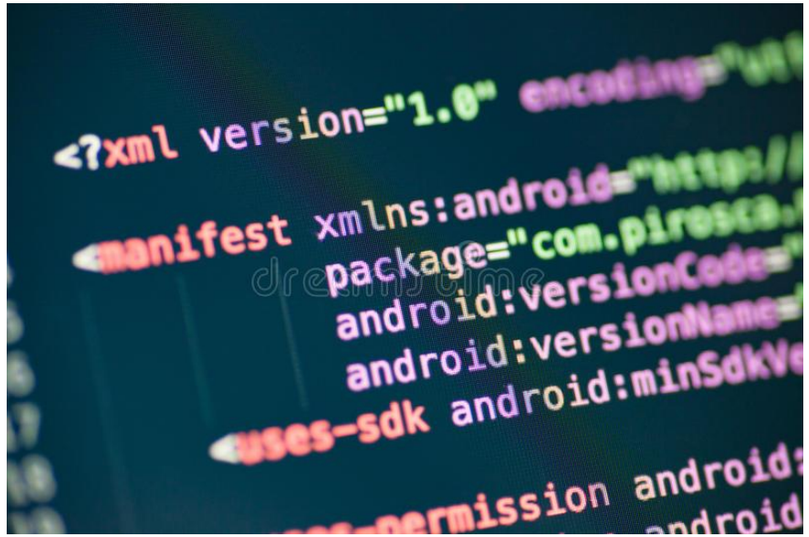
Lenguajes de marcas y sistemas de gestión de la información



UT03 – CSS

6 – Pseudoclases y pseudoelementos

IES Clara del Rey – Madrid

Pseudoclases

Es un tipo de selector que identifica los elementos que están en un estado específico. Podemos dividirlas en:

- De interacción. Relacionadas con las acciones del usuario.
- De ubicación. Relacionadas con enlaces.
- De estructura
- De formularios. Relacionadas con uso y validación de formularios
- De idioma. Relacionadas con el idioma establecido en los elementos.
- De estado. Relacionadas con el estado de la ventana o de elementos modales.

Sólo veremos algunas pseudoclases. Referencia completa en MDN.

Pseudoclases – Sintaxis

Las pseudoclases comienzan con dos puntos (:)

Ejemplo de pseudoclases:

- :hover
- :any-link
- :root

La mayoría pueden añadirse a un selector:

Ejemplos:

- a:hover
- p.clase-de-parrafo span:first-of-type

Pseudoclases – De interacción

Se pueden usar en cualquier elemento, pero es muy habitual usarlas en aquellos con los que interactúa el usuario.

- :hover – El usuario pasa el ratón sobre dicho elemento.
- :active – El usuario se encuentra pulsando dicho elemento.
- :focus – El elemento cuando tiene el foco.
- :focus-within – Uno de los hijos del elemento tiene el foco.
- :focus-visible – El elemento tiene el foco (se cumple :focus) y el elemento debe destacarse (esto lo determina el navegador)

Pseudoclases – De ubicación (enlaces)

Permiten cambiar el estado de enlaces u otros hipervículos (area).

- :any-link – Cualquier elemento que es un enlace (incluye area).
- :link – El enlace no se ha visitado aún.
- :visited – El enlace se ha visitado anteriormente.
- :target – Selecciona un enlace cuya ancla ("id") coincide con el ancla de la página actual (la parte detrás de # en la URL).

Pseudoclases – De estructura

Permiten seleccionar elementos en función de su posición o estructura.

- :root – Aplica estilo al elemento raíz (padre) del documento. Similar a html, pero más específica que html.
- :first-child – Primer elemento hijo (de cualquier tipo).
- :last-child – Último elemento hijo (de cualquier tipo).
- :nth-child(n) – Elemento hijo número n (de cualquier tipo).
- :nth-last-child(n) – Elemento hijo número n empezando desde el final (de cualquier tipo).

Pseudoclases – De estructura

Permiten seleccionar elementos en función de su posición o estructura.

- :first-of-type – Primer elemento hijo (cierto tipo de elemento).
- :last-of-type – Último elemento hijo (cierto tipo de elemento).
- :nth-of-type(n) – Elemento hijo número n (cierto tipo de elemento).
- :nth-last-child(n) – Elemento hijo número n empezando desde el final (cierto tipo de elemento).
- :only-child – Elemento que es hijo único (de cualquier tipo).
- :only-of-type – Elemento que es hijo único (cierto tipo de elemento).
- :empty Elemento vacío (sin hijos, ni texto).

Pseudoclases – De formularios (estado)

Permiten trabajar con los estilos de campos de formularios, en función de su estado.

- :disabled – El campo está deshabilitado.
- :enabled – El campo no está deshabilitado.
- :read-only – El campo es de solo lectura.
- :read-write – El campo es editable, no es read-only.
- :placeholder-shown – Se está mostrando el atributo placeholder.
- :default – El valor del campo tiene el valor por defecto.
- :checked – El campo está marcado. Para radio y checkbox.

Pseudoclases – De formularios (validación)

- :required – El campo es obligatorio (tiene atributo required).
- :optional – El campo es opcional – Por defecto son opcionales todos.
- :valid – El campo es válido, ha pasado la validación.
- :invalid – El campo no es válido, no ha pasado la validación.
- :user-valid – Igual que valid, pero requiere que el usuario haya interactuado con el campo antes de hacer la validación.
- :user-invalid – Igual que user-invalid, pero para campos no válidos.
- :in-range – El valor del campo está entre los valores indicados en los atributos min y max.
- :out-of-range – El valor del campo no está entre min y max.

Pseudoelementos

Permiten aplicar estilos a elementos "virtuales".

Los pseudoelementos no existen como tales en el documento HTML, no están marcados con etiquetas, sino que se "crean" dinámicamente.

Hay dos tipos:

- Pseudoelementos que identifica partes ya existentes del contenido, como primera línea de un texto, primera letra, errores ortográficos, etc.
- Pseudoelementos que permiten añadir contenido antes o después de otro elemento, sin necesidad de usar JavaScript.

Los pseudoelementos en CSS 3 comienzan con "::" para distinguirlos de las pseudoclases, que empiezan con ":". En CSS 2 empezaban igual, y esa sintaxis todavía se admite.

Pseudoelementos

Algunos pseudoelementos son todavía experimentales, y su comportamiento puede cambiar en el futuro.

- ::first-letter – Selecciona la primera letra del texto.
- ::first-line – Selecciona la primera línea del texto.
- ::selection – Aplica los estilos al fragmento de texto seleccionado por el usuario.
- ::target-text – Aplica estilos al fragmento de texto enlazado tras el ancla de una URL (tras el símbolo #)
- ::spelling-error y ::gramar-error – Aplica estilos a un fragmento resaltado por errores ortográficos / gramaticales

Pseudoelementos

- ::marker – Aplica estilos al "bullet" (ul) o número (ol) de cada elemento de una lista.
- ::backdrop – Aplica estilo al fondo que rodea a un elemento, sin que afecte al propio elemento.
- ::placeholder – Aplica estilos al placeholder de un campo input, cuando este es visible (el campo no tiene ningún valor)
- ::file-selector – Aplica estilos al botón que acompaña a un campo de tipo file (<input type="file"). Por motivos relacionados con la seguridad, los navegadores / sistemas operativos generan este botón con estilos muy "rústicos". Este pseudoelemento ayuda a dar estilos a un componente que antes era difícil de maquetar.

Pseudoelementos
Generación de contenido

Hay dos pseudoelementos que permiten añadir contenido al documento desde CSS:

- ::before – Añade contenido antes del selector asociado al pseudoelemento.
- ::after – Igual, pero añade el contenido después.

Ejemplos:

```
p::before { … }
a::after { … }
```

El primero añadiría contenido antes de todos los párrafos.

El segundo añadiría contenido después de cada elemento a.

Pseudoelementos
Generación de contenido

::before y ::after funcionan siempre junto a la propiedad "content".

content establece o reemplaza el contenido seleccionado por un selector. En el caso de ::before y ::after, del pseudoelemento generado.

La propiedad content puede recibir:

- none – No genera contenido.
- normal – Volver a comportamiento por defecto. En el caso de ::before y ::after, el comportamiento es como si usáramos none.
- Un texto – Crea el pseudoelemento con el texto indicado.
- open-quote y close-quote. Genera comillas de cita ("«" o "»").
- no-open-quote y no-close-quote. Elimina las comillas de cita.

Pseudoelementos
Generación de contenido

La propiedad content puede recibir (continuación):

- attr(nombre-del-atributo). Extrae el valor del atributo indicado y lo usa como contenido. Ejemplo:

```
a::after { content: attr(title); }
```

- Un degradado (linear-gradient, radial-gradient, etc.).
- url(url-de-elemento-multimedia). Inserta un elemento multimedia en el pseudoelemento. Normalmente se utiliza con imágenes, pero puede usarse con otros medios.
- counter(nombre de contador) – Inserta un contador que se va incrementando en cada ocasión que se cumple el selector. Los contadores CSS se definen/ personalizan con counter-reset, counter-increment y counter-set.