

The image features a dark navy blue background. On the left side, there are two overlapping geometric shapes: a blue parallelogram and a light green parallelogram, both tilted at an angle. The green shape is positioned slightly to the right and above the blue one. In the bottom left corner, there are several dark gray diagonal stripes.

# Variables en CSS



## Significado

- Crear variables para almacenar valores que usaremos en nuestras hojas de estilo

## Características

- Código más limpio
- Mejor reusabilidad
- Mejor legibilidad
- Acercamiento de las guías de estilos al código



## Variables globales

- Haremos uso de :root, pseudo-clase que selecciona el elemento raíz de nuestro documento html
- Dentro de él definiremos nuestras variables globales
- Representa el nivel más alto del DOM

```
:root{
```

```
    //aquí declaramos nuestras variables globales
```

```
}
```



## Variables locales

- Al principio del selector sobre el que queremos actuar

```
h1{
```

```
    //aquí declaramos nuestras variables locales
```

```
}
```



## Declaración y uso de variables

- Usamos los caracteres reservados -- para declarar las variables
- Usamos la función var() para usar las variables

```
:root{  
  --color_primario: green;  
  --fuente_normal: 1.5em;  
}
```

```
h1{  
  --color_primario: blue;  
  
  color: var(--color_primario);  
  font-size: var(--fuente_normal)  
}
```



## Declaración y uso de variables

- Las variables se heredan de la misma forma que las reglas de CSS
- Si queremos hacer operaciones debemos hacerlas con calc cuando definamos las variables

```
p{  
  --color_parrafos: red;  
  --fuente_parrafos: calc(15vw/5);  
}  
  
p.principal{  
  color: var(--color_parrafos);  
  font-size: var(--fuente_parrafos);  
}  
  
p.secundario{  
  color: var(--color_parrafos);  
  font-size: var(--fuente_parrafos);  
}
```



## Declaración y uso de variables

- El valor de las variables se puede modificar dentro de las media queries volviendo a hacer uso del selector que las originó

```
:root {  
  --color_primario: green;  
  --fuente_normal: 1.5em;  
}
```

```
p{  
  --tamanyo_fuente: 2em;  
  
  font-size: var(--tamanyo_fuente);  
}
```

```
@media (max-width:768px) {  
  :root {  
    --fuente_normal: 1em;  
  }  
}
```

```
p{  
  --tamanyo_fuente:1.5em;  
}  
}
```



## Declaración y uso de variables

- Actualmente no es posible utilizar variables de forma nativa con propiedades como max.width
- Se está trabajando en la función env() para generar variables de entorno

```
:root {  
  --tamanyo_normal: 768px;  
}
```

```
@media (max-width:var(--tamanyo_normal)) {  
  body{  
    background-color: red;  
  }  
}
```





## Variables CSS en JS

- Al formar parte del DOM, las variables son accesibles desde JS
- La forma de trabajar es la siguiente:

```
// Función para obtener el valor de una variable
function getEstiloComponente(componente, variable)
{
    // Obtenemos las propiedades y valores del componente
    let estiloComponente = getComputedStyle(document.querySelector(componente));
    console.log(`La variable: ${variable}, tiene el valor de: ${estiloComponente.getPropertyValue(variable)}`);
}
```

```
// Función para modificar el valor de una variable en css
function setEstiloComponente(componente, variable, valor)
{
    let selectorComponente = document.querySelector(componente);
    selectorComponente.style.setProperty(variable, valor);
}
```



## Variables CSS en JS

- Ejemplos del css y llamada de las funciones:

```
:root {  
  --color_primario: green;  
}
```

```
h1{  
  color: var(--color_primario);  
}
```

```
p{  
  --tamanyo_fuente: 2em;  
}
```

```
p.secundario{  
  font-size: var(--tamanyo_fuente);  
}
```

```
getEstiloComponente(":root", "--color_primario");  
setEstiloComponente(":root", "--color_primario", "blue");
```

```
getEstiloComponente("p.secundario", "--tamanyo_fuente");  
setEstiloComponente("p.secundario", "--tamanyo_fuente", "3em");
```