

Web garapena bezeroen ingurunean

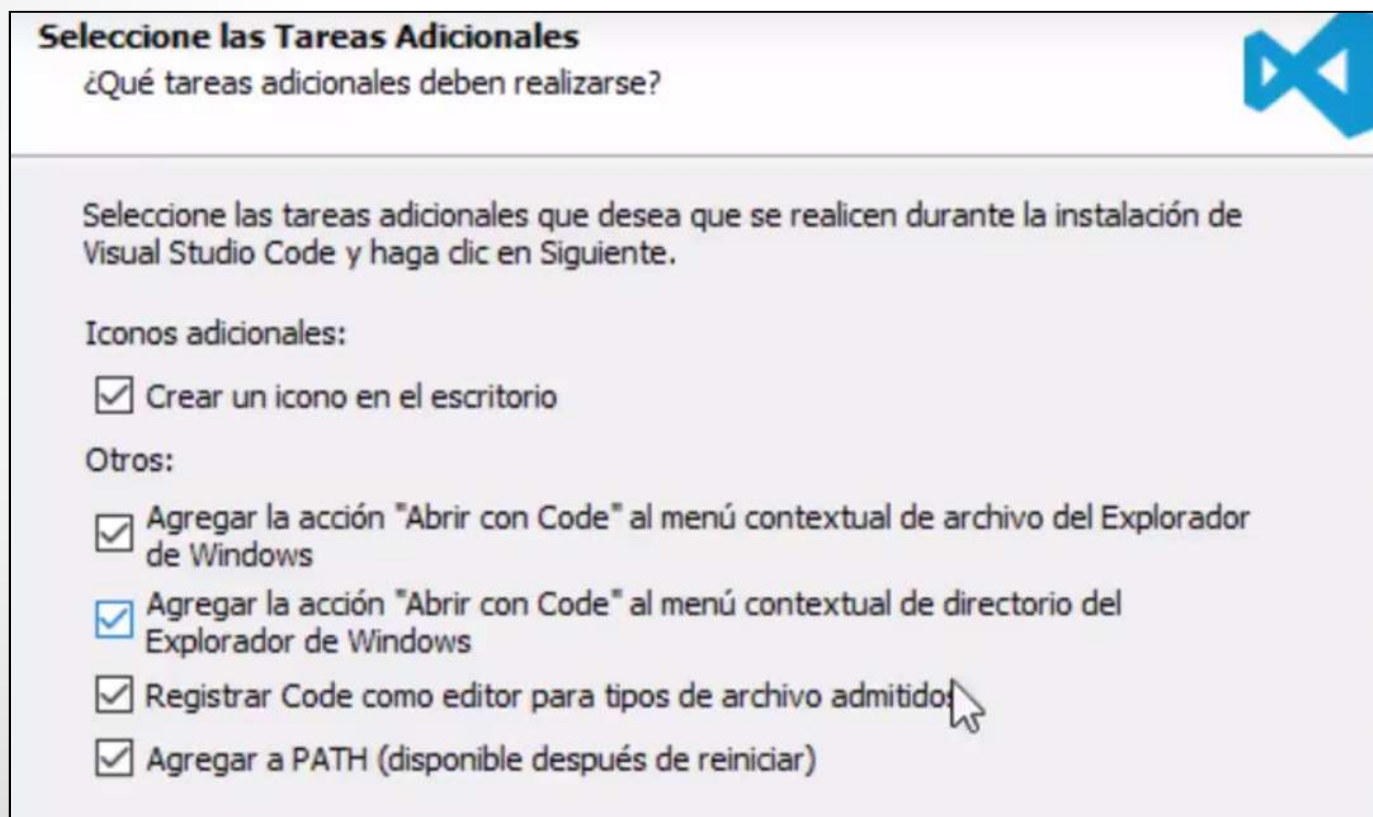
Erronka 1. Javascript-sarrera (Manual-Javascript-1)

1. Editoreak
2. Berrikuspena:
 - a) Nola idatzi Javascript-a HTML kodigo barruan
 - b) Sintaxia
 - c) Aldagaiak
 - d) Eragileak
 - e) Kodearen fluxua kontrolatzeko egiturak
 - f) Funtzioak
 - g) Objektuak
 - h) Array
 - i) Erroreak

Erronka 1. Javascript-sarrera

Javascript idazteko erabil daitekeen editore bat Visual Studio Code da, beraz, nola instalatu eta konfiguratu azalduko da.

Instalatzerakoan aukera hauek markatuko ditugu:



The screenshot shows the 'Selección de tareas adicionales' (Additional tasks selection) window during the Visual Studio Code installation. The window has a title bar with the text 'Selección de tareas adicionales' and a subtitle '¿Qué tareas adicionales deben realizarse?'. The Visual Studio Code logo is in the top right corner. The main content area contains the following text: 'Seleccione las tareas adicionales que desea que se realicen durante la instalación de Visual Studio Code y haga clic en Siguiente.' (Select the additional tasks you want to be performed during the installation of Visual Studio Code and click on Next). Below this, there are two sections: 'Iconos adicionales:' (Additional icons:) and 'Otros:' (Others:). Under 'Iconos adicionales:', there is a checkbox labeled 'Crear un icono en el escritorio' (Create an icon on the desktop) which is checked. Under 'Otros:', there are four checkboxes, all of which are checked: 'Agregar la acción "Abrir con Code" al menú contextual de archivo del Explorador de Windows' (Add the "Open with Code" action to the file context menu of Windows Explorer), 'Agregar la acción "Abrir con Code" al menú contextual de directorio del Explorador de Windows' (Add the "Open with Code" action to the directory context menu of Windows Explorer), 'Registrar Code como editor para tipos de archivo admitidos' (Register Code as editor for supported file types), and 'Agregar a PATH (disponible después de reiniciar)' (Add to PATH (available after restart)).

Selección de tareas adicionales
¿Qué tareas adicionales deben realizarse?

Seleccione las tareas adicionales que desea que se realicen durante la instalación de Visual Studio Code y haga clic en Siguiente.

Iconos adicionales:

- ☒ Crear un icono en el escritorio

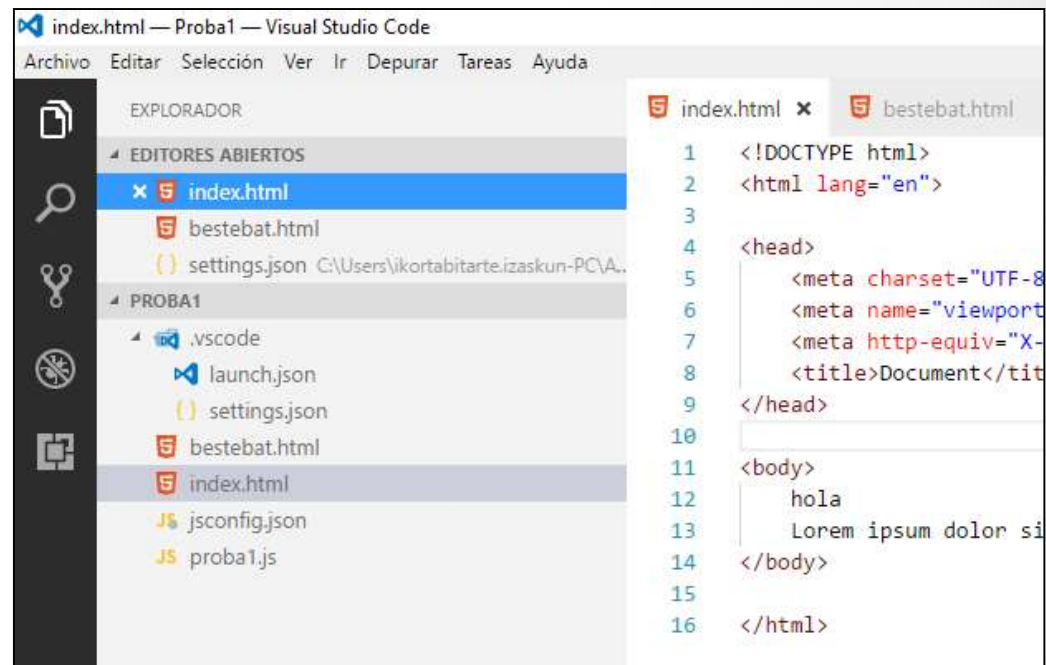
Otros:

- ☒ Agregar la acción "Abrir con Code" al menú contextual de archivo del Explorador de Windows
- ☒ Agregar la acción "Abrir con Code" al menú contextual de directorio del Explorador de Windows
- ☒ Registrar Code como editor para tipos de archivo admitidos
- ☒ Agregar a PATH (disponible después de reiniciar)

Erronka 1. Javascript-sarrera

Editorea konfiguratzeko, lehenengo luzapen batzuk instalatuko ditugu, momentuz hauek:

- HTML CSS Support
- Close HTML/XML tag
- vscode-icons
- JavaScript Debugger(Nightly)
- Debugger for Firefox
- Live Server
- Prettier-Code formatter
- Code Blue (Tema baltza)



- Hobespenak egin daitezke ere:
 - Archivo – Preferencias – Tema de color
 - Archivo – Preferencias – Tema de icono de archivos (vscode-icons)

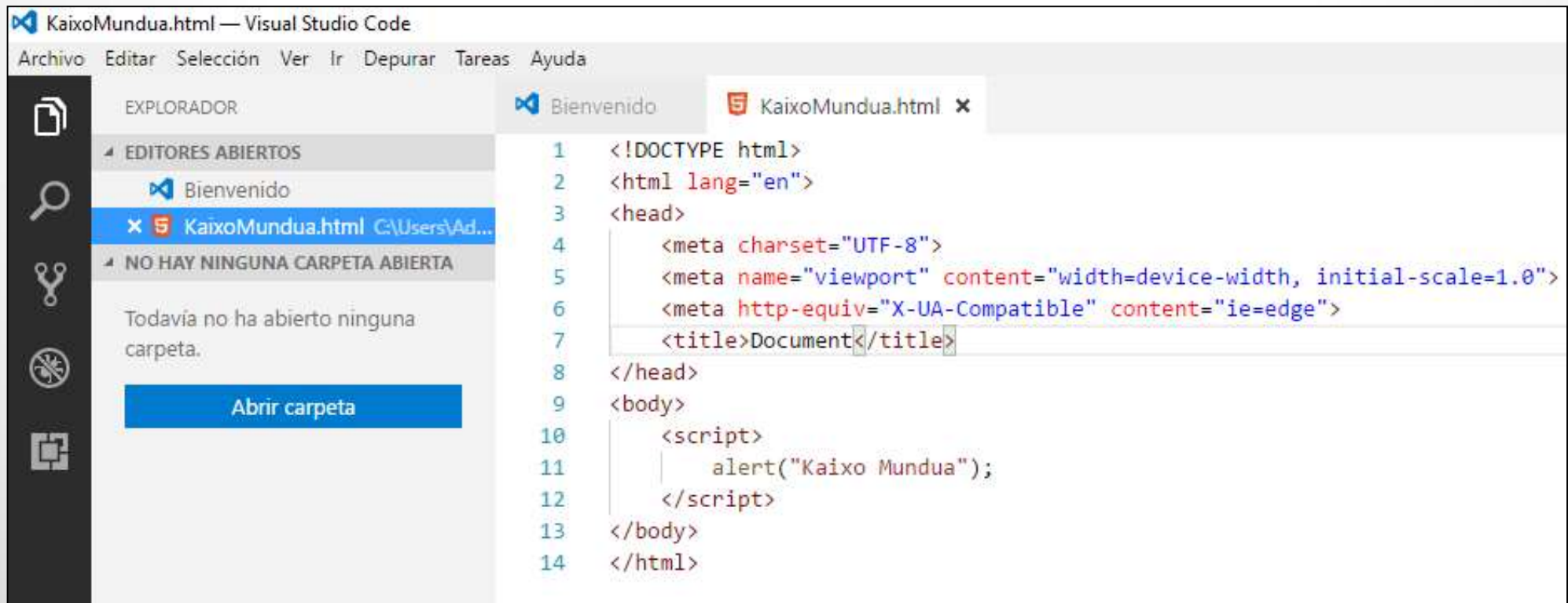
Erronka 1. Javascript-sarrera

Adibidea: Kaixo Mundua

Web berria egingo dugu artxibo berri batean:

"KaixoMundua.html"

Editorean **!** Karakterra idatzi eta **<tab>** egin.



```
KaixoMundua.html — Visual Studio Code
Archivo  Editar  Selección  Ver  Ir  Depurar  Tareas  Ayuda

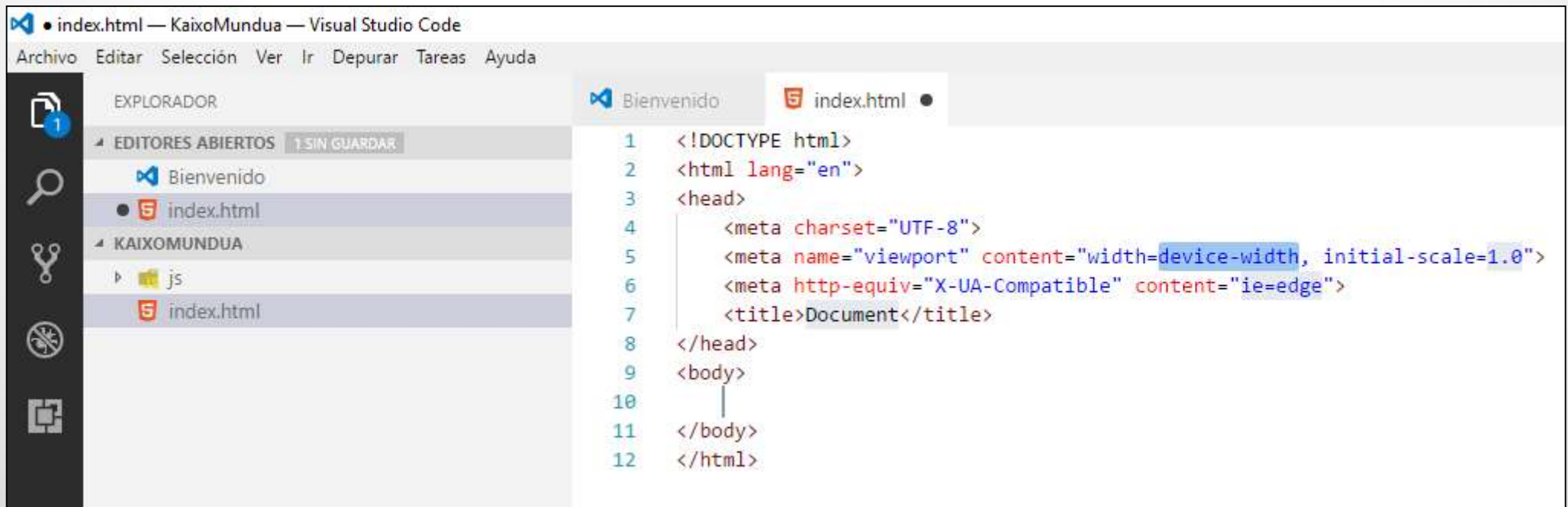
EXPLORADOR
  EDITORES ABIERTOS
    Bienvenido
    x KaixoMundua.html C:\Users\Ad...
  NO HAY NINGUNA CARPETA ABIERTA
    Todavía no ha abierto ninguna carpeta.
    Abrir carpeta

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7      <title>Document</title>
8  </head>
9  <body>
10     <script>
11         alert("Kaixo Mundua");
12     </script>
13 </body>
14 </html>
```

Erronka 1. Javascript-sarrera

Adibidea: Kaixo Mundua

Gauza berdina egingo dugu baina kanpoko javascript fitxategi bat erabiliz: Lehenengo karpeta bat sortuko dugu: "**KaixoMundua**" Eta fitxategi berria sortu, "**index.html**" eta azpikarpeta bat "**js**" Editorean **!** Karakterra idatzi eta **<tab>** egin



The screenshot shows the Visual Studio Code interface with the KaixoMundua project open. The Explorer sidebar on the left shows the project structure: a folder named 'KAIXOMUNDUA' containing a 'js' folder and an 'index.html' file. The main editor area displays the content of 'index.html'.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <title>Document</title>
8 </head>
9 <body>
10   |
11 </body>
12 </html>
```

Erronka 1. Javascript-sarrera

Adibidea: Kaixo Mundua

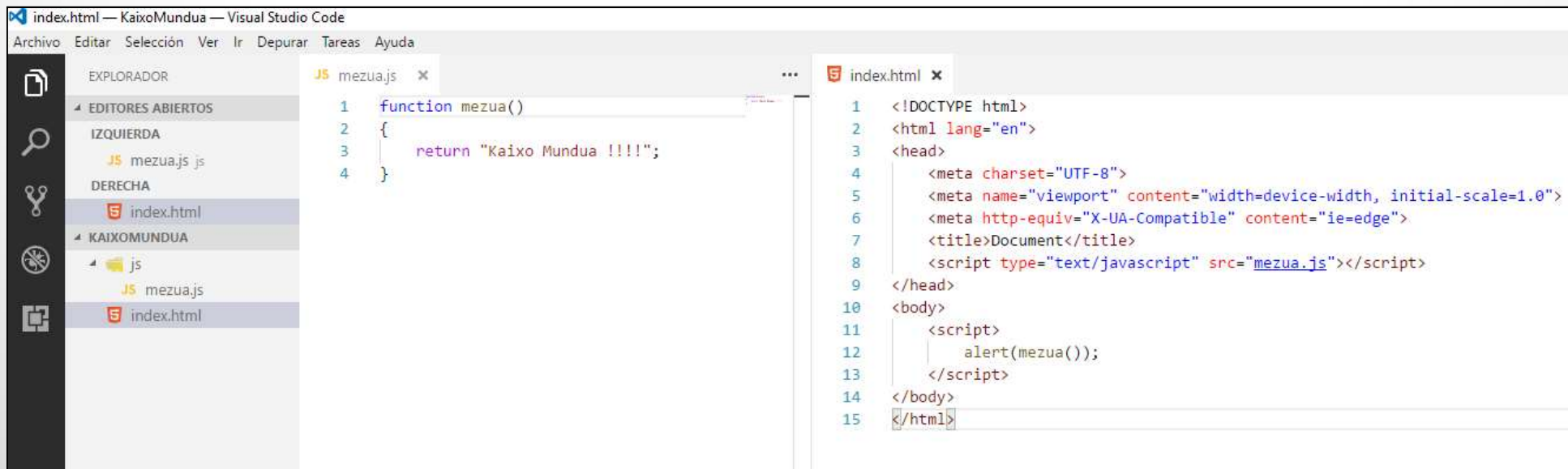
Js karpeta barruan fitxategia sortu: **mezua.js**

Kodigo hau jarri:

```
function mezua()  
{    return "Kaixo Mundua !!!";}
```

eta fijatu index.html aldatu egin behar dela ere, eta lerro hau gehitu:

```
<script type="text/javascript" src="mezua.js"> </script>
```



Erronka 1. Javascript-sarrera

Adibidea: Kaixo Mundua

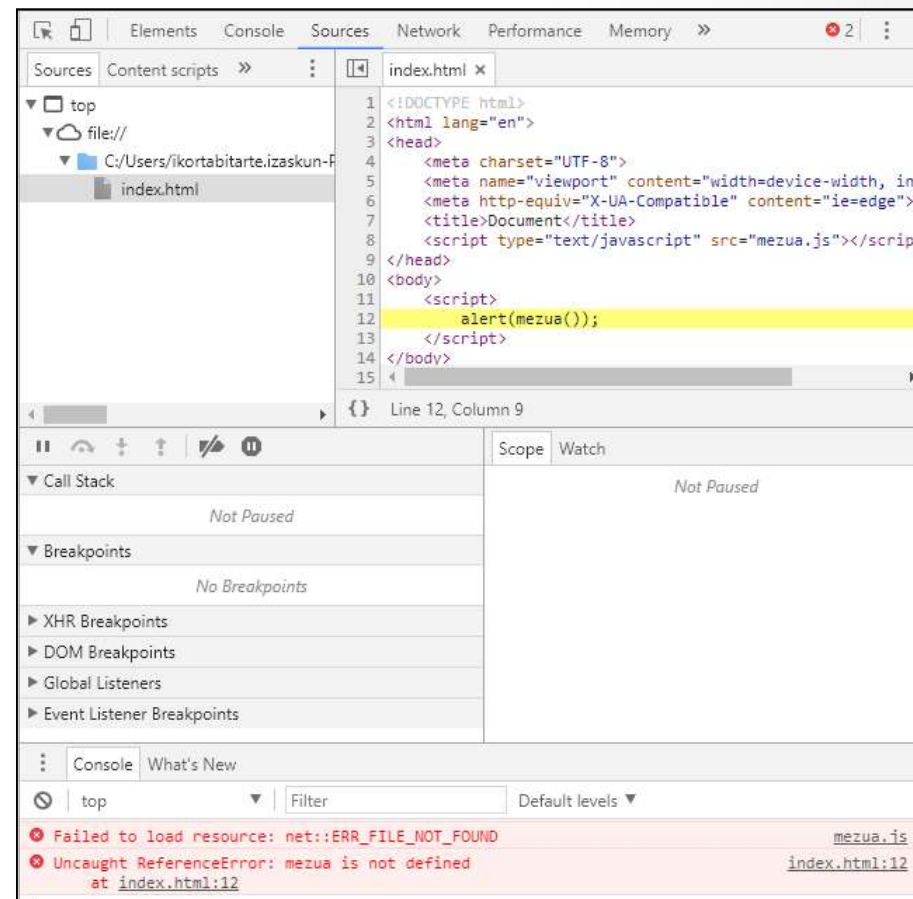
Exekutatzeko denean:

Eta errorea gertatuz gero, hon

mezua.js fitxategia ez du topatzen,
jartzerakoan errore bat dago.

Errore hau konpontzeko aldatu egin
behar dugu:

```
<script type="text/javascript"  
  src= "js/mezua.js"></script>
```

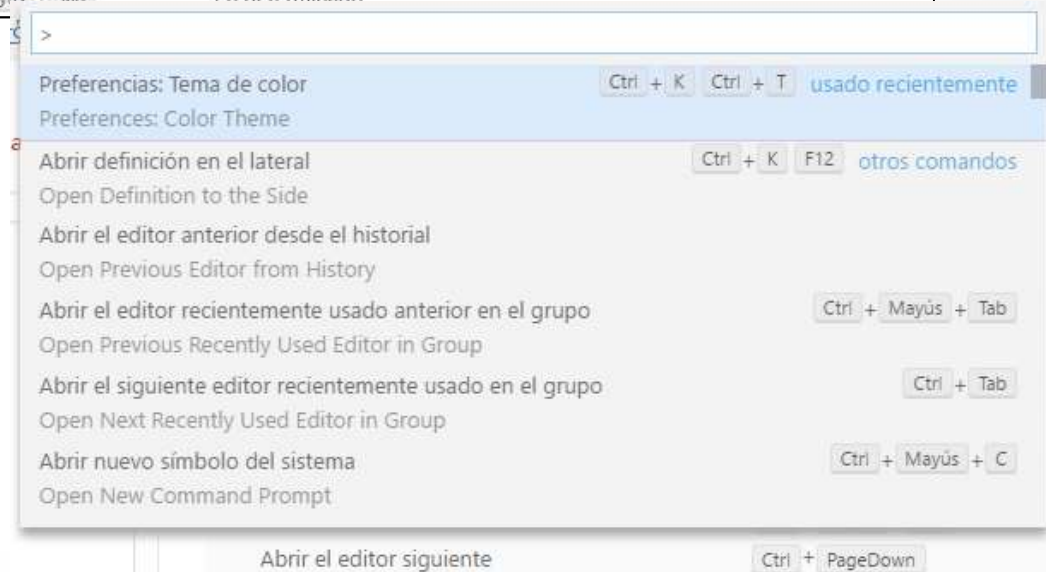


Erronka 1. Javascript-sarrera

Teklen konbinazio laburtuak ikusteko: Archivo – Preferencias – metodos abreviados



Konbinazio garrantzitsuena:
Ctrl+Shift+P
(komandoen paleta
zabaltzeko) eta visual
Studio Code-ko funtzionaltasun
osoa eskura izateko

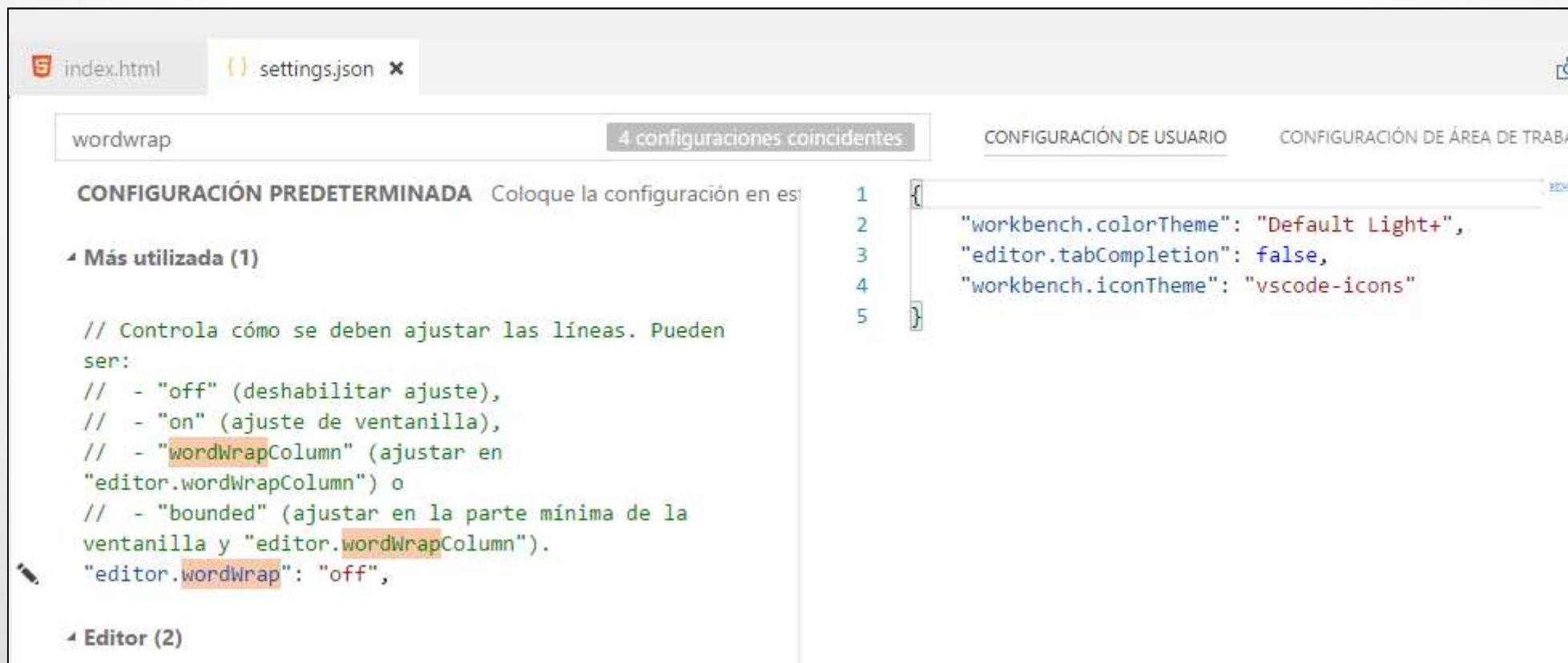


Erronka 1. Javascript-sarrera

Konfigurazio ikusteko: Archivo – Preferencias – Configuración

Hemen, ezkerreko panelean konfigurazio orokorra ikusten dugu eta eskuineko panelean norberaren konfigurazio pertsonala

Bilaketak egin daitezke: adibidez, **"wordwrap"**:



Erronka 1. Javascript-sarrera

Konfigurazio ikusteko: Archivo – Preferencias – Configuración

“**wordwrap**” zertarako balio duen ikusteko, idatzi index.html-an
“**Lorem**” eta **<tab>**. Lehenengo irudian **wordwrap = off**
dago, baina aktibatzen badugu, testua pantailara egokitzen du.

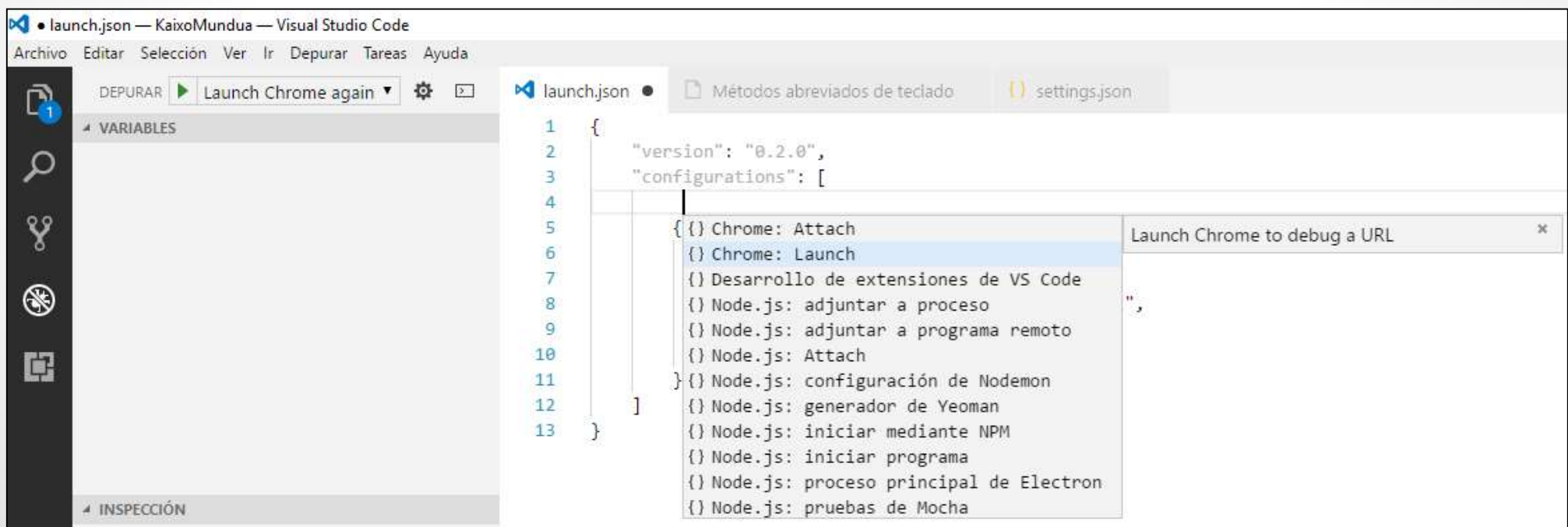
```
index.html • settings.json
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7      <title>Document</title>
8      <script type="text/javascript" src="js/mezua.js"></script>
9  </head>
10 <body>
11     <script>
12         alert(mezua());
13     </script>
14     Lorem ipsum dolor sit amet, consectetur adipisicing elit. Molestias esse qui, delectus, tempora veritatis vo
15 </body>
16 </html>
```

```
<body>
    <script>
        alert(mezua());
    </script>
    Lorem ipsum dolor sit amet, consectetur adipisicing elit. Molestias esse qui, delectus, tempora veritatis
    voluptatibus. Dolore, rem, magnam! Sint magni incidunt temporibus assumenda eligendi placeat atque
    recusandae commodi hic ducimus.
</body>
</html>
```

Erronka 1. Javascript-sarrera

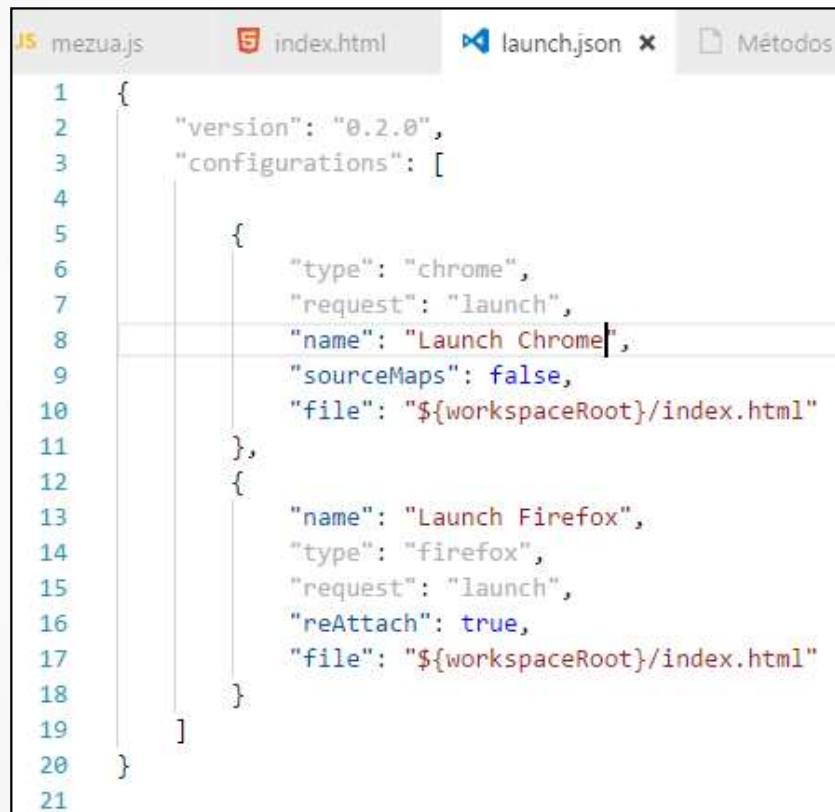
- Visual Studio Code-k depuratzeko tresna du. Luzapenekin erabiliko dugu: "Debugger for Chrome" eta/edo "Debugger for Firefox".

Depurazioa martxan jartzeko konfigurazio fitxategia prestatu behar da (4. ikonoa ezkerreko menuan): **"Agregar archivo de configuración launch.json – Chrome - Launch"** aukeratuta:



Erronka 1. Javascript-sarrera

- <https://code.visualstudio.com/docs/editor/debugging>

A screenshot of the Visual Studio Code editor interface. The top bar shows four tabs: 'mezua.js', 'index.html', 'launch.json', and 'Métodos'. The 'launch.json' tab is active, displaying a JSON configuration for debugging. The code is as follows:

```
1 {
2   "version": "0.2.0",
3   "configurations": [
4     {
5       "type": "chrome",
6       "request": "launch",
7       "name": "Launch Chrome",
8       "sourceMaps": false,
9       "file": "${workspaceRoot}/index.html"
10    },
11    {
12      "name": "Launch Firefox",
13      "type": "firefox",
14      "request": "launch",
15      "reAttach": true,
16      "file": "${workspaceRoot}/index.html"
17    }
18  ]
19 }
20
21
```

- Behin idatzita ikono berdeari klik egin

Erronka 1. Javascript-sarrera

- Beste editore posible bat **WebStorm** da, JetBrains enpresakoa:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Title</title>
6     <script type="text/javascript" src="funtzioak.js">
7     </script>
8 </head>
9 <body>
10     <h3>Lehenengoko aplikazioa</h3>
11     <script type="text/javascript">
12         alert(mezua());
13     </script>
14 </body>
15 </html>
```

Erronka 1. Javascript-sarrera

2. Berrikuspena:

- a) Nola idatzi Javascript-a HTML kodigo barruan
 - Adibidean ikusi dugun bezala eta beste modu honetan:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
  <title></title>
<meta charset="utf-8" />
  <script type="text/javascript" src="js/mezua.js"></script>
</head>
<body>
  <script type="text/javascript">
    alert(mezua());
  </script>

</body>
</html>
```


Erronka 1. Javascript-sarrera

2. Berrikuspena:

b) Sintaxia

i. Komentarioak

- **Komentarioak lerro batean:**

- // Lerro bateko komentarioa

- **Lerro bat baino gehiagoko komentarioak:**

- /* Lerro bat baino
gehiago
erabiltzen*/

- **Zertarako erabiltzen diren komentarioak:**

- Kodea argiago izateko eta aldagaiak eta funtzioak zertarako diren azaltzeko.
 - Frogak egiten gaudenean kode zati bat exekutatu ez dezan.
 - Gure datuak uzteko beste programatzaile batzuentzat.

Erronka 1. Javascript-sarrera

2. Berrikuspena:

c) Aldagaiak

- **Aldagaiak sortu "var" hitz erreserbatua erabiliz:**

var adina;

var adina1, adina2, adina3;

- **Balioa eman aldagai bateri:**

adina = 15;

- **Gauza biak batera egin:**

var adina = 15;

- **Aldagaien erabilera:**

Alfanumerikoak izan behar dira, eta _ erabili ahal da. Ezin da beste karakterrik erabili: hutsunerik, %, \$, etab.

Ez da komeni zenbakitik hasia. Saiatu sistema estandarra erabiltzen.

Ez dauka tipo berezirik esleituta, erabilerak esaten du zer den, zenbakia, letra, booleanoa, etab.

- .

Erronka 1. Javascript-sarrera

2. Berrikuspena:

c) Aldagaiak

i. **Datu motak**

- **Zenbakiak:**

Osoa: 726

Decimala: 3.75

Zientifikoa: 3e7 (30000000)

Octala: 0 bat aurretik: 0327

Hexadecimala: 0x aurretik:
0xA3F2.

- **Kataiak:**

- "Komatxo arteko testua"
- "7342"
- "Katai1" + "Katai2 gehituta"

- **Karaktek bereziak erabiltzeko:**

- Lerro saltoa: \n
- Tabuladorea: \t
- Komatxo bikoitzak: \"
- Komatxo sinplea: \'

Erronka 1. Javascript-sarrera

2. Berrikuspena:

c) Aldagaiak

i. Datu motak

- **Booleanoak:**

true

false

- **Objetuak:**

String

Date

Array

Etab.

.

.

- **Datu mota ezberdinetako aldaketak:**

Osoa + Float = Float

Zenbakia + Kataia = Kataia

- **Kataietik zenbakira aldatu:**

parseInt("32")

parseFloat("32.1")

- **Zenbakitik kataiara aldatu:**

" + zenbakia

Erronka 1. Javascript-sarrera

2. Berrikuspena:
d) Eragileak

Sintaxis	Nombre	Tipos de <u>operandos</u>	Resultados
+	Más	Entero, real, cadena	Entero, real, cadena
-	Menos	Entero, real	Entero, real
*	Multiplicación	Entero, real	Entero, real
/	División	Entero, real	Entero, real
%	Módulo	Entero, real	Entero, real
++	Incremento	Entero, real	Entero, real
--	Decremento	Entero, real	Entero, real
+valor	Positivo	Entero, real, cadena	Entero, real
-valor	Negativo	Entero, real, cadena	Entero, real

Operadores aritméticos

Erronka 1. Javascript-sarrera

2. Berrikuspena:
- d) Eragileak

Sintaxis	Nombre	<u>Operandos</u>	Resultados
&&	And	<u>Boolean</u>	Boolean
	<u>Or</u>	Boolean	Boolean
!	<u>Not</u>	<u>Boolean</u>	<u>Boolean</u>

Operadores booleanos

Erronka 1. Javascript-sarrera

2. Berrikuspena:

d) Eragileak

- **Punto:**
Objeto.propietatea
Objeto.metodoa
- **Kortxeteak:**
Array bat sortu: `var a = ["Bizkaia", "Araba", "Gipuzkoa"]`
Array bateko datu bat: `a[1] = "Araba";`
Objeto bateri balioa eman: `a["color"] = "azul";`
- **Delete:**
`Delete a[2];` // "Gipuzkoa" ezabatuko du eta undefined utzi
- **In:**
`true` bueltatzen du objetoak propietate edo metodo hori badu
Ej: `"write" in document`
- **Instanceof:**
`true` Javascript-eko objetua bada:
Ej: `a = new Array (1,2,3);`
`a instanceof Array;` // `true` bueltatzen du

Erronka 1. Javascript-sarrera

2. Berrikuspena:

d) Eragileak

- **Koma:**

Definizio bat baino gehiago jartzeko: var nombre, direccion, apellidos;

loop (errepikatzeko): for (var i=0, j=0; i<125M i++, j+10)

- **Galdera (baldintzazko operadorea):**

if ... else bertsio murriztua.

Baldintza ? egia bada: gezurra bada;

Ej: var a=3, b=5;

Var r = a > b ? a : b;

- **Typeof:**

- Aldagai edo espresioaren mota bueltatzen du.

- Motak dira: number, string, boolean, object, function, undefined.

- Ej: if (typeof miVariable == "number") alert ("Nire aldagaia number da");

Erronka 1. Javascript-sarrera

2. Berrikuspena:

e) Kontrol egiturak

- **If egitura:**

```
if (baldintza) // ebaluatzen den baldintza, egia edo gezurra
{
    // exekutatzen den kodea egia bada
}
```

Adibidea:

```
If (nireAdina >= 18)
{
    alert("Pertsona nagusia zara");
}
```

Erronka 1. Javascript-sarrera

2. Berrikuspena:

e) Kontrol egiturak

- **if...else egitura:**

```
if (baldintza) // ebaluatzen den baldintza, egia edo gezurra
{
    // exekutatzen den kodea egia bada
}
else
{
    // exekutatzen den kodea gezurra bada
}
```

Adibidea:

```
if (nireAdina >= 18)
{
    alert("Pertsona nagusia zara");
}
else
{
    alert ("Ez zara pertsona nagusia oraindik");
}
```

Erronka 1. Javascript-sarrera

2. Berrikuspena:

e) Kontrol egiturak

- **For buklea:**

```
for (hasiera; baldintza; gehiketa)
```

```
{
```

```
    // bukle barruan exekutatzeko den kodea
```

```
}
```

Adibidea:

```
for (var i=1; i<20; i++)
```

```
{
```

```
    //20 bider egingo den kodea
```

```
}
```

Erronka 1. Javascript-sarrera

2. Berrikuspena:

e) Kontrol egiturak

- **While buklea:**

```
while (baldintza)
{
    // Bukle barruan exekutatzen den kodea
}
```

Adibidea:

```
var i=0;
while (i <=10)
{
    //Exekutatzen den kodea i 10 baino gehiago balio arte
    i++;
}
```


Erronka 1. Javascript-sarrera

2. Berrikuspena:

e) Kontrol egiturak

- **While buklea:**

```
do {  
    // bukle barruko kodea  
} while (baldintza);
```

Adibidea:

```
var i=0;  
do {  
    //exekutatzen den kodea i 3 baino gutxiago den bitartean  
    i++;  
} while (i<3)
```

Erronka 1. Javascript-sarrera

2. Berrikuspena - Ariketak

1. Hiru nota eskatuko dira eta batazbestekoa 5 edo gehiago bada "Gaindituta" mezua erakutsiko dugu. (parseInt(zenb) eta prompt("Idatzi nota") erabili)
2. Eskatu pasahitza sartzeko birritan, eta berdinak badira "Ongi etorri" mezua erakutsi. Ez badira berdinak "Pasahitza okerra" mezua erakutsi. (== operadorea erabili)
2. Pasahitzen ariketa optimizatu: egitura errepikakorra prestatu pasahitza eskatzeko behin eta berriz zuzen sartu arte.

Erronka 1. Javascript-sarrera

2. Berrikuspena:

f) Funtzioak

```
<SCRIPT type="text/javascript" >  
    function miFuncion()  
{  
    document.write("Hau ondo doa");  
}  
</SCRIPT>
```

- Function hitz erreserbatua erabiltzen da
- Parentesiak erabiltzen dira beti, eta behar bada sarrerako parametroak jartzen dira
- Giltzak erabiltzen dira
- Instrukzio bakoitzaren amaieran

Eta gero funtzioa erabiltzeko:

```
<SCRIPT type="text/javascript" >  
    miFuncion();  
</SCRIPT>
```

Erronka 1. Javascript-sarrera

2. Berrikuspena:

f) Funtzioak

i. Funtzio globalak

Javascriptek funtzio eta propietate globalak ditut. Asko erabiliko ditugun batzuk:

- `parseInt()`
string bat analizatu eta integerrera pasatu
- `parseFloat()`
string bat analizatu eta dezimalak dituen zenbakira pasatu
- `Number()`
zenbakira pasatu
- `String()`
Karakterren kataira pasatu
- `isNaN()`
is Not a Number

Zerrenda oso bat: https://www.w3schools.com/jsref/jsref_obj_global.asp

Erronka 1. Javascript-sarrera

2. Berrikuspena:

g) Objektuak

i. Aurrez definitutako objektuak

Objektu bakoitzak metodo asko ditu:

- String (toUpperCase(),)
https://www.w3schools.com/js/js_string_methods.asp
- Date (getMonth(),)
https://www.w3schools.com/js/js_dates.asp
- Array (hurrengo diapositiban)
- Math (round(),.....)
https://www.w3schools.com/js/js_math.asp

ii. Norberak definitzen dituen objektuak, propietateekin eta metodoekin

```
• var person = {  
  firstName: "John",  
  lastName : "Doe",  
  id       : 5566,  
  fullName : function() {  
    return this.firstName + " " + this.lastName;  
  }  
};
```

Erronka 1. Javascript-sarrera

2. Berrikuspena:

g) Objektuak

ii. Norberak definitzen dituen objektuak, propietateekin, metodoekin eta eraikitzailearekin

```
• function Person(first, last, age, eyecolor) {  
  this.firstName = first;  
  this.lastName = last;  
  this.age = age;  
  this.eyeColor = eyecolor;  
}
```

```
var myFather = new Person("John", "Doe", 50, "blue");  
var myMother = new Person("Sally", "Rally", 48, "green");
```


Erronka 1. Javascript-sarrera

2. Berrikuspena:

g) Objektuak

iii. Window objektua

- Window objektua oso garrantzitsua da, zabaldu den leihoa adierazten du. Propietate eta metodo asko ditu. Asko erabiliko ditugun batzuk:

- Prompt()
- Alert()
- SetInterval()
- Document

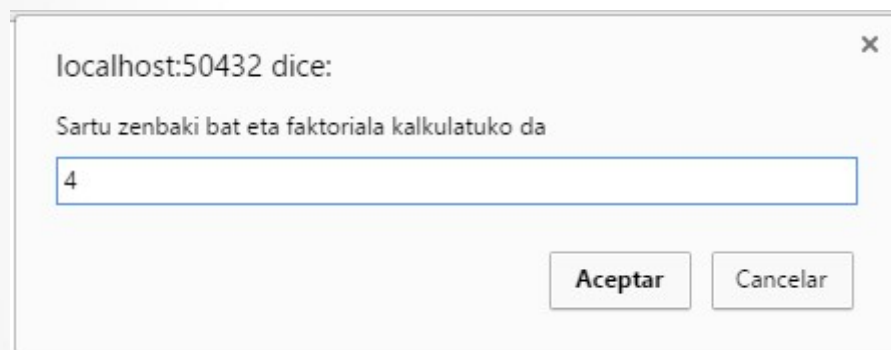
- Zerrenda osoa: https://www.w3schools.com/jsref/obj_window.asp

Erronka 1. Javascript-sarrera

2. Berrikuspena - Ariketak

3. Zenbaki bat eskatu eta bere faktoriala kalkulatu eta erakutsi. Zenbakia bakarrik sartzen dela

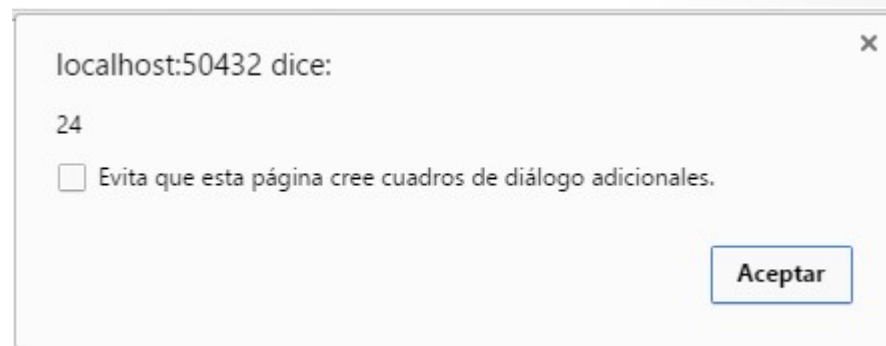
bermatu (isNaN funtzioa erabili) eta hori gertatu arte ez utzi aurrera jarraitzen.



localhost:50432 dice:

Sartu zenbaki bat eta faktoriala kalkulatu da

Aceptar Cancelar



localhost:50432 dice:

24

☐ Evita que esta página cree cuadros de diálogo adicionales.

Aceptar

Erronka 1. Javascript-sarrera

2. Berrikuspena:

h) Array

Balio asko gordetzeko balio duten aldagaiak. Array berria sortzeko ez da ezer berezirik egin behar:

```
var cars = ["Saab", "Volvo", "BMW"];
```

Array bateko balioak eskuratzeko edota balio bat aldatzeko, posizioa erabil daiteke:

```
var name = cars[0];  
cars[0] = "Opel";
```

Array bateko luzera jakiteko: `cars.length` eta array-a ordenatzeko: `cars.sort()`;

Array-en balio guztiak ikusteko bi modu:

```
var fruits, text, fLen, i;  
  
fruits = ["Banana", "Orange", "Apple", "Mango"];  
text = "<ul>";  
  
fLen = fruits.length;  
for (i = 0; i < fLen; i++) {  
    text += "<li>" + fruits[i] + "</li>";  
}
```

```
var fruits, text;  
  
fruits = ["Banana", "Orange", "Apple", "Mango"];  
text = "<ul>";  
  
fruits.forEach(myFunction);  
function myFunction(value) {  
    text += "<li>" + value + "</li>";  
}
```

Erronka 1. Javascript-sarrera

2. Berrikuspena:

h) Array

Balio berri bat sartzeko array batean:

```
fruits.push("Lemon");
```

Array-ak eta Objektuak ez dira nahastu behar. Array-ak objektu mota berezia dira.

Array batean edozein motako aldagai, objektu, array-ak eta abar sartu daitezke

Array baten objektuak sartzen:

```
var person1 = {firstName:"John", lastName:"Doe", age:46};  
var person2 = {firstName:"Jane", lastName:"Doe", age:18};
```

```
Var manyPersons = new Array();
```

```
manyPersons.push(person1);
```

```
manyPersons.push(person2);
```

Informazio gehiago: https://www.w3schools.com/js/js_arrays.asp

Erronka 1. Javascript-sarrera

2. Berrikuspena:

- i) Erroreak
 - Try-Catch-Throw-Finally egitura erabil daiteke erroreak kontrolatzeko

https://www.w3schools.com/js/js_errors.asp

```
try {  
    if(x == "") throw "hutsik dago";  
    if(isNaN(x)) throw "ez da zenbakia";  
    x = Number(x);  
    if(x > 10) throw "handiegia da";  
    if(x < 5) throw "txikiegia da";  
}  
catch(err) {  
    alert("Errorea: " + err + ".");  
}  
finally {  
    alert("Hurrengo zenbakia sartu:");  
}
```

Erronka 1. Javascript-sarrera

2. Berrikuspena - Ariketak

4. Gorde array batean liburuak, irakurrita eta irakurri barik dituzunak. Liburu bakoitzaren datuak: titulua, autoreia, irakurrita (boolean). Sartu datu batzuk aprobak egiteko. Irakurri array osoa eta irakurrita bezala badago, mezu hau ikusiko da “Liburu hau irakurrita daukazu” + titulua + autoreia. Ez badago irakurrita, “Liburu hau irakurri behar duzu” + titulua + autoreia.

Erronka 1. Javascript-sarrera

Estekak

- Visual Studio Code:

<https://code.visualstudio.com/docs/getstarted/userinterface>

- Espresio erregularrak ikasteko eta testeatzeko:

<https://lenguajejs.com/javascript/caracteristicas/expresiones-regulares/>

<https://regexr.com/>

<https://regex101.com/>

- Try-Catch-Throw-Finally egitura:

https://www.w3schools.com/js/js_errors.asp