

Desarrollo Web FullStack

Frontend *Node, NPM y React*

Páginas Web

VS

Aplicaciones Web

Página Web


- HTML
- CSS
- Plantillas
 - PHP
 - ...

[More Tutorials](#)
[DHTML School: DHTML HOME](#)
[DHTML Introduction](#)
[DHTML CSS](#)
[DHTML DOM](#)
[DHTML Event Handlers](#)
[DHTML Examples](#)

The DOM:
[DOM window](#)
[DOM navigator](#)
[DOM event](#)
[DOM collection](#)
[DOM document](#)
[DOM form](#)
[DOM input](#)
[DOM select](#)
[DOM option](#)
[DOM textarea](#)
[DOM table](#)
[DOM tablerow](#)
[DOM tablecell](#)
[DOM anchor](#)
[DOM image](#)
[DOM frameset](#)
[DOM frame](#)
[DOM iframe](#)

Resources:
[DHTML Tutorials](#)
[DHTML Resources](#)
[DHTML Links](#)

Their Speed



Welcome to DHTML School

[< Previous](#) [Next >](#)

After you have studied **DHTML School** you will know what DHTML is.

You will know how **Dynamic HTML** combines HTML, CSS, and JavaScript to create dynamic and interactive Web pages.

Most of the working examples requires Microsoft Internet Explorer 4.0 or higher


Table of contents

[Introduction to DHTML](#)
What DHTML is, and what DHTML is not. The importance of Cascading Style Sheets (CSS) and the Document Object Model (DOM). Browser incompatibilities.

[Cascading Style Sheets](#)
This chapter will teach you the importance of CSS when you are making dynamic documents.

[Document Object Model](#)
This chapter will teach you what the DOM is, and how to use it to make dynamic documents.

W3Schools
[The Best Web Building Site on the Net](#)
[100 MILLION hits in one year](#)
[Please link to us](#)
[Add to Favorites](#)
[Test a friend](#)
[About W3Schools](#)

[More Tutorials](#)
[Developer News](#)
[Browser News](#)

[Google Search](#)

Aplicación Web

- **HTML**
- **CSS**
- **JavaScript**
- **API (Backend)**
 - Python
 - Node
 - Java
 - PHP
 - ...



Figma

CincoDías**EL PAÍS ECONOMÍA**

Compañías Mercados Economía Mi Dinero Fortuna / Cotizaciones f t in

Compañías

TECNOLOGÍA

Adobe compra la plataforma de diseño colaborativo Figma por 20.000 millones



MARIMAR JIMÉNEZ

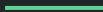
- La adquisición, la mayor de la compañía, forma parte del plan del gigante del software para reforzar su oferta de herramientas creativas. Figma fue valorada hace un año en 10.000 millones de dólares

Presentación del taller

Daniel Mateos Carballares

Desarrollador FullStack
en ***Kashin***

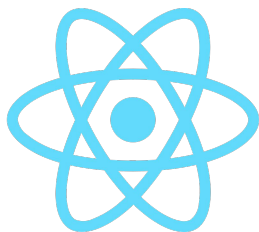
<https://danim47c.dev/>



Taller

I Frontend

Node, NPM y React



II Integración con Backend

Node, NPM, MongoDB y Fastify



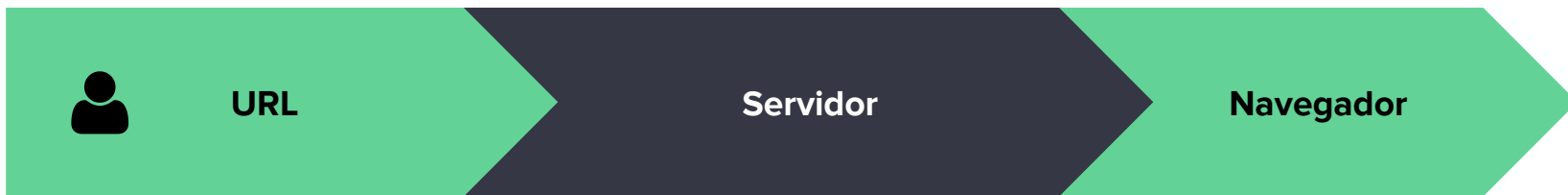
¿Cómo funciona el navegador?



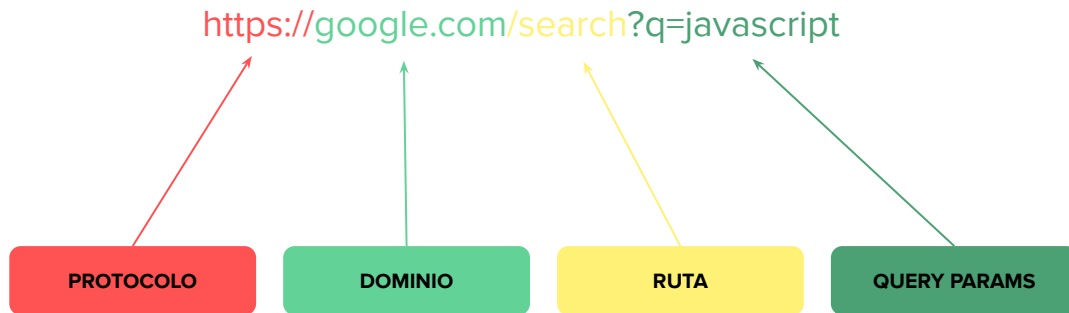
URL

Servidor

Navegador



URL





URL

Servidor

Navegador

Servidor



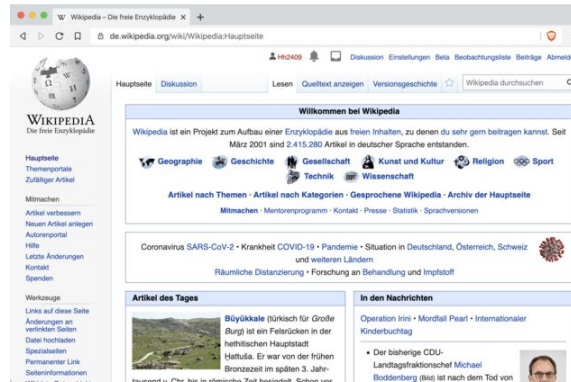


URL

Servidor

Navegador

Navegador

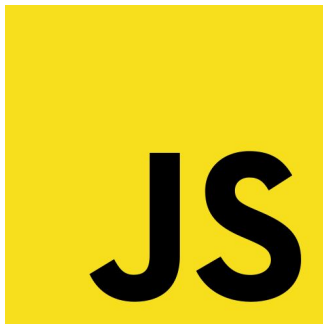


Frontend *React.js*

JavaScript

JavaScript

Interpretado, orientado a objetos y débilmente tipado.



Node.js

Sobre Chrome V8 Engine.



Ejemplo



```
function welcome(name) {  
  return `Hello ${name}!!`  
}  
  
const name = 'John'  
const message = welcome(name)  
  
console.log(message)
```

Variables



```
let variable = "Hello World"
variable = "Hello World 2" // Se puede reasignar

const constant = "Hello World"
constant = "Hello World 2" // Error

const obj = { name: "John" }
obj.name = "Jane" // Se puede modificar
```

Tipos de datos



```
const undef = undefined

const none = null

const bool = true

const num = 1

const str = "string" // o 'string' o `string`

const arr = [1, 2, 3]

const obj = { a: 1, b: 2, c: 3 }

const promise = new Promise()

const date = new Date()
```

Condicionales



```
for (let i = 0; i < 5; i++) {  
  }  
  
while (cond) {  
  }  
  
do {  
  } while (cond)
```

Funciones



```
function func(param1, param2) {  
  return param1 + param2  
}
```

```
const func2 = (param1, param2) => param1 + param2
```

Objetos



```
const obj = { a: 1, b: 2, c: 3 }
```

```
obj.a // 1
```

```
obj.d // undefined
```

```
const extendedObj = { ...obj, d: 4 }
```

Arrays



```
const array = [1, 2, 3, 4, 5]

const mapped =
  array.map((item) => item * 2) // [2, 4, 6, 8, 10]

const filtered =
  array.filter((item) => item % 2 === 0) // [2, 4]


const merge = [...array, 6] // [1, 2, 3, 4, 5, 6]
```


Funciones asíncronas



```
async function func(param1, param2) {  
  return param1 + param2  
}  
  
const func2 =  
  async (param1, param2) => param1 + param2
```

Promesas



```
const prom = async (param1) => {  
  const result = await promise // Otra promesa  
  
  return result  
}  
  
prom() // Promesa  
  .then(result => {  
    console.log(result) // Resultado  
  })
```

Módulos



// ES Modules

```
import { something } from './module'
```

```
export const something = "something"
```

// CommonJS

```
const { something } = require('./module')
```

```
module.exports = {  
  something: "something"  
}
```

React.js

JSX



```
<div>  
  This is JSX  
  
  <h1>Title</h1>  
  
  <p>Paragraph</p>  
  
  <button>Button</button>  
</div>
```

Templates *Plantillas*



```
<div>
  {string.toUpperCase()}

  {
    // This is JS
  }
</div>
```

Props *Parámetros*



```
<button  
  type="submit"  
  onClick={() => console.log("Clicked!")}  
>  
  Click  
</button>
```

Componentes



```
const Component = (props) => {  
  return (  
    <div>  
      <h1>{props.title}</h1>  
  
      {props.children}  
    </div>  
  )  
}
```


Componentes



```
const OtherComponent = () => {  
  return (  
    <Component title="Title">  
      <p>Content</p>  
    </Component>  
  )  
}
```

Hooks




```
useState()  
useEffect()  
useMemo()  
useCallback()  
useReducer()  
useContext()
```

useState



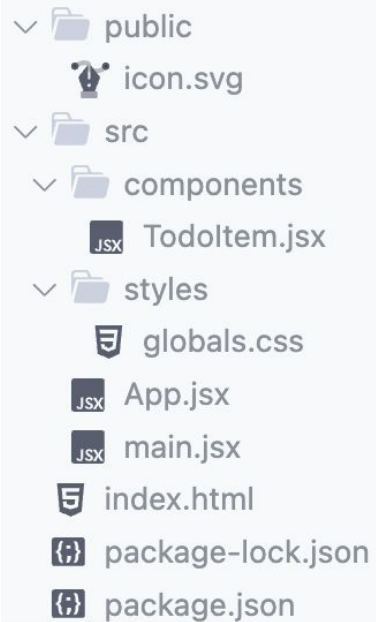
```
const Component = () => {  
  const [state, setState] = useState(0);  
  
  return <div>  
    {state}  
  
    <button onClick={() => setState(state + 1)}>  
      Increment  
    </button>  
  </div>  
}
```

useEffect



```
const Component = () => {  
  const [state, setState] = useState(0);  
  
  useEffect(() => {  
    setInterval(  
      () => {  
        setState(state => state + 1)  
      },  
      1000  
    )  
  }, [])  
  
  return <div>  
    {state}  
  </div>  
}
```

Vite.js



Backend *Fastify, MongoDB*

¿Qué es un Backend?



¿Qué es una *HTTP API Rest*?

GET	<i>/todos</i>
POST	<i>/todos</i>
GET	<i>/todos/:id</i>
PATCH	<i>/todos/:id</i>
DELETE	<i>/todos/:id</i>

```
{
  "null": null,
  "bool": true,
  "num": 1,
  "string": "Hola",
  "array": [1, 2, 3],
  "object": {
    "key": "value"
  }
}
```

JSON (Javascript Object Notation)

MongoDB

MongoDB

Base de Datos ***documental*** organizada en ***colecciones***.

- No relacional
- No SQL



```
{  
  _id: ObjectId("637b7074e0f727dd6538c147"),  
  content: "Some content",  
  createdAt: Date("2022-11-21T12:35:32.884Z")  
}
```

MongoDB Document



```
const { MongoClient, ObjectId } = require("mongodb")

const client = new MongoClient("<db uri>")

await client.connect()

const collection = client.db().collection("todos")

await collection.find({})

await collection.find({ price: { $gte: 5 } })

await collection.findOne({ _id: new ObjectId("637b7074e0f727dd6538c147") })

await collection.updateMany({}, {$set: { content: "some new content" }})
await collection.updateOne(
  { _id: new ObjectId("637b7074e0f727dd6538c147") },
  {$set: { content: "some new content" }}
)

await collection.deleteMany({})
await collection.updateOne({ _id: new ObjectId("637b7074e0f727dd6538c147") })

await collection.findOneAndUpdate({ ... }, {$set: { ... }})
await collection.findOneAndDelete({ ... })
```

Fastify

Fastify



```
const fastify = require("fastify")

const app = fastify()

app.get("/", (req, rep) => {
  return {
    hello: "world"
  }
})

app.listen({ port: 3000 })
```

Fastify Plugins



```
const fastify = require("fastify")

const app = fastify()

app.register(require("@fastify/cors"), {
  origin: "*"
})

app.register(require("@fastify/mongodb"), {
  url: "<db uri>",
})

app.register(require("./todos")) // custom module

app.listen({ port: 3000 })
```