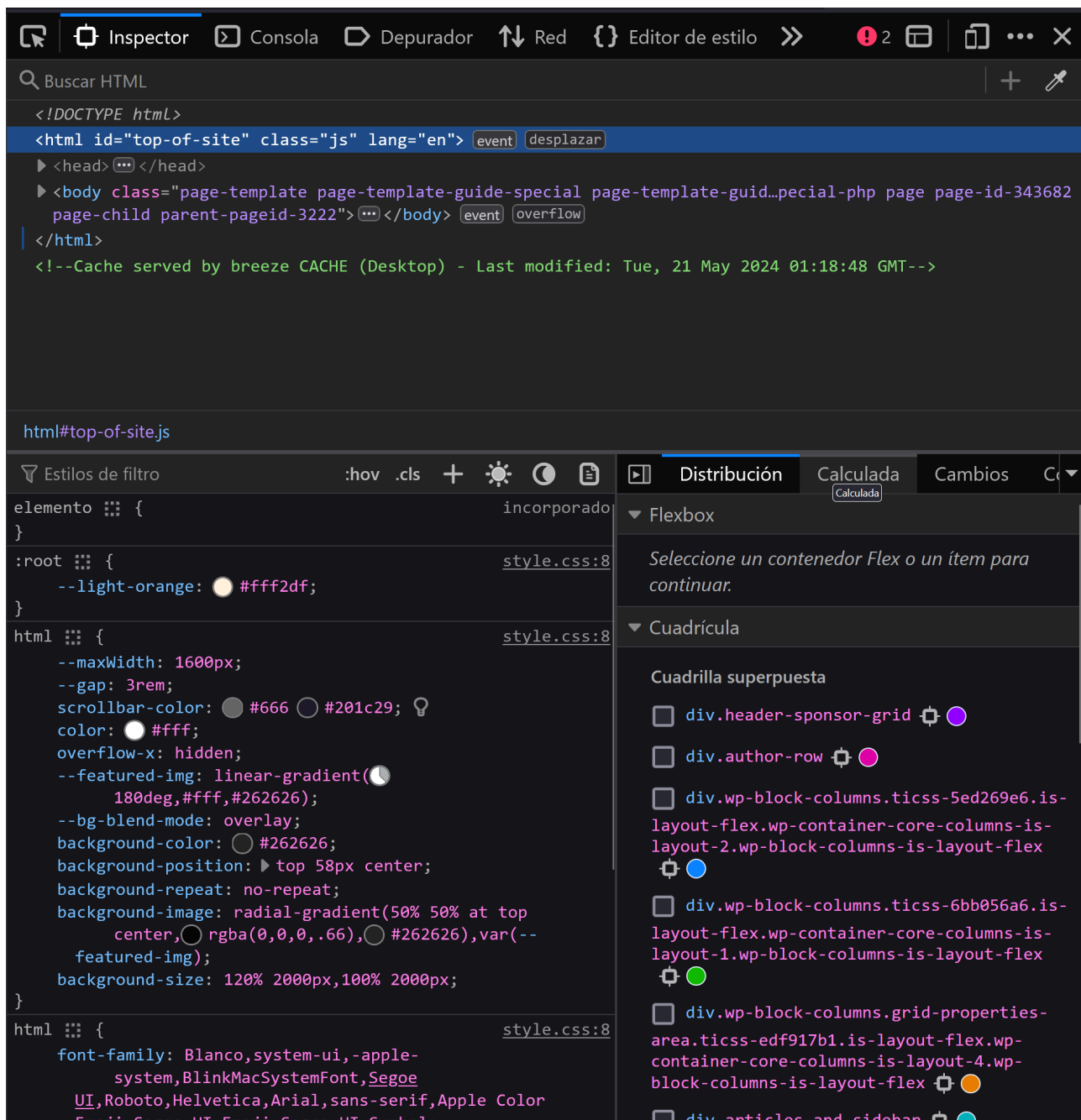
Además de un editor de código, necesitarás un navegador web para visualizar y probar tus proyectos web. Aunque puedes utilizar cualquier navegador web para este propósito, te recomiendo utilizar **Google Chrome** o **Mozilla Firefox**, ya que son dos de los navegadores más compatibles con las tecnologías web modernas.

Puedes descargar Google Chrome desde su [sitio web oficial](#) y Mozilla Firefox desde su [sitio web oficial](#). Una vez que hayas instalado un navegador web en tu computadora, estarás listo para comenzar a programar con JavaScript.

# Consola del navegador

La consola del navegador es una herramienta que te permite interactuar con el código JavaScript de una página web en tiempo real. Puedes utilizar la consola del navegador para depurar tu código, probar nuevas funcionalidades y ver los resultados de tus scripts JavaScript. Para abrir la consola del navegador, sigue estos pasos:

1. Abre tu navegador web (Google Chrome o Mozilla Firefox).
2. Haz clic derecho en cualquier parte de la página web y selecciona la opción "Inspeccionar" o "Inspeccionar elemento". También puedes utilizar el atajo de teclado `Ctrl + Shift + I` en Windows o `Cmd + Option + I` en macOS.



3. Se abrirá la herramienta de desarrollo del navegador. Haz clic en la pestaña "Consola" para acceder a la consola del navegador.

## Estructura del repositorio

Para organizar tus proyectos y archivos de código, te recomiendo crear una estructura de carpetas y archivos coherente en tu repositorio. A continuación, te muestro una estructura de carpetas y archivos básica para un proyecto web:

```
1 mi-proyecto/  
2 |  
3 └─ index.html  
4 └─ styles.css  
5 └─ script.js
```

En esta estructura de carpetas y archivos:

- `index.html` es el archivo principal de tu proyecto, donde se encuentra el código HTML de tu página web.
- `styles.css` es el archivo de estilos de tu proyecto, donde se encuentra el código CSS que define la apariencia de tu página web.
- `script.js` es el archivo de scripts de tu proyecto, donde se encuentra el código JavaScript que añade interactividad y funcionalidad a tu página web.

Puedes añadir más archivos y carpetas según las necesidades de tu proyecto.

Con esta estructura de carpetas y archivos, podrás organizar y gestionar tus proyectos de forma eficiente. Te recomiendo crear una carpeta para cada proyecto y mantener los archivos relacionados con ese proyecto dentro de esa carpeta.

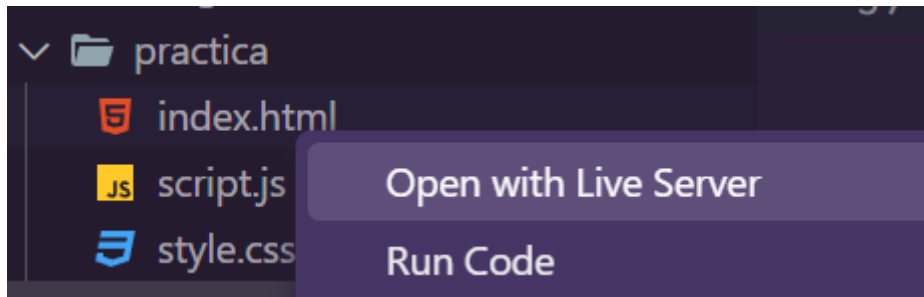
## ¡Empieza a explorar y crear desde el primer día!

Ahora que has configurado tu entorno de desarrollo, es hora de empezar a explorar y crear con JavaScript. En la carpeta de este capítulo encontrarás un ejemplo práctico en la carpeta `practica` para que puedas empezar a experimentar con JavaScript desde el primer día.

Para ejecutar el ejemplo práctico, sigue estos pasos:

1. Descarga la carpeta `practica` a tu computadora.
2. Abre la carpeta `practica` en tu editor de código (Visual Studio Code).

3. Haz clic derecho en el archivo `index.html` y selecciona la opción "Abrir con Live Server".



4. Se abrirá una ventana del navegador con el ejemplo práctico. Puedes interactuar con el ejemplo y ver cómo funciona el código JavaScript.

## index.html

```
1 <!DOCTYPE html>
2 <html lang="es">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-
6       scale=1.0" />
7     <link rel="stylesheet" href="./styles.css" />
8     <title>Canvas Circle</title>
9   </head>
10  <body>
11    <canvas id="myCanvas" width="500" height="500"></canvas>
12    <p id="area"></p>
13    <script src="script.js"></script>
14  </body>
</html>
```

## script.js

```
1 document.addEventListener("DOMContentLoaded", function () {
2   const canvas = document.getElementById("canvas");
3   const ctx = canvas.getContext("2d");
4
5   const centerX = canvas.width / 2;
6   const centerY = canvas.height / 2;
7
8   const radius = 120;
9
10  // Dibujar el círculo
```

```

11   ctx.beginPath();
12   ctx.arc(centerX, centerY, radius, 0, 2 * Math.PI);
13   ctx.strokeStyle = "#000000";
14   ctx.stroke();
15
16   // Dibujar la línea del radio
17   ctx.beginPath();
18   ctx.moveTo(centerX, centerY);
19   ctx.lineTo(centerX + radius, centerY);
20   ctx.strokeStyle = "#FF0000";
21   ctx.stroke();
22
23   // Calcular el área del círculo
24   const area = Math.PI * Math.pow(radius, 2);
25
26   // Mostrar el área en el documento
27   document.getElementById(
28     "area"
29   ).textContent = `El área del círculo es: ${area.toFixed(
30     2
31   )} unidades cuadradas.`;
32 });

```

## style.css

```

1  * {
2    box-sizing: border-box;
3  }
4
5  body {
6    margin: 0;
7    font-family: system-ui, sans-serif;
8    color: black;
9    background-color: white;
10 }
11
12 nav {
13   display: flex;
14   flex-wrap: wrap;
15   align-items: center;
16   justify-content: center;

```

```

17 padding: 0.5rem;
18 gap: 0.5rem;
19 border-bottom: solid 1px #aaa;
20 background-color: #eee;
21 }
22
23 nav a {
24 display: inline-block;
25 min-width: 9rem;
26 padding: 0.5rem;
27 border-radius: 0.2rem;
28 border: solid 1px #aaa;
29 text-align: center;
30 text-decoration: none;
31 color: #555;
32 }
33
34 nav a[aria-current="page"] {
35 color: #000;
36 background-color: #d4d4d4;
37 }
38
39 main {
40 padding: 1rem;
41 }
42
43 h1 {
44 font-weight: bold;
45 font-size: 1.5rem;
46 }
47
48 #myCanvas {
49 border: 1px solid #000;
50 }

```

## Descripción del ejemplo práctico

En este ejemplo práctico, hemos creado un círculo en un lienzo HTML5 utilizando JavaScript. El círculo se dibuja en el centro del lienzo con un radio de 120 píxeles. También hemos dibujado una línea que representa el radio del círculo y hemos calculado el área del círculo utilizando la fórmula  $\pi * r^2$ .



Iremos desglosando el código y explicando cada parte en detalle en los capítulos siguientes. Por ahora, puedes ejecutar el ejemplo práctico y experimentar con el código para ver cómo funciona.

## Archivo `index.html`

En el archivo `index.html`, es donde se define la estructura básica de la página web. HTML es un lenguaje de marcado que se utiliza para definir la estructura y el contenido de una página web. Este archivo HTML se define primero con la etiqueta `<!DOCTYPE html>` que indica al navegador que se trata de un documento HTML5. A continuación, se definen las etiquetas `<html>`, `<head>` y `<body>` que contienen el contenido de la página web.

En el `<head>` se definen los metadatos de la página, como el juego de caracteres, la vista previa y el título de la página. También se enlaza el archivo de estilos `style.css` que define la apariencia de la página web. En el `<body>` se define el lienzo `<canvas>` donde se dibujará el círculo y se enlaza el archivo de scripts `script.js` que contiene el código JavaScript para dibujar el círculo.

```
1 <canvas id="canvas"></canvas>
```

En esta línea, se define un lienzo `<canvas>` con el identificador `canvas`. El lienzo es un elemento HTML5 que se utiliza para dibujar gráficos, animaciones y juegos en una página web. En este caso, utilizaremos el lienzo para dibujar el círculo y la línea del radio.

```
1 <script src="./script.js"></script>
```

En esta línea, se enlaza el archivo de scripts `script.js` que contiene el código JavaScript para dibujar el círculo y calcular el área. El archivo de scripts se carga al final del `<body>` para asegurar que el lienzo y los elementos HTML estén disponibles antes de ejecutar el código JavaScript.

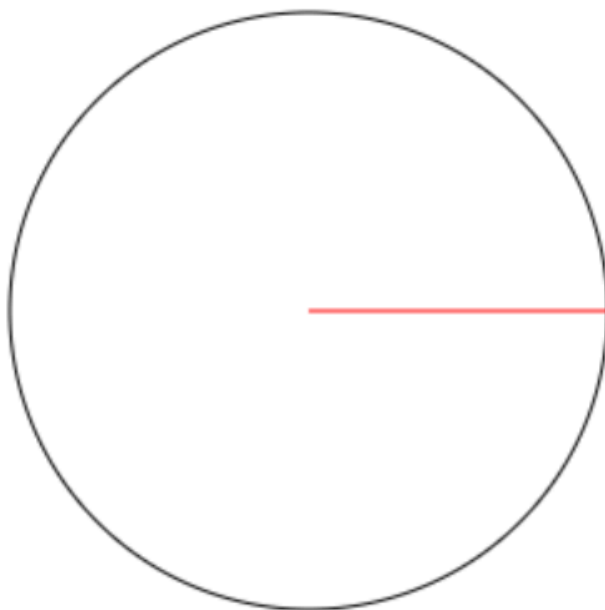
## Archivo `script.js`

En el archivo `script.js`, es donde se define el código JavaScript que dibuja el círculo en el lienzo y calcula el área del círculo. JavaScript es un lenguaje de programación de alto nivel que se utiliza para añadir interactividad y funcionalidad a una página web. En este caso, utilizaremos JavaScript para dibujar el círculo y calcular el área del círculo.

Además, el código JavaScript realiza una manipulación del DOM para mostrar el área del círculo en el documento. El DOM (Document Object Model) es una representación en forma de árbol de la estructura de una página web que permite acceder y manipular los elementos HTML de la página.

## Archivo `styles.css`

En el archivo `styles.css`, es donde se define el código CSS que define la apariencia de la página web. CSS es un lenguaje de estilos que se utiliza para dar formato y estilo a los elementos HTML de una página web. En este caso, utilizaremos CSS para definir los estilos del lienzo, el texto y los elementos HTML de la página web.



El área del círculo es: 45238.93 unidades cuadradas.

## Conclusión

En este capítulo, hemos introducido la programación creativa y hemos configurado un entorno de desarrollo básico para programar con JavaScript. Hemos explorado las herramientas y tecnologías necesarias para desarrollar proyectos creativos y hemos creado un ejemplo práctico para que puedas empezar a experimentar con JavaScript desde el primer día.

No te preocupes si no entendiste el 99% del código del ejemplo práctico. En los capítulos siguientes, iremos abordando distintos temas y conceptos de JavaScript de forma gradual y detallada. Verás como poco a poco irás comprendiendo y dominando el lenguaje.

## Recursos adicionales

- [Código de la clase](#)
- [Repositorio de material y ejemplos](#)