

Unidad 7: Animación con JavaScript

Principios de JavaScript

Vincular un archivo JavaScript

La forma de vincular un archivo JavaScript en un HTML es la siguiente:

```
<script src="ruta/archivo.js"></script>
```

Se recomienda que esta etiqueta se ponga al final del documento, antes del cierre de la etiqueta body, para no bloquear el renderizado de la página mientras el archivo se carga.

```
var edadUsuario = 20;
```

Las variables se declaran con **var**. Los valores se asignan con un **=**. Los valores numéricos van sin comillas. Las instrucciones se finalizan con un **;** de manera similar a CSS.

```
var nombreUsuario = "Bill Murray";
```

Los valores literales van con comillas.

```
var aceptaCondiciones = true;
```

Los valores booleanos (**true** o **false**) van sin comillas.

```
// Comentarios
```

Los comentarios de una línea se preceden de **//**.

```
/*
```

Los comentarios de bloque funcionan igual que en CSS.

```
*/
```

```
if (edadUsuario > 20) {...}
```

Las instrucciones se agrupan entre llaves. Se pueden condicionar (**if**) a que se dé una situación concreta.

```
alert("Hola");
```

Las funciones son bloques de código reutilizable que podemos ejecutar. Tienen un paréntesis al final, a través del cual podemos pasar parámetros a una función (por ejemplo, el texto a mostrar en un **alert**).

```
document.getElementById("globo");
```

Existen muchas formas de acceder a partes concretas de la página. En este caso, mediante el método **getElementById()**, estamos accediendo al elemento cuyo **id** sea igual a un valor (**id="globo"**).

Al contrario que con otros métodos, en este ya estamos indicando que el valor que pasaremos es un **id**, por lo que no es necesario precederlo de la almohadilla (#). De hecho, en este caso concreto, con # no funcionará. **Presta atención a este detalle** en los próximos ejemplos, porque dependiendo del tipo de función la necesitaremos o no.

En este caso, el punto no significa “clase” (como las de CSS), sino que indica pertenencia. Así, **getElementById()** es un método (función propia) del objeto **document**, que representa toda la página web renderizada.

Incorporar JS en un SVG

<script><![CDATA[...]]></script>

De la misma forma que con CSS, podemos incorporar JavaScript a un SVG (en este caso, al final del mismo, antes del cierre de la etiqueta **</svg>**).

También como en CSS, es habitual englobar el código JavaScript en una etiqueta **CDATA** para minimizar los conflictos entre el JS y el resto del XML.

Un SVG cargado a través de la etiqueta **** o como imagen de fondo podría no ejecutar el JavaScript que tenga incrustado. Es una medida que los navegadores suelen emplear para evitar problemas de seguridad.

Pasos para crear una animación de morphing con GreenSock

Para realizar la animación de morphing se han seguido los siguientes pasos:

1. Definir un SVG con una serie de etiquetas **path**, cada una de ellas identificada con un id, que serán las formas del morphing.
2. Cargar las siguientes librerías de GreenSock en el HTML, a través de la etiqueta **<script>**:
cdnjs.cloudflare.com/ajax/libs/gsap/1.18.0/TweenMax.min.js
s3-us-west-2.amazonaws.com/s.cdpn.io/16327/MorphSVGPlugin.min.js
s3-us-west-2.amazonaws.com/s.cdpn.io/16327/findShapeIndex.js
3. En la CSS, ocultar los path que no sean el primero:
#perro,
#jirafa{visibility: hidden;}

4. En una etiqueta `<script>`, introducir el siguiente código, que localiza el elemento con el id inicial, y luego le asigna las transformaciones de morphing, separadas por un segundo entre ellas:

```
var tl = new TimelineLite(),
    circle = document.getElementById("globo");
tl.to(circle, 1, {morphSVG:"#perro"}, "+=1").to(circle, 1,
{morphSVG:"#jirafa"}, "+=1");
```

Pasos para crear una animación sobre trazo con GreenSock

Para realizar la animación sobre trazo se han seguido los siguientes pasos:

1. Definir un SVG con una etiqueta **path**, identificada con un **id**. Si no quisiéramos que este trazo fuera visible, se podría definir dentro de la etiqueta `<defs>`. Definir también el dibujo a animar (en el ejemplo, un `<circle>`).
2. Cargar las siguientes librerías de GreenSock en el HTML:
cdnjs.cloudflare.com/ajax/libs/gsap/1.19.1/TweenMax.min.js
s3-us-west-2.amazonaws.com/s.cdn.io/16327/MorphSVGPlugin.min.js
3. En una etiqueta `<script>`, escribir el siguiente código, donde se asigna la animación al **id** del elemento a animar, y el **id** del **path** de referencia para la animación:

```
TweenLite.set("#circulo", {xPercent:-50, yPercent:-50});
```

```
TweenMax.to("#circulo", 3, {
  bezier: {
    values: MorphSVGPlugin.pathDataToBezier("#trazo",
{align:"#circulo"}),
    type: "cubic"
  },
  ease: Linear.easeNone,
  repeat: -1
});
```