

# CSS Pseudo Classes

UTN-FRA

# Pseudoclases

- ▶ Una pseudoclase CSS es una palabra clave que se añade a los selectores y que especifica un estado especial del elemento seleccionado.
- ▶ Algunos ejemplos son: link, visited, active, lang(), first-child, hover, focus, enabled, disabled, checked, indeterminate, target, root, nth-child(), nth-last-of-child(), nth-of-type(), nth-last-of-type(), last-child, first-of-type, last-of-type, only-child, only-of-type, empty, not(), default, valid, invalid, in-range, out-of-range, required, optional, read-only, y read-write.

# Sintaxis

- ▶ El esquema general de sintaxis de CSS es el siguiente:

```
selector #id .clase :pseudoclase ::pseudoelemento [atributo] {  
    propiedad : valor ;  
    propiedad : valor  
}
```

# Enlaces

- ▶ Existen algunas pseudoclasas orientadas a los enlaces o hipervínculos. En este caso, permiten cambiar los estilos dependiendo del comportamiento del enlace:



Pseudoclase	Descripción
:link	Aplica estilos cuando el enlace no ha sido visitado todavía.
:visited	Aplica estilos cuando el enlace ha sido visitado anteriormente.

```
a:link {  
  color: green;  
  font-weight: bold  
}
```

# Mouse

- ▶ Aunque en un principio, las siguientes pseudoclases se utilizaban solamente en enlaces (navegadores como Internet Explorer no los interpretaban con otros elementos), actualmente pueden ser utilizadas con muchos otros elementos y no solamente con enlaces <a>.



Pseudoclase	Descripción
:hover	Aplica estilos cuando pasamos el ratón sobre un elemento.
:active	Aplica estilos cuando estamos pulsando sobre el elemento.

```
a:hover {  
  background-color: cyan;  
  padding: 2px  
}
```

# Interacción del usuario

- ▶ Existen pseudoclasas orientadas principalmente a los campos de formulario de páginas webs y la interacción del usuario con ellos, veamos otro par interesante:

Pseudoclase	Descripción
:focus	Aplica estilos cuando el elemento tiene el foco.
:checked	Aplica estilos cuando la casilla está seleccionada.

```
input:focus {  
  border: 2px dotted #444  
}
```



# Campos activados/desactivados

- ▶ Por norma general, los elementos de un formulario HTML están siempre activados, aunque se pueden desactivar añadiendo el atributo disabled al elemento HTML en cuestión. Esto es una práctica muy utilizada para impedir al usuario escribir en cierta parte de un formulario porque, por ejemplo, no es aplicable.

Pseudoclase	Descripción
:enabled	Aplica estilos cuando el campo del formulario está activado.
:disabled	Aplica estilos cuando el campo del formulario está desactivado.

```
input:enabled {  
  color: black;  
}
```

# Elementos de solo lectura/escritura

- ▶ Los elementos HTML de un formulario suelen ser modificables por el usuario (campos de texto, casillas de verificación, listas de selección, etc...), sin embargo, dichos campos con atributos como disabled o readonly dejan de ser modificables. En ambos casos, los campos pasan a ser campos que no se pueden modificar. Existen varias pseudoclases para modificar sus estilos dependiendo de su naturaleza:

Pseudoclase	Descripción
:read-only	Aplica estilos cuando el campo es de sólo lectura.
:read-write	Aplica estilos cuando el campo es editable por el usuario.

```
input:read-only {  
  background-color: darkred;  
  color: white  
}
```



# Validaciones HTML5

- ▶ En HTML5 es posible dotar de capacidades de validación a los campos de un formulario, pudiendo interactuar desde Javascript o incluso desde CSS. Con estas validaciones podemos asegurarnos de que el usuario escribe en un campo de un formulario el valor esperado que debería. Existen algunas pseudoclases útiles para las validaciones, como por ejemplo las siguientes:

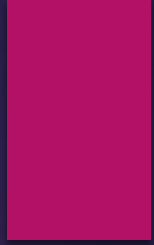
Pseudoclase	Descripción
:required	Aplica estilos a los campos del formulario que son obligatorios.
:optional	Aplica estilos a los campos del formulario que son opcionales.
:invalid	Aplica estilos a los campos que no cumplen la validación HTML5.
:valid	Aplica estilos a los campos que cumplen la validación HTML5.
:out-of-range	Aplica estilos a los campos numéricos que están fuera del rango.
:in-range	Aplica estilos a los campos numéricos que están dentro del rango.

# Negacion

- ▶ Existe una pseudoclase muy útil, denominada pseudoclase de negación. Permite seleccionar todos los elementos que no cumplan los selectores indicados entre paréntesis.
- ▶ El siguiente fragmento de código nos indica que todos los párrafos (elementos <p>) que no pertenezcan a una clase general, se les aplique el estilo especificado.

```
p:not(.general) {  
  border: 1px solid #DDD;  
  padding: 8px;  
  background: #FFF;  
}
```

# Child



Pseudoclase	Descripción
:first-child	Selecciona el elemento que es primer hijo del padre.
:last-child	Selecciona el elemento que es el ultimo hijo del padre.
:nth-child(n)	Selecciona cada elemento que es el n hijo del padre.
:nth-last-child(n)	Selecciona cada elemento que es el n hijo de su padre, contando desde el último hijo
:only-child	Selecciona cada elemento que es el único hijo de su padre

# Type



Pseudoclase	Descripción
:first-of-type	Selecciona cada elemento <p> que es el primer elemento <p> de su padre
:last-of-type	Selecciona cada elemento que es el último del tipo de su padre
:nth-last-of-type(n)	Selecciona cada elemento que es el segundo elemento de su padre, contando desde el último hijo.
:nth-of-type(n)	Selecciona cada elemento que es el segundo elemento del tipo de su padre
:only-of-type	Selecciona cada elemento que es el único elemento del tipo de su padre

# Otras pseudoclases

- ▶ Para finalizar el apartado de pseudoclases, quiero mencionar algunas que no encajan en los apartados anteriores, pero que pueden ser muy útiles en algunos casos:

Pseudoclase	Descripción
:lang(es)	Aplica estilo a los elementos con el atributo lang="es".
:target	Aplica estilo al elemento que coincida con el ancla de la URL.
:root	Aplica estilo al elemento raíz (padre) del documento.
:default	Experimental. Aplica estilo al elemento por defecto.
:dir(A)	Experimental. Aplica al elemento que coincide con la dirección ltr o rtl.
:indeterminate	Experimental. Aplica estilo al elemento <progress> sin definir.
:fullscreen	Experimental. Aplica estilo si la página está en fullscreen.
:scope	Experimental. Aplica estilo a los elementos en el ámbito indicado.
:any(A)	Experimental. Aplica estilo si coincide algún elemento indicado en A.



# Pseudo Elementos

- ▶ Al igual que las pseudo-classes, los pseudo-elementos se añaden a los selectores, pero en cambio, no describen un estado especial sino que, permiten añadir estilos a una parte concreta del documento.
- ▶ La lista de pseudo elementos existentes es la siguiente: after, before, first-letter, first-line, placeholder, marker, selection, inactive-selection, spelling-error y grammar-error.

# Sintaxis

- ▶ Para usar los pseudo elementos se utilizan dos puntos dobles (::) en vez de solo uno (:). Esto forma parte de CSS3 y de un intento para distinguir pseudo-elementos de pseudo-clases.
- ▶ El esquema general de sintaxis de CSS es el siguiente:

```
selector #id .clase :pseudoclase ::pseudoelemento [atributo] {  
    propiedad : valor ;  
    propiedad : valor  
}
```

# ::after

- ▶ Permite agregar una imagen o un elemento al finalizar el texto.
- ▶ Es comunmente usado para añadir contenido cosmético a un elemento con la propiedad **content**.

```
a::after {  
  content: "→";  
}
```

# ::before

- ▶ Nos permite agregar una imagen o elemento al iniciar el texto.
- ▶ Es usado normalmente para añadir contenido estético a un elemento, usando la propiedad **content**.
- ▶ Este elemento se muestra en línea con el texto de forma predeterminada.

```
h1::before {  
  content: url(imagen.png);  
}
```

✓✓ **Cualquier idea para modificar el diseño de nuestra web es valida.**

# ::first-letter

- ▶ Modifica la primer letra de nuestros elementos.
- ▶ No aplica cuando el texto es precedido de otro contenido (como imagenes o tablas).
- ▶ La puntuacion que sigue inmediatamente a la primera letra esta incluida

```
p::first-letter {  
  font-size: 130%;  
}
```



# ::first-line

- ▶ Modifica la primer línea de nuestros párrafos .
- ▶ No aplica cuando el texto es precedido de otro contenido (como imagenes o tablas).
- ▶ Tenga en cuenta que la longitud de la primera línea depende de muchos factores, incluido el ancho del elemento, el ancho del documento y el tamaño de fuente del texto.

```
p::first-line {  
  color: red;  
}
```

# ::placeholder

- ▶ Representa el texto del placeholder que proporciona una pista al usuario.

```
::placeholder {  
  color: blue;  
  font-size: 1.5em;  
}
```

# ::marker

Representa un cuadro de marcado generado automáticamente.  
Las propiedades que se pueden aplicar son todas las de text y color.

```
p.note::marker {  
  content: "Note " counter(note-counter) ":";  
  text-align: left;  
}
```

# Pseudo Elementos Highlight

Los pseudoelementos highlight representan partes de un documento a las que se les ha dado un estado particular y que generalmente tienen un estilo diferente para indicar ese estado al usuario.

Por ejemplo, las partes seleccionadas del documento generalmente se resaltan (se les dan colores alternativos de fondo y primer plano, o un color) para indicar su estado seleccionado. Se definen los siguientes pseudoelementos highlight: selection, inactive-selection, spelling-error y grammar-error.

Generalmente estos elementos, excepto highlight, son usados en contextos de edición.

# ::selection

- ▶ Aplica reglas a una porción de un documento que ha sido destacado (por ejemplo: selección con el mouse o algún otro puntero en un dispositivo) del usuario.
- ▶ Sólo un pequeño subconjunto de propiedades CSS pueden ser usadas en una regla `::selection` en el selector de: `color`, `background`, `background-color` y `text-shadow`.
- ▶
- ▶ Tenga en cuenta que, en particular, `background-image` es ignorado, como cualquier otra propiedad.

```
::-moz-selection { color: gold; background: red; }
```

```
::selection { color: gold; background: red; }
```



