25 de junio de 2022

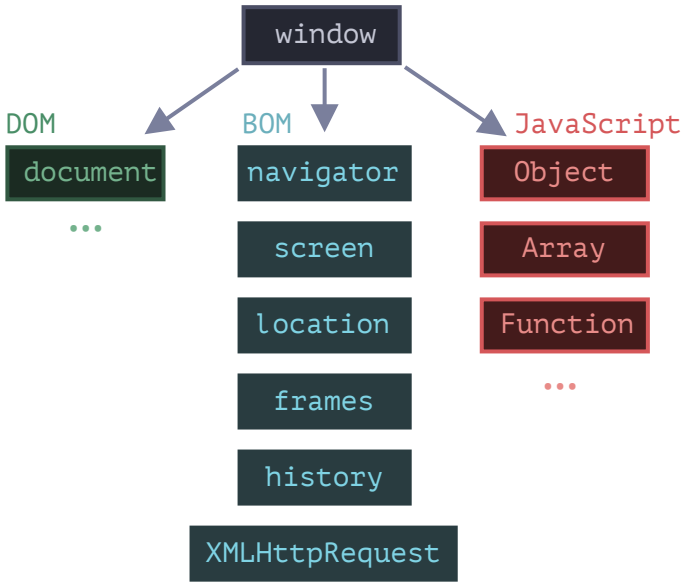
Entorno del navegador, especificaciones

El lenguaje JavaScript fue creado inicialmente para los navegadores web. Desde entonces, ha evolucionado en un lenguaje con muchos usos y plataformas.

Una plataforma puede ser un navegador, un servidor web u otro *host* ("anfitrión"); incluso una máquina de café "inteligente", si puede ejecutar JavaScript. Cada uno de ellos proporciona una funcionalidad específica de la plataforma. La especificación de JavaScript llama a esto *entorno de host*.

Un entorno host proporciona sus propios objetos y funciones adicionales al núcleo del lenguaje. Los navegadores web proporcionan un medio para controlar las páginas web. Node.js proporciona características del lado del servidor, etc.

Aquí tienes una vista general de lo que tenemos cuando JavaScript se ejecuta en un navegador web:



Hay un objeto "raíz" llamado `window`. Tiene dos roles:

1. Primero, es un objeto global para el código JavaScript, como se describe en el capítulo Objeto Global.
2. Segundo, representa la "ventana del navegador" y proporciona métodos para controlarla.

Por ejemplo, podemos usarlo como objeto global:

```
function sayHi() {
  alert("Hola");
}


```

```
// Las funciones globales son métodos del objeto global:
window.sayHi();
```

Y podemos usarlo como una ventana del navegador. Para ver la altura de la ventana:

```
alert(window.innerHeight); // altura interior de la ventana
```

Hay más métodos y propiedades específicos de `window`, los que cubriremos más adelante.

DOM (Modelo de Objetos del Documento)

Document Object Model, o DOM, representa todo el contenido de la página como objetos que pueden ser modificados.

El objeto `document` es el punto de entrada a la página. Con él podemos cambiar o crear cualquier cosa en la página.

Por ejemplo:

```
// cambiar el color de fondo a rojo
document.body.style.background = "red";

// deshacer el cambio después de 1 segundo
setTimeout(() => document.body.style.background = "", 1000);
```

Aquí usamos `document.body.style`, pero hay muchos, muchos más. Las propiedades y métodos se describen en la especificación: DOM Living Standard.

> **DOM no es solo para navegadores**
>
> La especificación DOM explica la estructura de un documento y proporciona objetos para manipularlo. Hay instrumentos que no son del navegador que también usan DOM.
>
> Por ejemplo, los scripts del lado del servidor que descargan páginas HTML y las procesan, también pueden usar DOM. Sin embargo, podrían admitir solamente parte de la especificación.

> **CSSOM para los estilos**
>
> También hay una especificación separada, CSS Object Model (CSSOM) para las reglas y hojas de estilo CSS, que explica cómo se representan como objetos y cómo leerlos y escribirlos.
>
> CSSOM se usa junto con DOM cuando modificamos las reglas de estilo para el documento. Sin embargo, en la práctica rara vez se requiere CSSOM, porque rara vez necesitamos modificar las reglas CSS desde JavaScript (generalmente solo agregamos y eliminamos clases CSS, no modificamos sus reglas CSS), pero eso también es posible.

BOM (Modelo de Objetos del Navegador)

El Modelo de Objetos del Navegador (Browser Object Model, BOM) son objetos adicionales proporcionados por el navegador (entorno host) para trabajar con todo excepto el documento.

Por ejemplo:

- El objeto navigator proporciona información sobre el navegador y el sistema operativo. Hay muchas propiedades, pero las dos más conocidas son: `navigator.userAgent` : acerca del navegador actual, y `navigator.platform` : acerca de la plataforma (ayuda a distinguir Windows/Linux/Mac, etc.).
- El objeto location nos permite leer la URL actual y puede redirigir el navegador a una nueva.

Aquí vemos cómo podemos usar el objeto `location` :

```
1  alert(location.href); // muestra la URL actual
2  if (confirm("Ir a wikipedia?")) {
3    location.href = "https://wikipedia.org"; // redirigir el navegador a otra URI
4  }
```

Las funciones `alert/confirm/prompt` también forman parte de BOM: no están directamente relacionadas con el documento, sino que representan métodos puros de comunicación del navegador con el usuario.

Especificaciones

BOM es la parte general de la especificación de HTML specification.

Sí, oíste bien. La especificación HTML en https://html.spec.whatwg.org no solo trata sobre el "lenguaje HTML" (etiquetas, atributos), sino que también cubre un montón de objetos, métodos y extensiones DOM específicas del navegador. Eso es "HTML en términos generales". Además, algunas partes tienen especificaciones adicionales listadas en https://spec.whatwg.org.

Resumen

En términos de estándares, tenemos:

La especificación del DOM

Describe la estructura del documento, las manipulaciones y los eventos; consulte https://dom.spec.whatwg.org.

La especificación del CSSOM

Describe las hojas de estilo y las reglas de estilo, las manipulaciones con ellas y su vínculo a los documentos. Consulte https://www.w3.org/TR/cssom-1/.

La especificación del HTML

Describe el lenguaje HTML (por ejemplo, etiquetas), y también el BOM (modelo de objeto del navegador) que describe varias funciones del navegador como `setTimeout` , `alert` , `location` , etc. Esta toma la especificación DOM y la extiende con muchas propiedades y métodos adicionales. Consulta https://html.spec.whatwg.org.

Adicionalmente, algunas clases son descritas separadamente en https://spec.whatwg.org/.

Ten en cuenta los enlaces anteriores, ya que hay tantas cosas que es imposible cubrir y recordar todo.

Cuando desees leer sobre una propiedad o un método, el manual de Mozilla en https://developer.mozilla.org/es/search es un buen recurso, pero leer las especificaciones correspondientes puede ser mejor: es más complejo y hay más para leer, pero hará que su conocimiento de los fundamentos sea sólido y completo.

Para encontrar algo, a menudo es conveniente usar una búsqueda como "WHATWG [término]" o "MDN [término]". Por ejemplo https://google.com?q=whatwg+localstorage, https://google.com?q=mdn+localstorage.

Ahora nos concentraremos en aprender el DOM, porque `document` juega el papel central en la interfaz de usuario.

Lección anterior | Próxima lección

Compartir

Mapa del Tutorial

Comentarios

- Si tiene sugerencias sobre qué mejorar, por favor enviar una propuesta de GitHub o una solicitud de extracción en lugar de comentar.
- Si no puede entender algo en el artículo, por favor explique.
- Para insertar algunas palabras de código, use la etiqueta `<code>`, para varias líneas – envolverlas en la etiqueta `<pre>`, para más de 10 líneas – utilice una entorno controlado (sandbox) (plnkr, jsbin, codepen...)