

Pseudoclases

Módulo 03

Pseudoclasas uso

Si nos preguntamos qué **es una pseudo-clase**, básicamente, una pseudo clase permite hablar sobre un estado en particular de un elemento, vamos a ver un ejemplo en la imagen debajo,

```
/*afecto al vínculo en todos sus estados ya que estoy trabajando con el selector genérico*/  
a{ background: #ccc; display: inline-block; padding: 5px; border: 1px dashed #333;}
```

```
/* link no visitado */  
a:link {  
    color: #333  
}
```

) 1ª Pseudo clase link nos permite modificar el estilo de por ejemplo un "link"

```
/* link visitado */  
a:visited {  
    color: pink;  
}
```

) visited es lo mismo, pero aquí visited referenciaré el estado cuando ese link fue visitado.

Pseudoclasas uso

vinculos δ^0 link
 δ^0 visited

Vamos a entender que :link y :visited son pseudo clases privativas de los vínculos, es decir que solamente podemos utilizarlas en vínculos por ejemplo,

```
/* cuando paso el mouse por arriba de un vinculo */  
a:hover {  
    color: yellow;  
}  
  
/* cuando hago click o elijo en un sistema de tabs un vinculo */  
a:active {  
    color: orange;  
}
```

(Cúando lo estoy presionando)

Pseudoclasses

Podemos usarlas combinadas con ID para afectar sólo a un vínculo,

```
<style>
#vinculo:link { color: red; text-decoration: none; }
</style>
```

Pseudoclasas

Por otro lado :hover y :active pueden aplicarse a otros elementos tanto de línea, **bloque o línea-bloque**,
veamos el siguiente caso,

¿cambio
de color
según si lo
presiono o le pongo
el mouse encima?

```
<style>
p{ background: red; color: white;}
p:hover { background: blue;}
p:active {background:orange;}

</style>
```

focus:

También tenemos una pseudo clase muy utilizada en los elementos de formulario.

Esta pseudo clase hace que cuando hacemos foco en elemento de formulario (campo de texto, de email, textarea, menú, etc) cambie sus características si así lo deseamos.

cuando está
enfocado
se incrementa. (

```
<style>

input { width: 200px;}
input:focus { width: 400px;}

</style>
```

:first-child

Esta pseudoclase referencia al primer hijo no importa el tipo de elemento de un elemento padre. Por ejemplo si sólo en mi CSS escribo **:first-child**, esto afectará a todos los que sean primeros hijos de un padre (cada nuevo parent o padre hay un nuevo hijo), por ejemplo,

```
<style>
p:first-child { background: orange;}
</style>
</head>
<body>

<div>
<p> primer hijo y es párrafo </p>
<p> Segundo párrafo no es un primer hijo sino que es el segundo </p>
</div>

<div>
<p> primer hijo, es párrafo y es un primer hijo porque le div que lo contiene determina un nuevo padre </p>
<p> segundo hijo</p>
</div>
```

:last-child

Como contrapartida de **:first-child** nos encontramos con **:last-child**, que por el contrario hace referencia al último hijo, sin importar el tipo de elemento , de un padre

```
<style>

p:last-child { color: blue; }

</style>
```


PseudoElementos

:first-line

Afecta a la primer línea de un contenido de texto. Es importante saber que no todas las propiedades son aplicables a través de este pseudo elemento, sólo las siguientes:

- propiedades de fuente
- color
- propiedades de fondo
- word-spacing
- letter-spacing
- text-decoration
- vertical-align
- text-transform
- line-height
- clear

```
<style>

p::first-line {
    font-style: italic;
}

</style>
```

:first-letter

Afecta a la primer letra de un contenido de texto. Es importante saber que no todas las propiedades son aplicables a través de este pseudo elemento, sólo las siguientes:

- propiedades de fuente
- color
- propiedades de fondo
- word-spacing
- letter-spacing
- text-decoration
- vertical-align
- text-transform
- line-height
- clear

```
<style>
p::first-letter
{
    color: red; }
</style>
```

Revisión

- Repase todos las **pseudoclases**
- Trabaje con combinando diferentes tipos de selectores
- Aplique **pseudoelementos**
- Trabaje con estos nuevos elementos en el **proyecto Integrador**
- Realice las preguntas necesarias al/la docente antes de continuar



¡Muchas gracias!

¡Sigamos con el siguiente módulo!