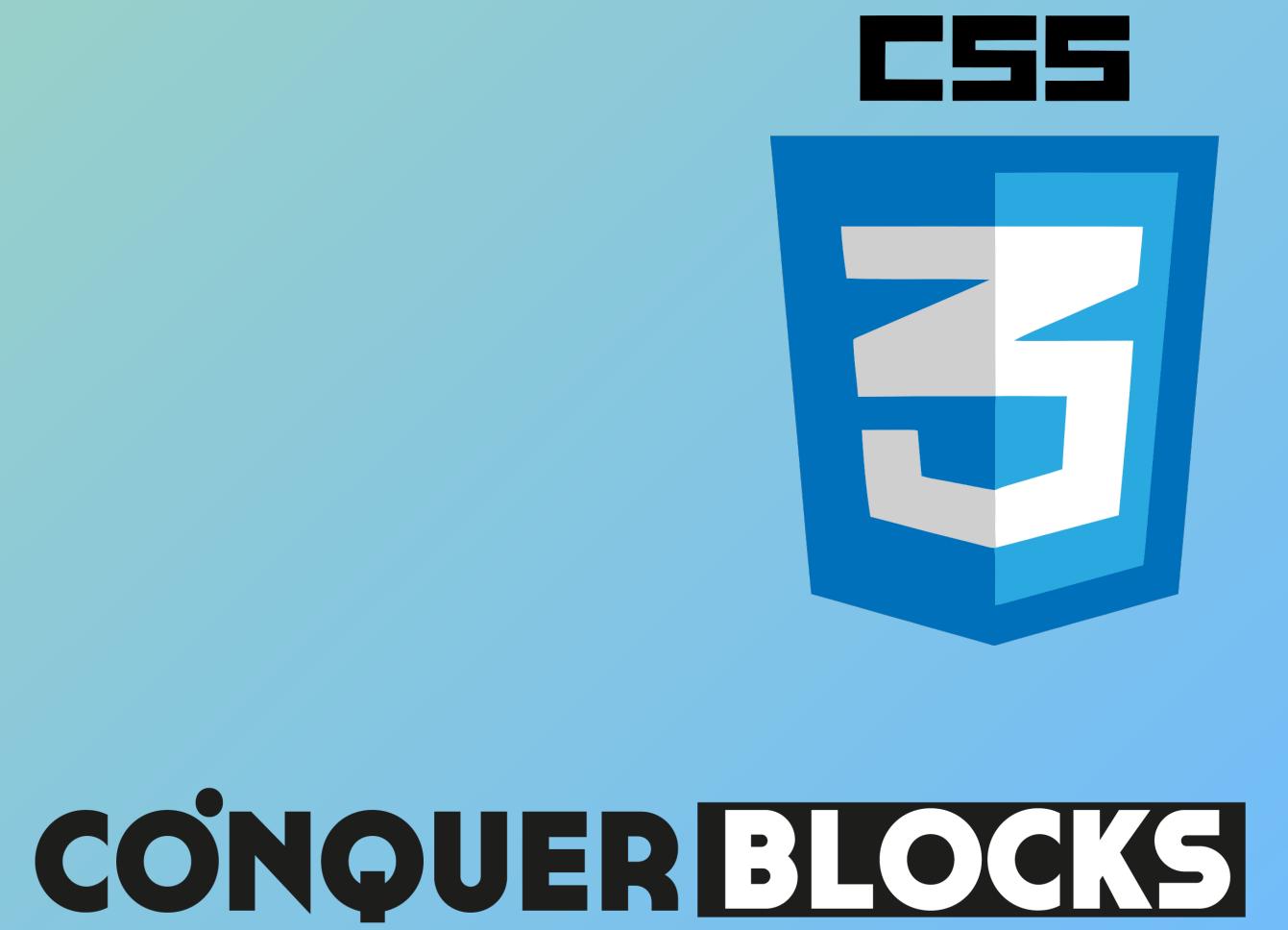


# {css}

Clase 08



# Clase anterior

**Clase anterior**

**Selectores III**

**Pseudoclases**

**Pseudoclases de hijos**

**Pseudoclases de hijos según tipo de elemento**

**Pseudoclases de interacción**

**Pseudoclases de ubicación**

# Clase 08

# <índice>

## Clase 7: Selectores III

**Pseudoclases de formularios**

---

**Pseudoelementos**

---

**Resolución Ejercicios Selectores II**

---

**Resolución Ejercicios Selectores III**

# Pseudoselectores de formulario

# Pseudoselectores de formulario

Existe una serie de pseudoclases en CSS que pueden ser utilizadas para usar en formularios de una página web. Estas pseudoclases permiten seleccionar elementos para darle estilo dependiendo de temas relacionados con los formularios o los campos que están en el interior de un formulario.

# Pseudoselectores de formulario

- Pseudoclases de **interacción** en formularios: Seleccionar elementos cuando cambia el estado de un elemento.
- Pseudoclases de **estado** en formularios: Seleccionar elementos cuando se encuentran en un estado concreto.
- Pseudoclases de **validación**: Seleccionar elementos si cumplen o no un cierto criterio de validación.

# Pseudoselectores de formulario

De interacción en formularios

- `:checked`: para seleccionar aquellos input de tipo radio o checkbox que estén marcados
- `:indeterminate`: Selecciona el elemento cuando la casilla está en un estado indeterminado

# Pseudoselectores de formulario

```
input:checked + span {  
    color: green;  
}
```

# Pseudoselectores de formulario

DEMO

# Pseudoselectores de formulario

## De estado

Existen varias pseudoclases para detectar si un campo de un formulario está activado o desactivado, o cierta información relacionada con su estado

# Pseudoselectores de formulario

## De estado

Pseudoclase	Descripción
:enabled	Selecciona cuando el campo del formulario está activado.
:disabled	Selecciona cuando el campo del formulario está desactivado.
:read-only	Selecciona cuando el campo es de sólo lectura.
:read-write	Selecciona cuando el campo es editable por el usuario.
:placeholder-shown	Selecciona cuando el campo está mostrando un placeholder.
:default	Selecciona cuando el elemento tiene el valor por defecto.

# Pseudoselectores de formulario

:enabled

Utilizando la autoexplicativa pseudoclase :enabled, podemos seleccionar elementos que se encuentren activados (comportamiento por defecto)

# Pseudoselectores de formulario

## :disabled

Sin embargo, lo más interesante de este tema viene al poder darle estilo a elementos desactivados con la pseudoclase :disabled, donde se seleccionan los elementos a los que se le ha añadido el atributo disabled

# Pseudoselectores de formulario

## :read-only

La pseudoclase :read-only selecciona aquellos elementos <input> de un formulario que están marcados con el atributo de sólo lectura readonly. La diferencia entre un campo con atributo disabled y un campo con atributo readonly es que la información del campo con readonly se enviará a través del formulario, mientras que la del campo con disabled no se enviará.

# Pseudoselectores de formulario

## :read-write

Por otro lado, la pseudoclase :read-write es muy útil para dar estilos a todos aquellos elementos que son editables por el usuario, sean campos de texto <input> o <textarea>.

# Pseudoselectores de formulario

## :placeholder-show

Con la pseudoclase :placeholder-show se nos permite seleccionar y dar estilo a los elementos que están actualmente mostrando un placeholder.

# Pseudoselectores de formulario

## De validación

En HTML5 es posible dotar de capacidades de validación a los campos de un formulario, pudiendo interactuar con ellos 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.

# Pseudoselectores de formulario

## De validación

Pseudoclase	¿Cuándo aplica estilos?
:required	Cuando el campo es obligatorio, o sea, tiene el atributo <code>required</code> .
:optional	Cuando el campo es opcional (por defecto, todos los campos).
:valid	Cuando los campos cumplen la validación HTML5.
:invalid	Cuando los campos no cumplen la validación HTML5.
:user-valid	Idem a :valid, pero cuando el usuario ha interactuado.
:user-invalid	Idem a :invalid, pero cuando el usuario ha interactuado.
:in-range	Cuando los campos numéricos están dentro del rango.
:out-of-range	Cuando los campos numéricos están fuera del rango.

# Pseudoselectores de formulario

## :required

Por lo general, los campos de un formulario son siempre opcionales. Para hacer obligatorio un campo, tenemos que indicar en el elemento HTML el atributo required, al cuál será posible darle estilo mediante la pseudoclase :required:

# Pseudoselectores de formulario

:optional

Por otra parte, los camposopcionales son todos aquellos que no tienen el atributo required. Pueden seleccionarse con la pseudoclase :optional

# Pseudoselectores de formulario

:valid

Actualmente, desde HTML5 se brinda un excelente soporte de validaciones desde el lado del cliente sin necesidad de Javascript

```
<input type="text" name="age" pattern="[0-9]+>
```

# Pseudoselectores de formulario

## :invalid

De forma opuesta, se pueden seleccionar elementos y aplicar ciertos estilos si no se cumple el patrón de validación

```
<input type="text" name="age" pattern="[0-9]+>
```

# Pseudoselectores de formulario

## :in-range

Lo ideal sería establecer un rango, algo que se suele hacer muy a menudo si tenemos campos numéricos de formulario mediante los atributos

min y max

```
<input type="number" name="age" min="18" max="100" />
```

# Pseudoselectores de formulario

:out-of-range

Lo contrario de lo anterior

```
<input type="number" name="age" min="18" max="100" />
```

# Pseudoelementos

# Pseudelementos

Los pseudelementos son una característica de CSS que permite hacer referencias a «comportamientos virtuales no tangibles», dicho de otra forma, seleccionar y dar estilo a elementos que no existen en el HTML, o que no son un simple elemento en sí.

# Pseudoelementos

La sintaxis de los pseudoelementos está precedida de dos puntos dobles (::) para diferenciarlos de las pseudoclases, las cuales sólo tienen dos puntos (:).

# Pseudelementos

```
selector #id .clase [atributo] :pseudoclasE ::pseudelemento {  
    propiedad : valor ;  
    propiedad : valor  
}
```

# Pseudelementos

Elemento	Descripción
content	Propiedad para generar contenido. Sólo usable en ::before o ::after.
::before	Aplica los estilos <b>antes</b> del elemento indicado.
::after	Aplica los estilos <b>después</b> del elemento indicado.
attr()	Función que hace referencia al atributo HTML indicado.

# Pseudoselectores de formulario

## ::content

La propiedad content admite parámetros de diverso tipo, incluso concatenando información mediante espacios. Podemos utilizar varios tipos de contenido

# Pseudoselectores de formulario

## ::content

Valor	Descripción	Ejemplo
STRING "..."	Añade el contenido de texto indicado.	content: "Contenido:";
attr( atributo )	Añade el valor del atributo HTML indicado.	content: attr(href);
IMAGE url(...)	Añade la imagen indicada en la URL.	content: url(icon.png);
GRADIENT	Añade un gradiente del tamaño indicado.	content: linear-gradient(red, blue);
COUNTER	Define un <a href="#">contador CSS</a> para mostrar.	content: counter(item);

# Pseudoselectores de formulario

## ::content

Ten en cuenta que la propiedad content sólo funciona dentro de los pseudoelementos ::before y ::after que explicaremos a continuación. No puede utilizarse en otro lugar.

# Pseudoselectores de formulario

## ::before / ::after

Los pseudoelementos ::before y ::after permiten hacer referencia a «justo antes del elemento» y «justo después del elemento», respectivamente. Además de selectores básicos como clase o id, combinadores, atributos o pseudoclases, puedes añadir un pseudoelemento, precedido por doble ::.

# Pseudoselectores de formulario

```
q::before {  
    content: "«";  
    color: #888;  
}
```

```
q::after {  
    content: "»";  
    color: #888;  
}
```

# Pseudoselectores de formulario

## Función attr()

Es interesante recalcar la utilidad de la función CSS attr(A). Esta función se puede utilizar en la propiedad content para recuperar el valor del atributo HTML especificado en A. Veamos un ejemplo para clarificarlo, concatenándolo con un texto aplicado en la propiedad content

# Pseudoselectores de formulario

## Función attr()

```
a::after {  
    content: " (" attr(href) ")";  
}
```

# Pseudoselectores de formulario

Este pequeño ejemplo de código CSS, realiza una búsqueda de elementos que sean enlaces `<a>`, y posteriormente (`::after`) muestra un contenido de texto posterior donde, envuelto entre paréntesis, escribe el contenido del atributo href del enlace tratado.

# Pseudelementos de resaltado

# Pseudelementos de resaltado

Existe una serie de pseudelementos orientados a la selección o resaltado de texto en un documento HTML mostrado a través de un navegador

# Pseudelementos de resaltado

Pseudelemento	Descripción
::selection	Aplica estilos al fragmento de texto seleccionado por el usuario.
::target-text	Aplica estilos al fragmento de texto enlazado tras el ancla de la URL.
::spelling-error	Aplica estilos al fragmento de texto resaltado por error ortográfico.
::grammar-error	Aplica estilos al fragmento de texto resaltado por error gramatical.

# Pseudoselectores de resaltado

## ::selection

Cuando seleccionamos un fragmento de texto, el navegador suele aplicar un color de fondo que depende del sistema operativo, del tema, o similar.

# Pseudoselectores de resaltado

```
::selection {  
background: indigo;  
color: white;  
}
```

# Pseudelementos tipográficos

# Pseudelementos tipográficos

Aunque ::before y ::after suelen ser los ejemplos de pseudelementos más frecuentes, existen muchos otros pseudelementos. Por ejemplo, dentro de una categoría de pseudelementos tipográficos, podemos encontrar los pseudelementos ::first-letter o ::first-line:

# Pseudelementos tipográficos

## Pseudelemento Descripción

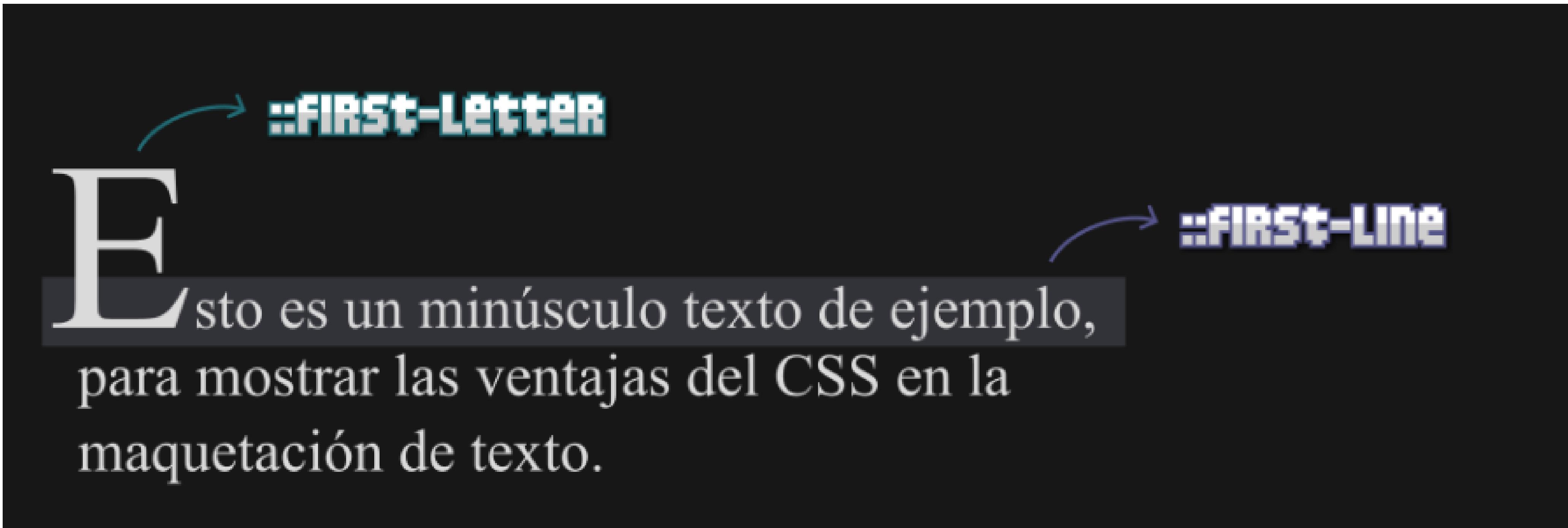
`::first-letter` Aplica los estilos en la primera letra del texto.

`::first-line` Aplica los estilos en la primera línea del texto.

# Pseudelementos tipográficos

```
p {  
    font-family: 'Open Sans', sans-serif;  
    font-size: 1rem;  
}  
  
p::first-letter {  
    font-family: 'Fancy Decorative', serif;  
    font-size: 4rem;  
}
```

# Pseudelementos tipográficos



# Ejercicios

# Ejercicios Selectores II

1. Selecciona los elementos div que sean hijos de otro div
2. Selecciona los hijos de los elementos p
3. Selecciona los elementos p que sean hijos de article
4. Selecciona los p que esten dentro de un article
5. Selecciona los titulos h1 y los h3
6. Selecciona los ol que esten dentro de body
7. Cualquier elemento que tenga la clase “amarillo”
8. Los elementos <ul> que sean hijos de <body>
9. Los elementos <a> que sean hijos de <p>
10. Selecciona el elemento con id “platano”
11. Todos los elementos <div> que sean hermanos de un elemento <h2>
12. Todos los elementos <p> con clase = paisa, descendientes de un elemento <ul>
13. Selecciona los <h1>, <h2>, <h3> que tengan la clase “headertitle”, hijos del header

# Ejercicios Selectores III

1. El primer elemento `<a>` hijo de `<p>`
2. Selecciona los checkbox que estén marcados, hijos de un `<form>`
3. Selecciona el último `<p>` que sea descendiente de `<article>`
4. Seleccionar el segundo hijo de un elemento `section`
5. Seleccionar los hijos pares de un elemento `ol`
6. Seleccionar el antepenúltimo `li` de un `ul` de longitud indefinida
7. El segundo `<a>` de dentro de un `<p>`
8. Los `button` que sean hijos únicos
9. Los `span` que sean primeros hijos de un `p`
10. Penúltimo hijo de `body`

# <Despedida>

Email

**bienvenidosaez@gmail.com**

Instagram

**@bienvenidosaez**

Youtube

**youtube.com/bienvenidosaez**

**CONQUERBLOCKS**