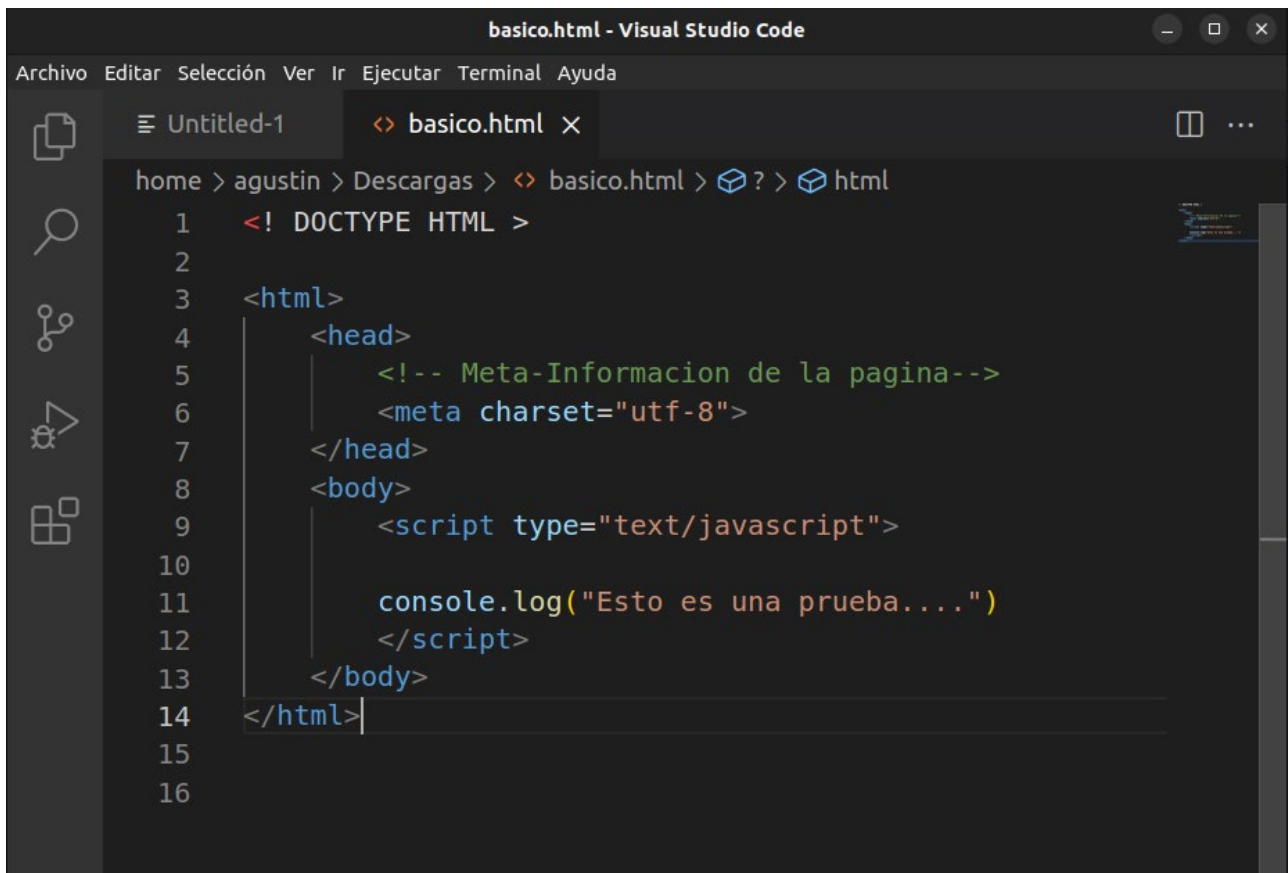


PRÁCTICA

Para más información puedes consultar el manual de devtools de chrome:

- <https://developer.chrome.com/docs/devtools/>

1.- Realiza la estructura básica de una página web



The image shows a screenshot of the Visual Studio Code editor interface. The title bar at the top reads "basico.html - Visual Studio Code". The menu bar includes "Archivo", "Editar", "Selección", "Ver", "Ir", "Ejecutar", "Terminal", and "Ayuda". The Explorer sidebar on the left shows a file named "basico.html" selected. The main editor area displays the following HTML code:

```
home > agustin > Descargas > basico.html > ? > html
1  <!DOCTYPE HTML >
2
3  <html>
4      <head>
5          <!-- Meta-Informacion de la pagina-->
6          <meta charset="utf-8">
7      </head>
8      <body>
9          <script type="text/javascript">
10
11              console.log("Esto es una prueba....")
12          </script>
13      </body>
14  </html>
15
16
```

2.- Describe la funcionalidad de cada una de las pestañas de la consola web:

Inspector: este apartado sirve para identificar errores en la web que estamos inspeccionando. Esta herramienta nos permite cambiar la apariencia y el contenido de una página web editando sus archivos CSS y HTML.

Consola: se utiliza para probar código y nos ayuda a encontrar posibles errores que pueda haber en el código de una página o aplicación.

Fuentes: permite ver la jerarquía de los documentos y nos señala con el depurador si existe algún tipo de error.

Red: se utiliza cuando necesitamos asegurarnos que los recursos se descargan o cargan debidamente.

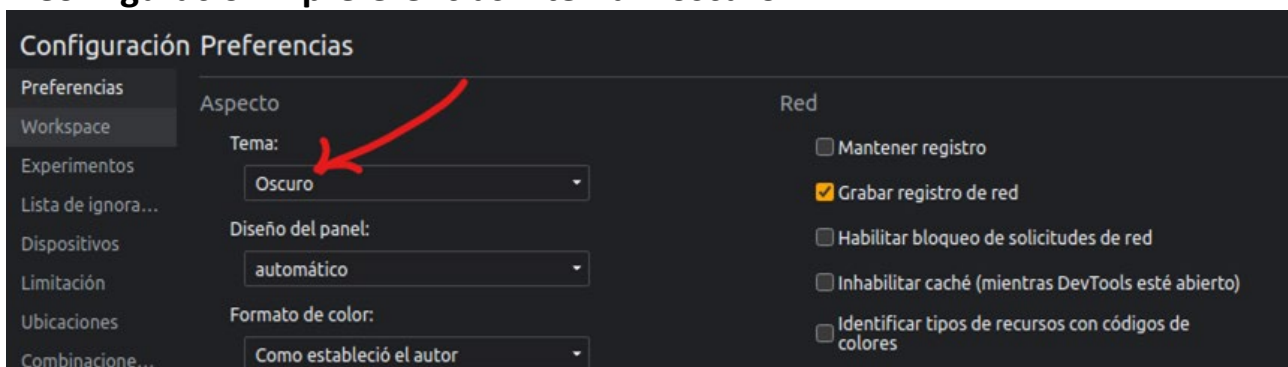
Memoria: te ofrece información del espacio ocupado por los elementos usados en el documento.

3.- Indica los caminos para activar la “consola web” del navegador

- Atajo de teclado: **F12** o **Ctrl+Mayus+I**.
- Opción de menú: **los 3 puntitos arriba a la derecha, más herramientas y a continuación darle a herramientas para desarrolladores.**
-: **Click derecho y le damos a inspeccionar.**

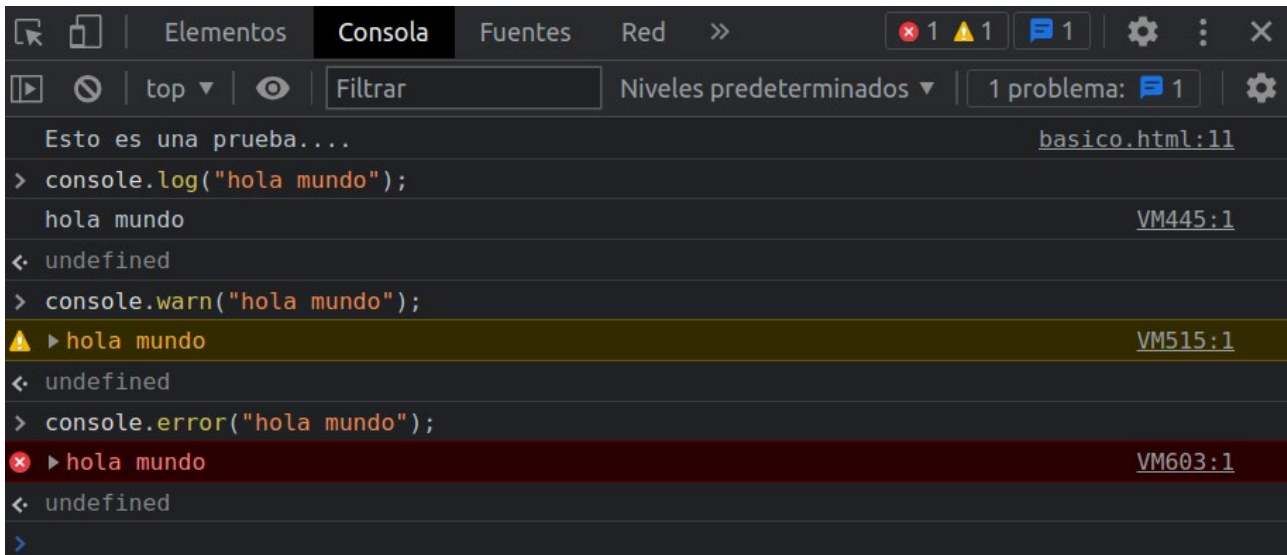
4.- Cambiar tema de apariencia a un color oscuro:

Configuración – preferencias – tema – oscuro



5.- Utilizando la consola del navegador, realizar un “hola mundo” utilizando la función console.

```
console.log
console.warn
console.error
```



```
Esto es una prueba... basico.html:11
> console.log("hola mundo");
hola mundo VM445:1
< undefined
> console.warn("hola mundo");
! ▶ hola mundo VM515:1
< undefined
> console.error("hola mundo");
x ▶ hola mundo VM603:1
< undefined
>
```

6.- Añadir parámetros gráficos a la función console, usada en el punto anterior:

- Para aumentar el tamaño.
- Para cambiar el color



```
1 problema: 1
> console.log("%c aumentar el tamaño", "font-size:20px;");
aumentar el tamaño VM2029:1
< undefined
> console.log("%c cambiar el color", "color:black; background-color:pink");
cambiar el color VM2041:1
< undefined
> |
```

7.- Insertar un array con el siguiente contenido, mostrar a continuación su estructura:



```
> var empleados = [{ 'nombre': 'Miriam', 'edad': 77 },  
                    { 'nombre': 'Jimena', 'edad': 93 },  
                    { 'nombre': 'Susana', 'edad': 55 }]; console.log(empleados);
```

▼ (3) [{...}, {...}, {...}] ⓘ

- ▶ 0: {nombre: 'Miriam', edad: 77}
- ▶ 1: {nombre: 'Jimena', edad: 93}
- ▶ 2: {nombre: 'Susana', edad: 55}


length: 3

▶ [[Prototype]]: Array(0)

← undefined

>

8.- Mostrar el array anterior en formato tabla:



```
> console.table(empleados);
```

VM1194:1

(índice)	nombre	edad
0	'Miriam'	77
1	'Jimena'	93
2	'Susana'	55

▶ Array(3)

← undefined

9.-Cuál es el significado/uso de \$0,\$1....:

\$0: apuntará al último nodo de selección.

\$1: se apuntará al nodo que se seleccionó anteriormente.

Almacenan su historial de selección reciente en una variable especial accesible en la consola. Su selección más reciente (en el inspector de elementos o en la página con 'Inspeccionar elemento') se almacena en \$0. Su segundo más reciente es \$1, y así sucesivamente, hasta \$4.

10.- Escribe el siguiente código en tu navegador, comenta y analiza sus características:

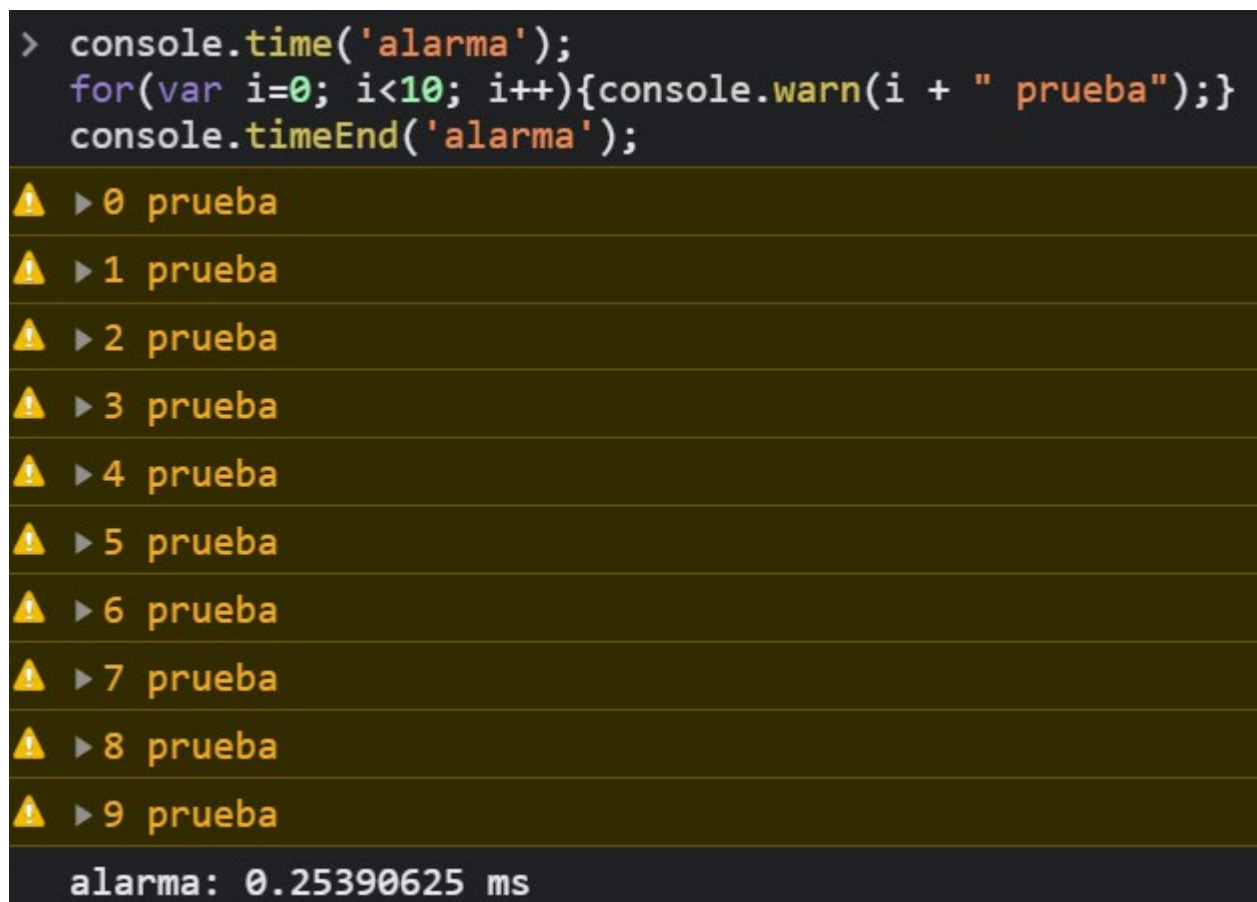
A screenshot of a web browser's developer console. The 'Console' tab is selected. The code entered is:

```
> console.time('loop');  
for(var i=0; i<10000; i++){console.log(i);}   
console.timeEnd('loop');
```

Usamos `console.time()` para realizar un seguimiento de la duración de una operación. Lo que se encuentra dentro es 'loop', el nombre del temporizador. Al final se vuelve a llamar al método, pero en este caso con `console.timeEnd()` lo que finaliza el temporizador, de nuevo añadiendo el nombre.

En medio nos encontramos con el bucle `for`, que va incrementando (`i++`) de 0 (`i=0`) a 9999 como lo indica (`i<10000`). Esto sale en la consola debido al `console.log(i)`.

Prueba:

A screenshot of a web browser's developer console. The code entered is:

```
> console.time('alarma');  
for(var i=0; i<10; i++){console.warn(i + " prueba");}  
console.timeEnd('alarma');
```

The console output shows ten warning messages:
▶ 0 prueba
▶ 1 prueba
▶ 2 prueba
▶ 3 prueba
▶ 4 prueba
▶ 5 prueba
▶ 6 prueba
▶ 7 prueba
▶ 8 prueba
▶ 9 prueba
At the bottom, the timing result is displayed: `alarma: 0.25390625 ms`

11.- Escribe el siguiente código en tu navegador, comenta y analiza sus características:

```
> [1,2,4,5];
< ▼ (4) [1, 2, 4, 5] ⓘ
  0: 1
  1: 2
  2: 4
  3: 5
  length: 4
  ► [[Prototype]]: Array(0)

> $ _
< ▼ (4) [1, 2, 4, 5] ⓘ
  0: 1
  1: 2
  2: 4
  3: 5
  length: 4
  ► [[Prototype]]: Array(0)

> $_.length
< 4
>
```

Creamos un array de cuatro posiciones con lo sintaxis siguiente: [1,2,4,5];

Observamos que se almacenan en las cuatro posiciones (0-3).

Invocamos al array anterior con el siguiente comando: \$ _

Por último, vemos la longitud del array con el método length.

12.- ¿Qué IDE prefieres? Indica las principales características:

Prefiero Visual Studio Code porque es con el trabajamos el año pasado y me gusta cómo se organizan las carpetas, las ventanas y la variedad de lenguajes que soporta.

Características:

Multiplataforma: es una característica importante en cualquier aplicación y más si trata de desarrollo. Visual Studio Code está disponible para Windows, GNU/Linux y macOS.

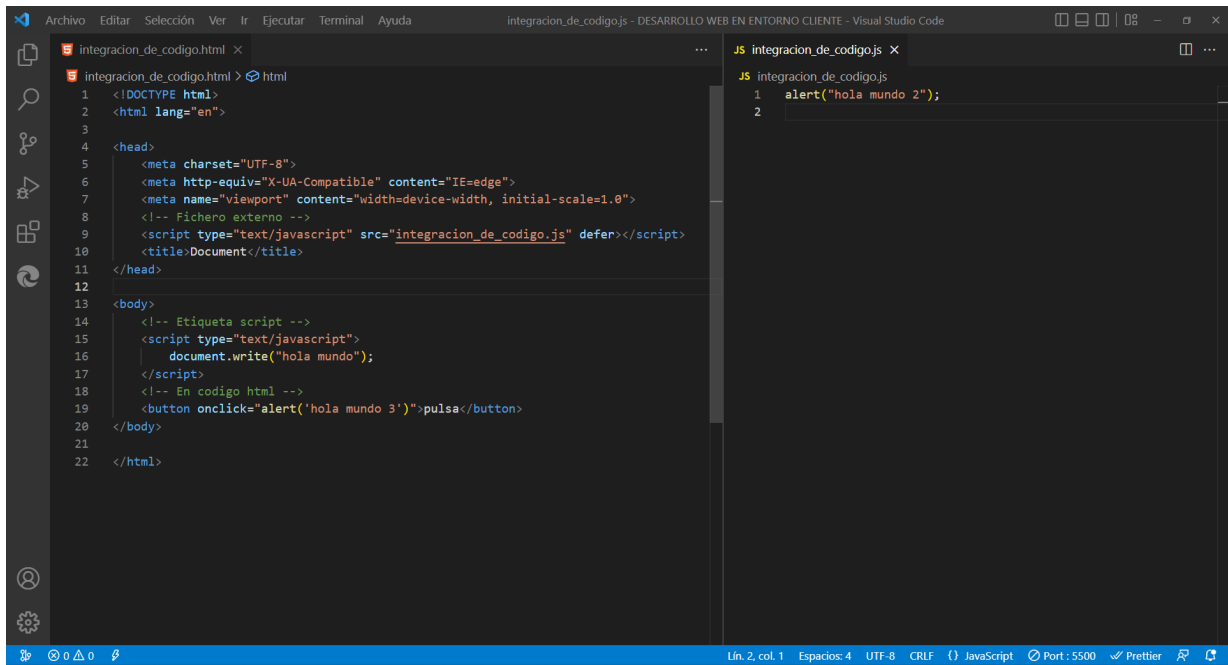
IntelliSense: esta característica está relacionada con la edición de código, autocompletado y resaltado de sintaxis, lo que permite ser más ágil a la hora de escribir código.

Depuración: incluye la función de depuración que ayuda a detectar errores en el código.

Uso del control de versiones: tiene compatibilidad con Git.

Extensiones: las extensiones nos permiten personalizar y agregar funcionalidad adicional de forma modular y aislada

13 .- Ejercicio de integración de código. Realizar un programa que muestre un “hola mundo”, utilizando los 3 mecanismos de integración de código.



The screenshot shows the Visual Studio Code editor with two files open: `integracion_de_codigo.html` and `integracion_de_codigo.js`. The HTML file contains a meta tag to load the external script and a button that triggers an alert. The JavaScript file contains the `alert` function.

```
integracion_de_codigo.html
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <!-- Fichero externo -->
9   <script type="text/javascript" src="integracion_de_codigo.js" defer></script>
10  <title>Document</title>
11 </head>
12
13 <body>
14   <!-- Etiqueta script -->
15   <script type="text/javascript">
16     document.write("hola mundo");
17   </script>
18   <!-- En codigo html -->
19   <button onclick="alert('hola mundo 3')">pulsa</button>
20 </body>
21
22 </html>
```

```
integracion_de_codigo.js
1 alert("hola mundo 2");
2
```



14.- Utilizar el validador de W3C y analizar alguna página conocida que utilices con frecuencia.

- Analiza los resultados ofrecidos después del proceso de validación.

<https://validator.w3.org/>

Nu Html Checker

Esta herramienta es un experimento en curso para una mejor verificación HTML, y su comportamiento sigue sujeto a cambios.

Mostrando resultados para <https://app.pomodortimer.online/es/>

Entrada de la marque

Mostrar ☐ fuente ☐ esquema ☐ informe de imagen

Verificar por

Use el botón Filtrado de mensajes a continuación para ocultar / mostrar mensajes particulares y para ver el recuento total de errores y advertencias.

Filtrado de mensajes

- Error** Elemento `style` no permitido como hijo del elemento `div` en este contexto. (Suprimiendo más errores de este subárbol.)
Desde la línea 7, columna 22; alinear 7, columna 28
`<div><style></style></div>`
Contextos en qué elemento `style` puede ser usado:
Donde `contenido de meladatos` es esperado.
En un `noscript` elemento que es hijo de un `head` elemento.
Modelo de contenido para elemento `div`:
Si el elemento es hijo de un `div` elemento: uno o más `div` elementos seguidos de uno o más `div` elementos, opcionalmente entremezclados con `script` - elementos de soporte.
Si el elemento no es hijo de un `div` elemento: `contenido de flujo`.
- Error** A `script` elemento con un `src` el atributo no debe tener un `type` atributo cuyo valor es cualquier cosa que no sea la cadena vacía, un tipo MIME de JavaScript o `module`.
Desde la línea 8, columna 3; alinear 8, columna 82
`<script src="/nuxt/0106533.js" type="b194b6a1cef2d56f0bc4bff3-text/javascript"></script>`
- Error** A `script` elemento con un `src` el atributo no debe tener un `type` atributo cuyo valor es cualquier cosa que no sea la cadena vacía, un tipo MIME de JavaScript o `module`.
Desde la línea 8, columna 92; alinear 8, columna 171
`<script src="/nuxt/012186a.js" type="b194b6a1cef2d56f0bc4bff3-text/javascript"></script>`
- Error** A `script` elemento con un `src` el atributo no debe tener un `type` atributo cuyo valor es cualquier cosa que no sea la cadena vacía, un tipo MIME de JavaScript o `module`.
Desde la línea 8, columna 181; alinear 8, columna 260
`<script src="/nuxt/012186a.js" type="b194b6a1cef2d56f0bc4bff3-text/javascript"></script>`

Nos indica los errores de sintaxis que tiene la página. El primero error lo encontramos en el `<style>`. Vemos la línea donde se encuentra el error y a continuación la explicación.

From line 7, column 22; to line 7, column 28

La etiqueta de estilo no está permitido como hijo de un `div`. Lo comprobamos revisando el código.

```
<div>
  <noscript>
    <iframe src="https://www.googletagmanager.com/ns.html?id=GTM-PMPRG34" height="0" width="0" style="display: none; visibility: hidden;">
  </noscript>
  <#document>
    <!DOCTYPE html>
    <html dir="ltr" lang="es" subframe>
      <head>
        <meta charset="utf-8">
        <meta name="color-scheme" content="light dark">
        <meta name="theme-color" content="#fff">
        <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no">
        <title>www.googletagmanager.com</title>
        <style> == $0
        /* Copyright 2017 The Chromium Authors. All rights reserved.
         * Use of this source code is governed by a BSD-style license that can be
         * found in the LICENSE file. */
```