CSS Cheat Sheet

Javier Flores
www.linkedin.com/in/javier-flores-borrego

1. Selección de Elementos:

- `elemento { propiedad: valor; }`: Aplica estilos a un elemento específico.

2. Selectores:

- ID: `#id`
- Clase: `.clase`
- Elemento HTML: `elemento`
- Selector de Atributo: `elemento[atributo='valor']`

3. Propiedades Comunes:

- `color`: Color del texto.
- `font-size`: Tamaño de la fuente.
- `font-family`: Tipo de fuente.
- `background-color`: Color de fondo.
- `border`: Borde del elemento.
- `margin`: Margen exterior.
- `padding`: Relleno interior.
- `width` y `height`: Ancho y alto del elemento.
- `display`: Tipo de visualización (por ejemplo, block, inline, none).

4. Pseudo-clases y Pseudo-elementos:

- `:hover`: Estilo cuando el cursor se sitúa sobre el elemento.
- `:nth-child(n)`: Estilo para el n-ésimo hijo de un elemento.
- `::before` y `::after`: Pseudo-elementos para agregar contenido antes o después del contenido real del elemento.

5. Agrupación y Herencia:

- Agrupa selectores separados por coma para aplicar los mismos estilos.
- Los estilos se heredan de elementos padres a elementos hijos, a menos que se anulen explícitamente.

6. Unidades de Medida:

- `px`: Píxeles.
- `%`: Porcentaje del tamaño del elemento contenedor.
- `em` y `rem`: Relativo al tamaño de la fuente del elemento.

7. Media Queries:

- `@media screen and (max-width: 600px) { /* Estilos */ }`: Aplica estilos específicos según el tamaño de la pantalla.

Tips para QA Automation

1. Utiliza selectores específicos: Al seleccionar elementos del DOM, utiliza selectores CSS específicos y directos para identificar los elementos de forma precisa. Evita selectores demasiado genéricos que puedan cambiar con el tiempo o ser ambiguos.

2. Inspecciona el DOM: Utiliza herramientas de desarrollo del navegador como DevTools para inspeccionar el DOM y encontrar selectores CSS adecuados para los elementos que necesitas interactuar en tus pruebas.

3. Evita el uso de la posición del elemento esto hace que el selector sea muy flaky ante posibles cambios en la aplicación

4. Evita selectores complejos: Aunque los selectores complejos pueden funcionar, tienden a ser más frágiles y difíciles de mantener. Opta por selectores simples y directos siempre que sea posible.

5. Agrupa selectores: Si tienes varios elementos similares que necesitas seleccionar, considera agruparlos utilizando clases o atributos personalizados para facilitar la selección y mantenimiento.

6. Reutiliza selectores: Si encuentras que estás utilizando el mismo selector en múltiples pruebas, considera encapsularlo en una función o variable para reutilizarlo y mantener un código más limpio y modular.

7. Actualiza selectores según sea necesario: A medida que la aplicación evoluciona, es posible que necesites actualizar tus selectores CSS para reflejar cambios en la estructura del DOM. Mantén tus pruebas actualizadas con los cambios correspondientes.

8. Considera XPath: Aunque CSS es comúnmente preferido, en algunas situaciones XPath puede ser más eficaz para seleccionar elementos, especialmente en casos donde los selectores CSS son difíciles de construir.

9. Prueba tus selectores: Antes de ejecutar tus pruebas en un flujo de trabajo de automatización completo, prueba tus selectores individualmente para asegurarte de que estén seleccionando los elementos correctos.

10. Documenta tus selectores: Asegúrate de documentar claramente los selectores que estás utilizando en tus pruebas, junto con una breve descripción de su propósito, para facilitar el mantenimiento futuro y la colaboración con otros miembros del equipo.