

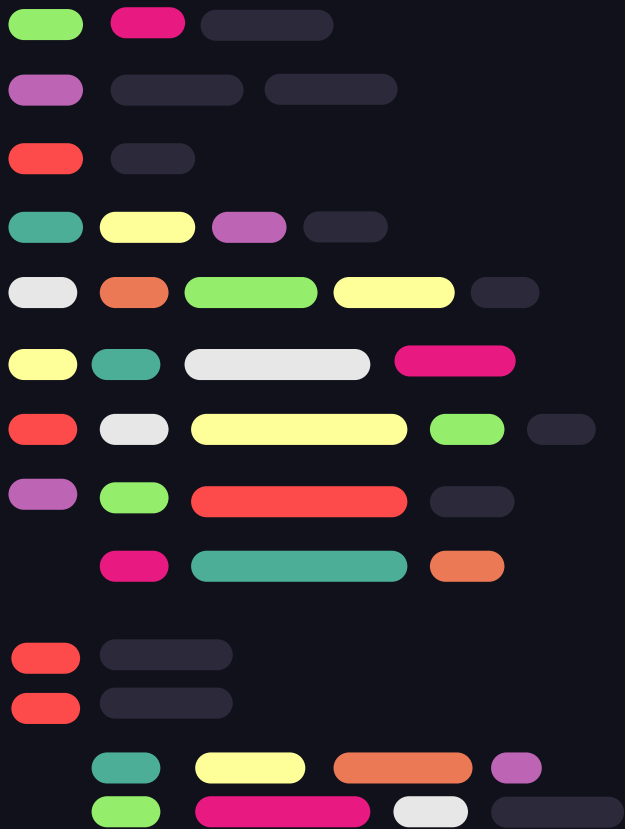


# NODE.JS y JQUERY

< FRONTEND y BACKEND >

Esqueda Jiménez Eduardo  
Zalazar Almanza Mariana





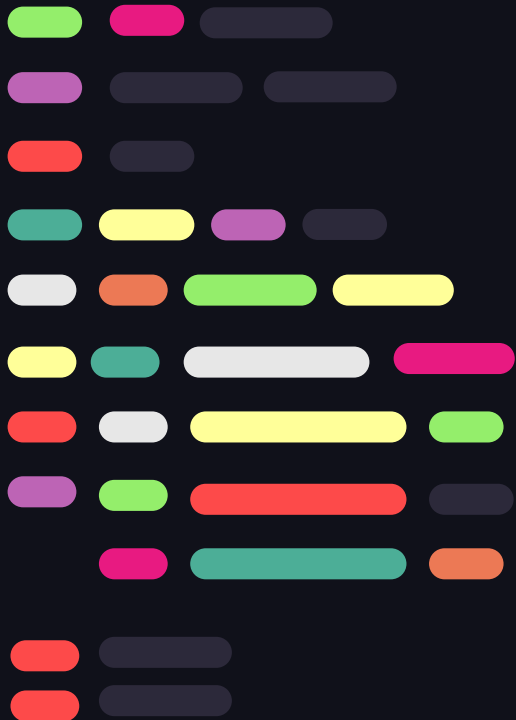
# NODE . JS



# Qué es?

Node.js es un entorno de tiempo de ejecución de JavaScript, Este entorno de tiempo de ejecución en tiempo real incluye todo lo que se necesita para ejecutar un programa escrito en JavaScript





- Tanto JS como Node.JS se ejecutan en el motor de tiempo de ejecución JavaScript V8
- V8 es el nombre del motor de JavaScript que alimenta Google Chrome
- Este motor toma el código JavaScript y lo convierte en un código de máquina más rápido.





# CARACTERÍSTICAS PRINCIPALES



## Velocidad

Node.js está  
construido sobre el  
motor de JavaScript  
V8 de Google Chrome

## Sin Bufer

Las aplicaciones de  
Node.js generan los  
datos en trozo

## Asíncrono

No detiene el  
proceso por  
esperar las señal  
de una API

## Sub Proceso

Responde  
rápidamente sin  
bloquearse



# ¿Para qué nos sirve Node.js

Puedes utilizar Node.js para diferentes tipos de aplicaciones. ● ● ●

Aplicaciones  
en tiempo  
real

Aplicaciones  
basadas en  
JSON:API

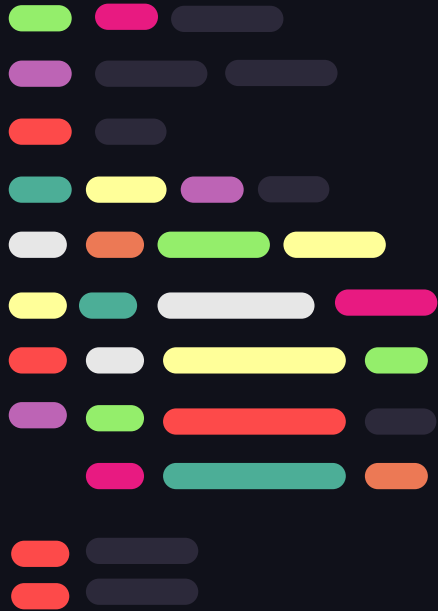
Aplicaciones  
vinculadas a  
E/S

Aplicaciones  
de Streaming

Aplicaciones  
de página  
única



# ¿Quién usa Node.js?



Netflix

General Electric

Paypal

Microsoft

Uber

LinkedIn

GoDaddy

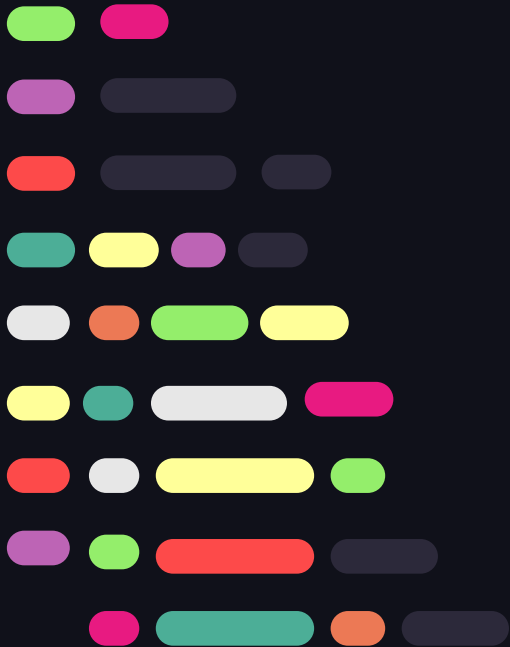
Medium

eBay

NASA



# ¿COMÓ FUNCIONA?



`Node.js` opera en un solo subproceso, utilizando el modelo entrada y entrada sin bloqueo de la salida, lo que le permite soportar decenas de miles de conexiones al mismo tiempo mantenidas en el bucle de eventos.

El nodo está completamente controlado por eventos. Resumiendo podemos decir que el servidor consta de un subproceso que procesa un evento tras otro.





# Librerías estandar



`fs`: Para manipular ficheros

`path`: Para trabajar con directorios.

`http`: Para crear servidores y clientes

`url`: Para analizar URLs y extraer segmentos y variables de ellas.

# Ejemplo de código



Se crea una forma sencilla un servidor web básico para responder cualquier petición simplemente usando el paquete HTTP de *Node*, como se muestra abajo. Este, creará un servidor y escuchará cualquier clase de peticiones en la URL `http://127.0.0.1:8000/`; cuando se reciba una petición, se responderá enviando en texto la respuesta: "Hola Mundo!".



```
// Se carga el módulo de HTTP
```

```
var http = require("http");
```

```
// Creación del servidor HTTP, y se define la escucha
```

```
// de peticiones en el puerto 8000
```

```
http
```

```
.createServer(function (request, response) {
```

```
  // Se define la cabecera HTTP, con el estado HTTP (OK: 200) y el tipo de contenido
```

```
  response.writeHead(200, { "Content-Type": "text/plain" });
```

```
  // Se responde, en el cuerpo de la respuesta con el mensaje "Hello World"
```

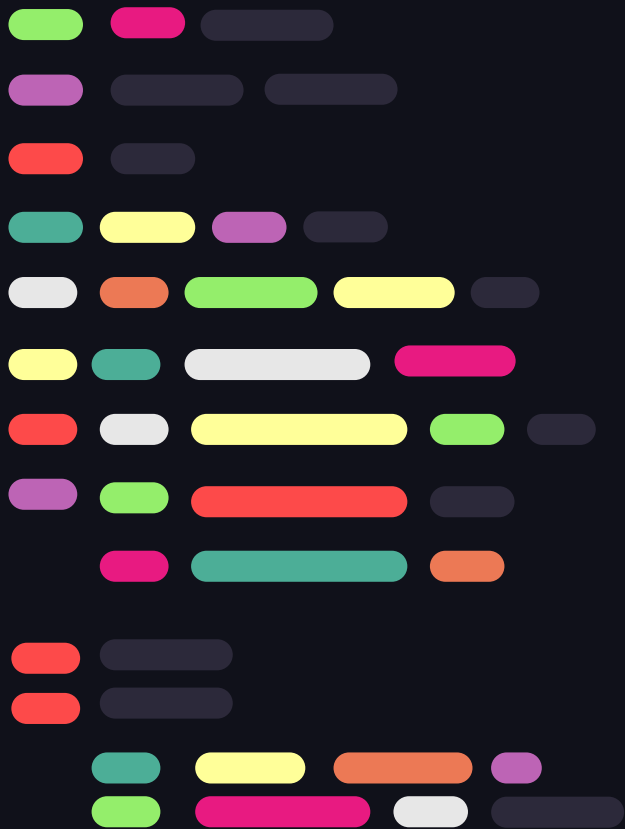
```
  response.end("Hola Mundo!\n");
```

```
})
```

```
.listen(8000);
```

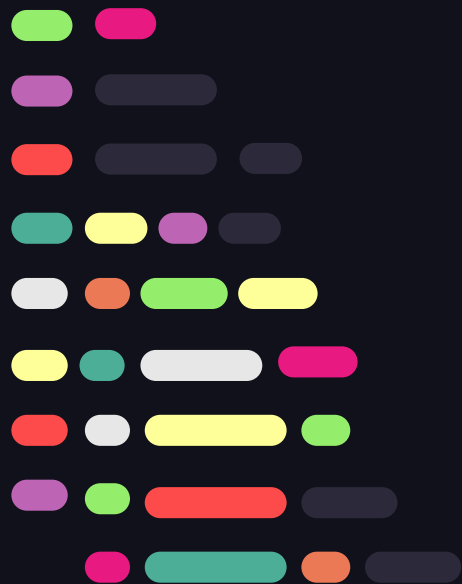
```
// Se escribe la URL para el acceso al servidor
```

```
console.log("Servidor en la url http://127.0.0.1:8000/");
```



# JQUERY

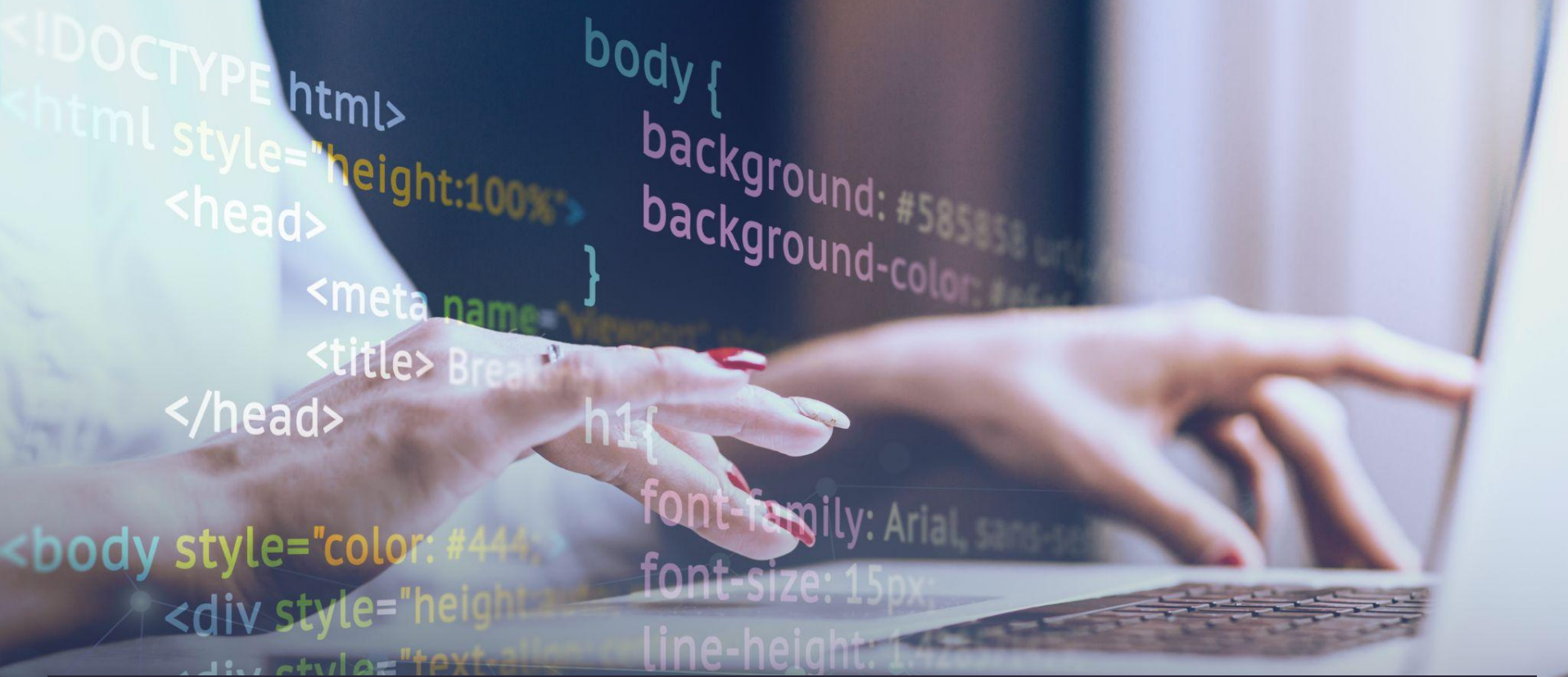




Es una biblioteca de JavaScript que simplifica la manipulación del DOM y la interacción con JavaScript en general.

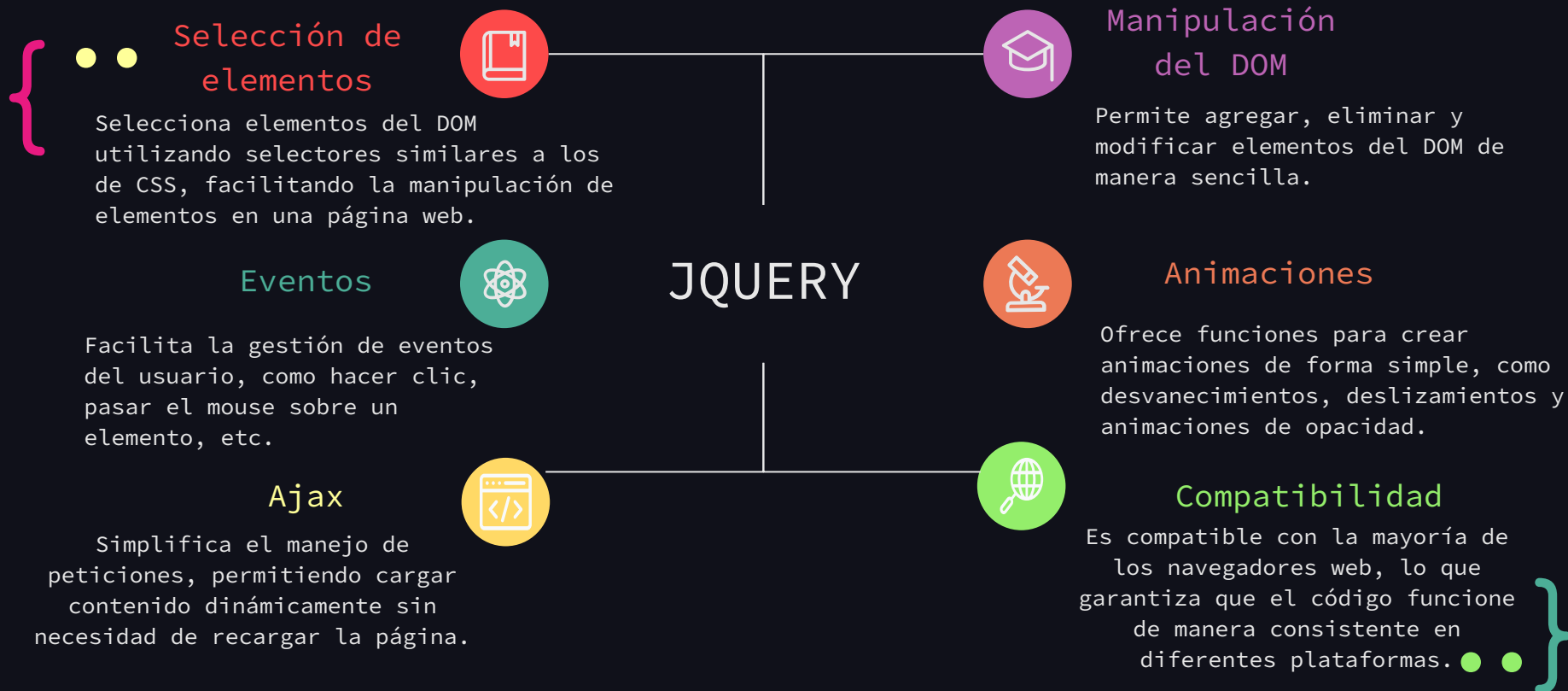
¿Qué es?





jQuery es una de las bibliotecas de JavaScript más populares del mundo.

# Características principales



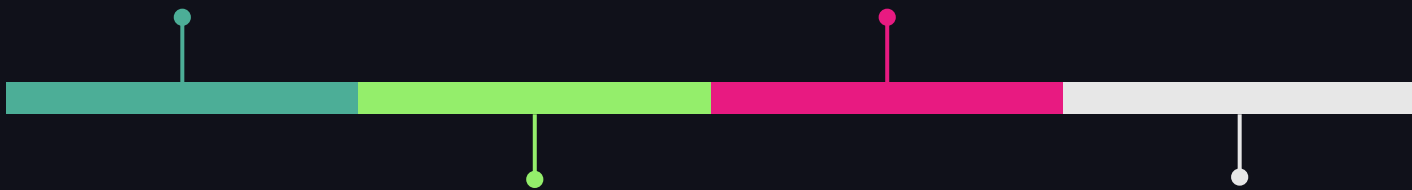


# Ventajas de usar JQuery



**Facilidad de uso:**  
simplifica la  
escritura de  
código JS.

**Gran comunidad:**  
gran comunidad de  
desarrolladores  
que contribuyen.



**Compatibilidad:**  
maneja diferencias  
entre navegadores,  
lo que facilita el  
desarrollo  
multiplataforma.

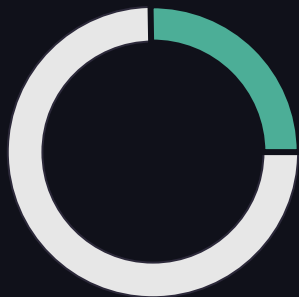
**Rendimiento:** Es  
ligero y ofrece un  
buen rendimiento  
en la manipulación  
del DOM y otras  
tareas.







# Limitaciones



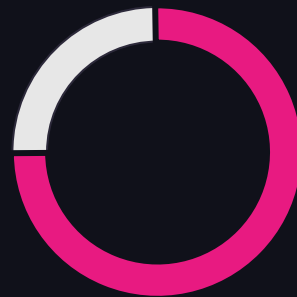
## Peso

jQuery agrega un peso adicional al cargar una página web, lo que puede afectar el rendimiento en dispositivos móviles o conexiones lentas.



## Uso Excesivo

En proyectos grandes, el uso excesivo de jQuery puede hacer que el código sea difícil de mantener y comprender.

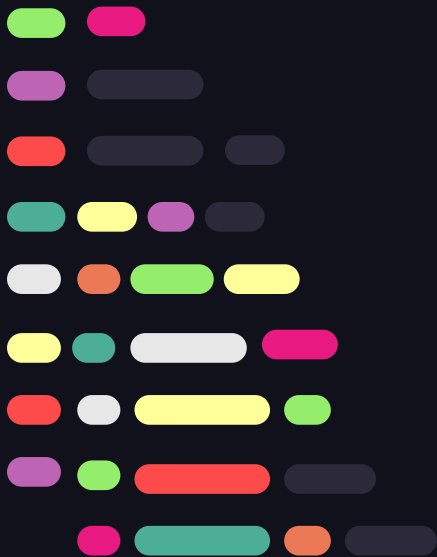


## Dependencia

Algunos desarrolladores prefieren evitar dependencias externas y optan por escribir JavaScript puro en lugar de utilizar jQuery.



# Características Importantes



`hide()`: oculta los elementos HTML, haciendo que dejen de afectar a la página HTML.

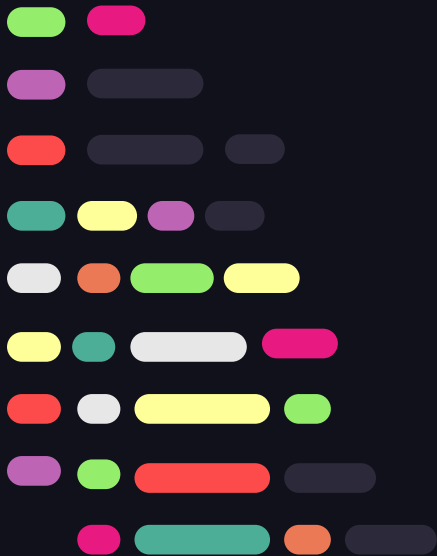
`fadeIn()`: modifica la opacidad de los elementos HTML para que aparezcan gradualmente en la página HTML.

`slideUp()/slideDown()`: oculta/muestra elementos con una animación deslizando.





# Características **Importantes**



`fadeToggle()/toggle()`: modifica la visibilidad de los elementos HTML en función de su propiedad de visualización CSS mediante un evento de clic. Si un elemento es visible, esta función lo ocultará. Si está oculto, ocurrirá lo contrario.



- Selector de inicio es quien inicia la acción.  
En JQuery se pueden usar como selectores elementos HTML como a (enlaces) h1, h2, etc. También clases usando (.nombre) e identificadores (#nombre).
- Evento puede ser click, hover, etc.
- Selector de destino es el que identifica el elemento a modificar.
- Las funciones o métodos son quienes finalmente ejecutan la tarea y los atributos son opcionales en algunos casos.

