SAROJ PODDAR

WEB Technology

ICT. ED. 437 | BACHELOR
SANOTHIMI CAMPUS, BHAKTAPUR

PREPARED BY
SAROJ KUMAR PODDAR

PREPARED FOR
VIJAYA MALLA

What is a pseudo-class?

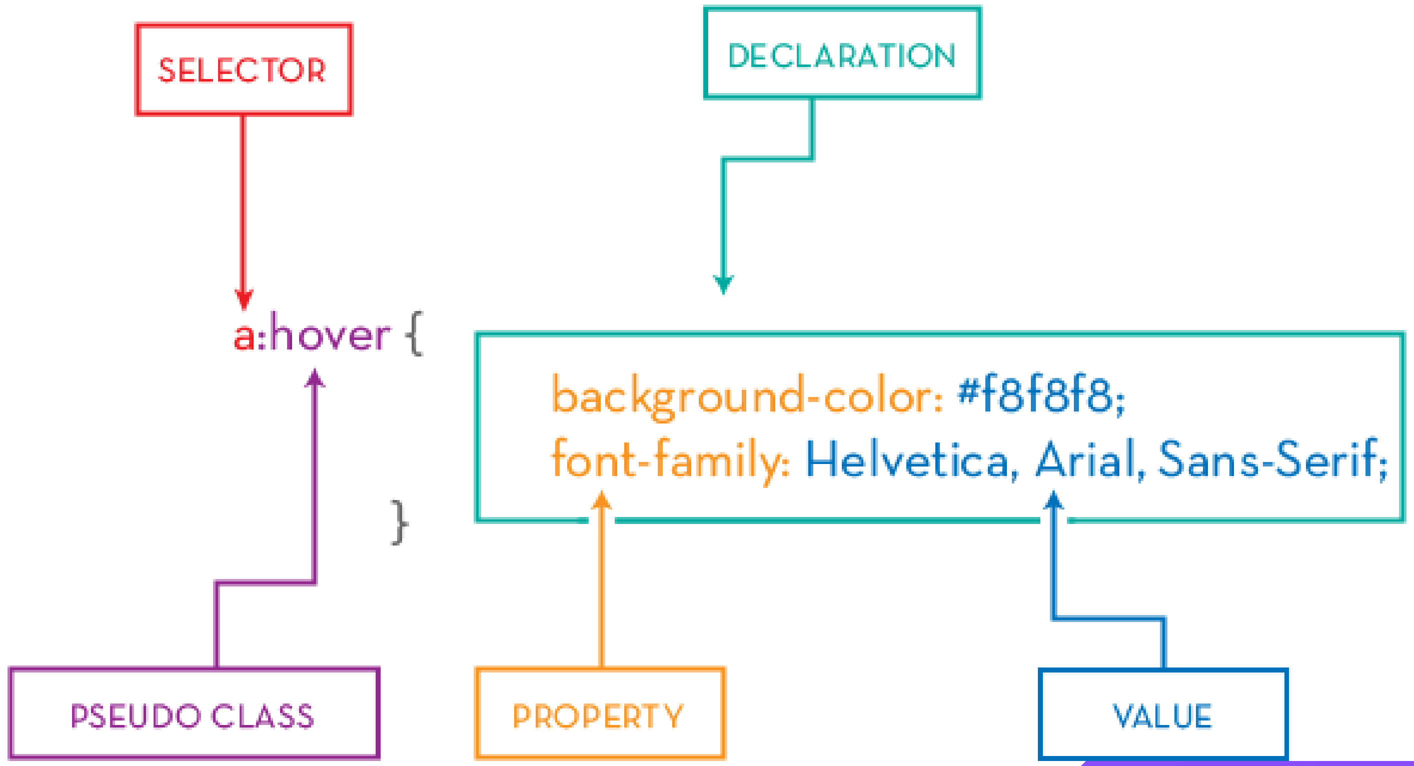
A pseudo-class is a selector that selects elements that are in a specific state, e.g. they are the first element of their type, or they are being hovered over by the mouse pointer. They tend to act as if you had applied a class to some part of your document, often helping you cut down on excess classes in your markup, and giving you more flexible, maintainable code.

Pseudo-classes enable you to target an element when it's in a particular state, as if you had added a class for that state to the DOM

Declaration *(selector:state)*

:pseudo-class-name

CLASS SELECTOR WITH PSEUDO CLASS CSS RULE



```
a:hover {
    background-color: #f8f8f8;
    font-family: Helvetica, Arial, Sans-Serif;
}
```

Pseudo-classes

selector:hover

- selects links on mouseover

a:link

- selects the unvisited links

a:visited

- selects the visited links.

a:active

- selects the active links

(a link the moment it is clicked)

selector:empty

- selects every element that has no children)

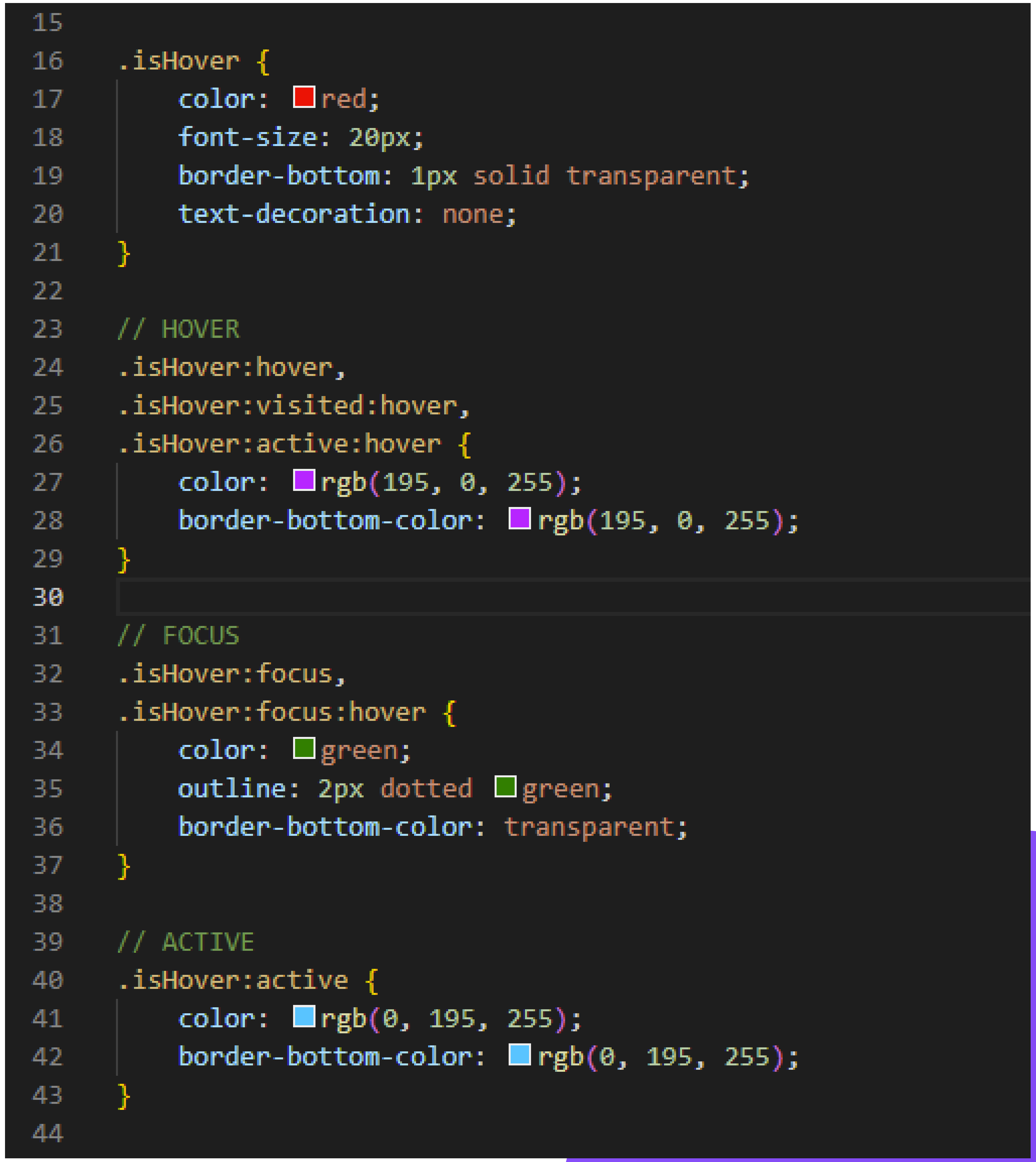
selector:first-child / ***selector:last-child***

- selects every elements that is the first/last element (child) of its container (parent).

selector:nth-child(number)

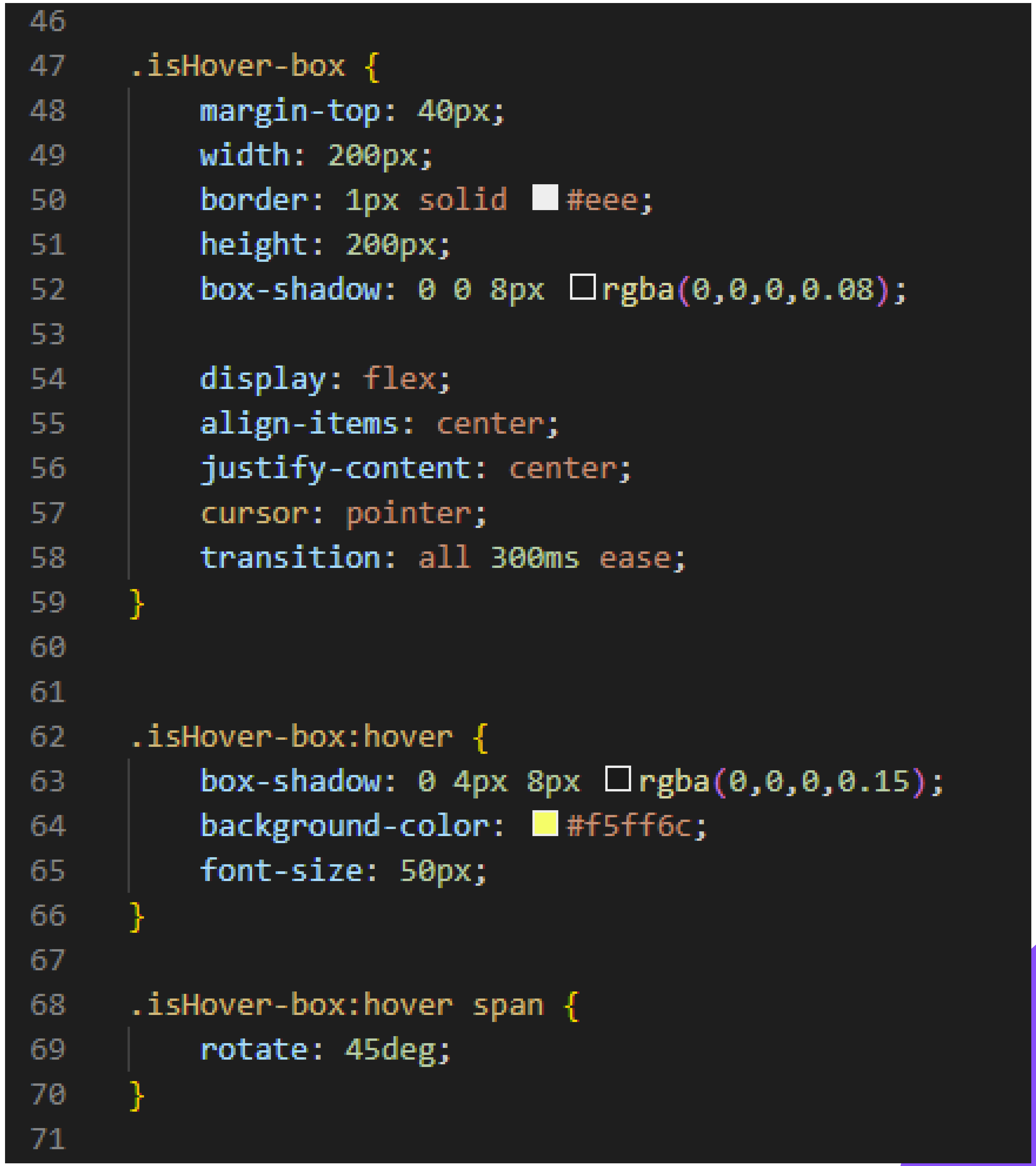
- selects every element that is the nth element (child) of its container (parent).

<a class="isHover" href="/">I am link 😊!</a>

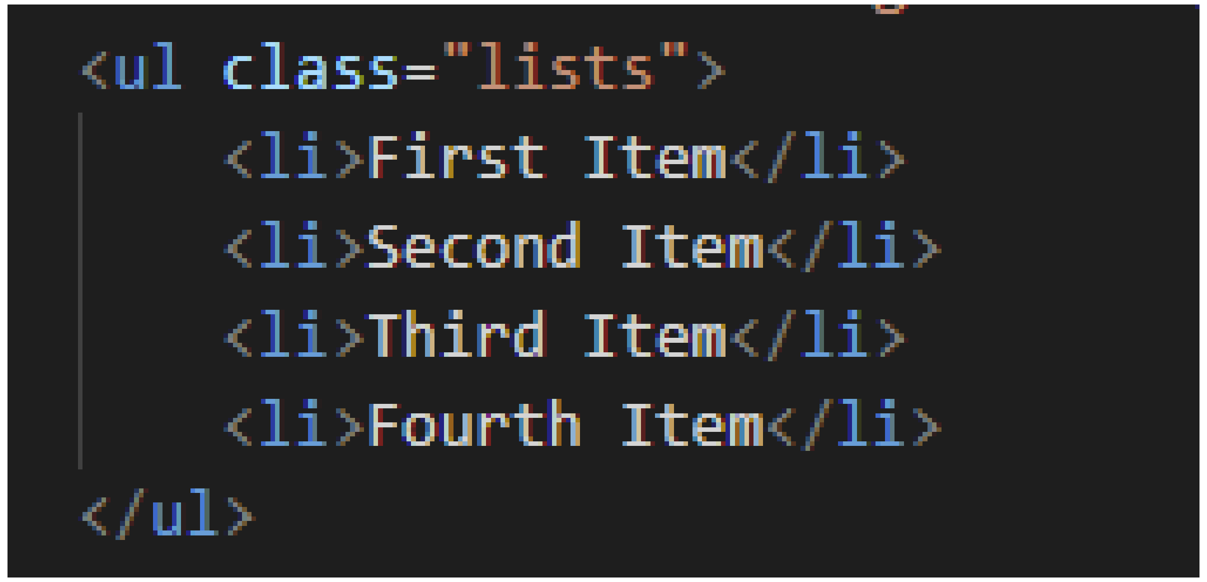


```css

.isHover {
    color: red;
    font-size: 20px;
    border-bottom: 1px solid transparent;
    text-decoration: none;
}

// HOVER
.isHover:hover,
.isHover:visited:hover,
.isHover:active:hover {
    color: rgb(195, 0, 255);
    border-bottom-color: rgb(195, 0, 255);
}

// FOCUS
.isHover:focus,
.isHover:focus:hover {
    color: green;
    outline: 2px dotted green;
    border-bottom-color: transparent;
}

// ACTIVE
.isHover:active {
    color: rgb(0, 195, 255);
    border-bottom-color: rgb(0, 195, 255);
}

```

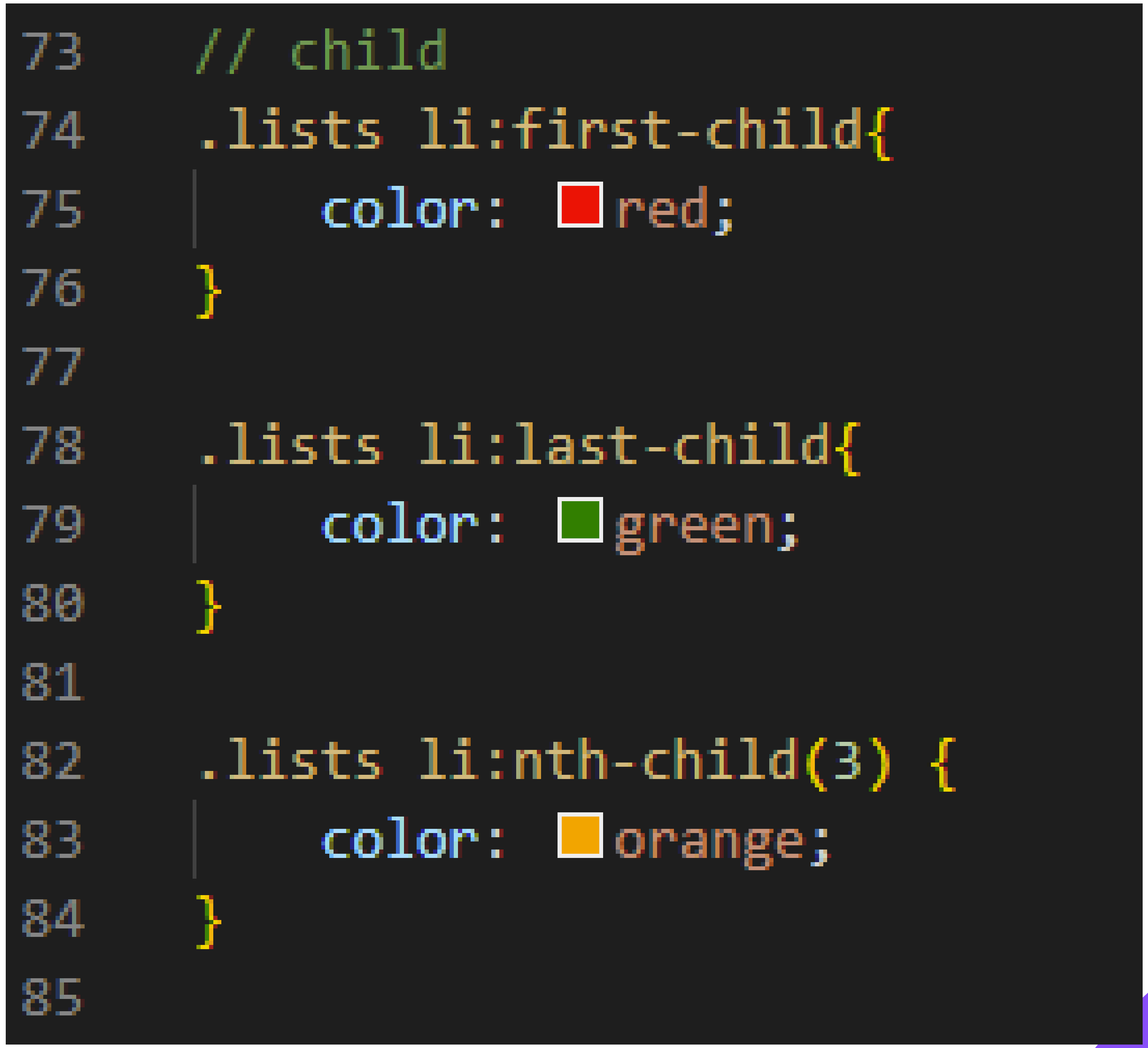
```
<div class="isHover-box">
   <span>  </span>
</div>
```



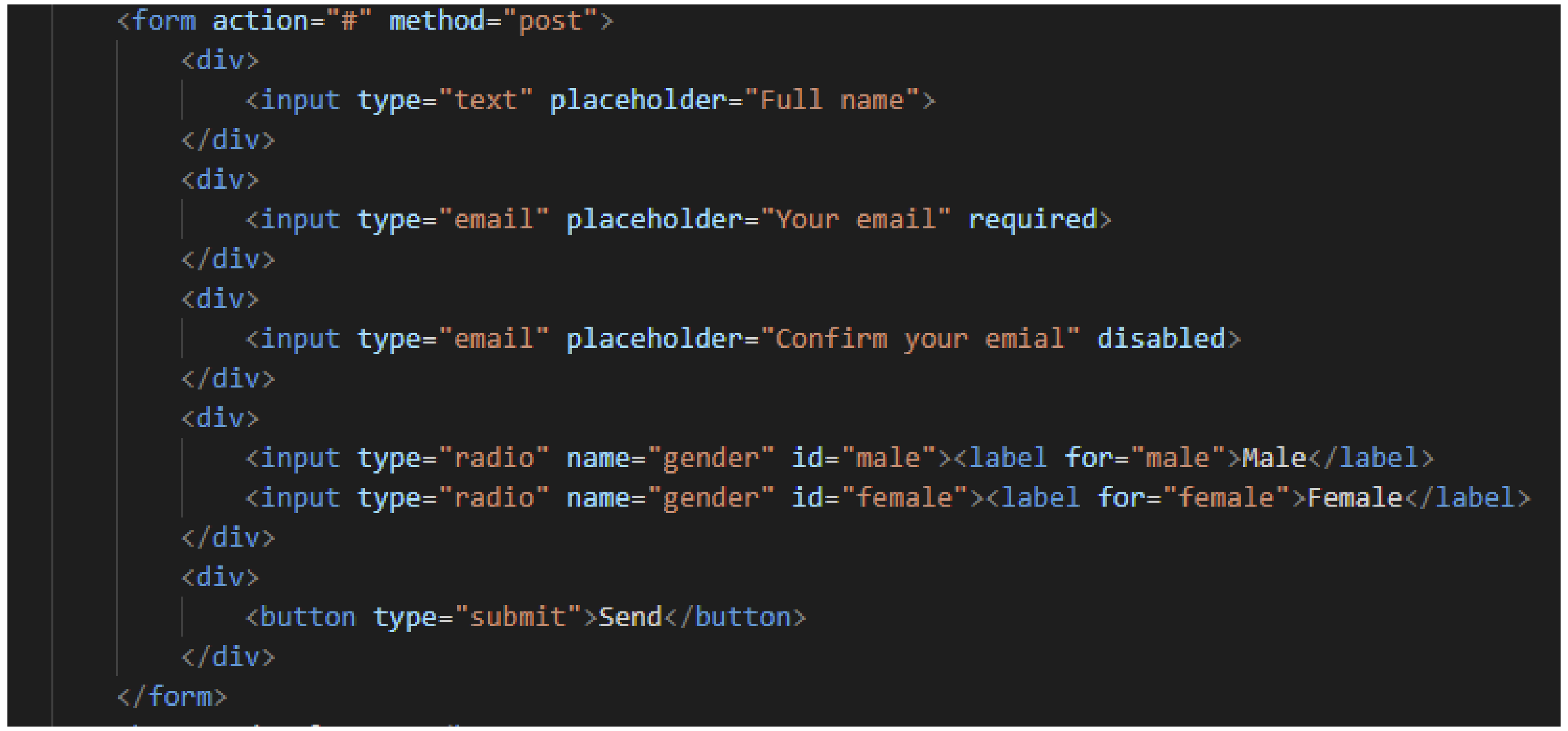
```
.isHover-box {
    margin-top: 40px;
    width: 200px;
    border: 1px solid #eee;
    height: 200px;
    box-shadow: 0 0 8px rgba(0,0,0,0.08);

    display: flex;
    align-items: center;
    justify-content: center;
    cursor: pointer;
    transition: all 300ms ease;
}


.isHover-box:hover {
    box-shadow: 0 4px 8px rgba(0,0,0,0.15);
    background-color: #f5ff6c;
    font-size: 50px;
}

.isHover-box:hover span {
    rotate: 45deg;
}
```



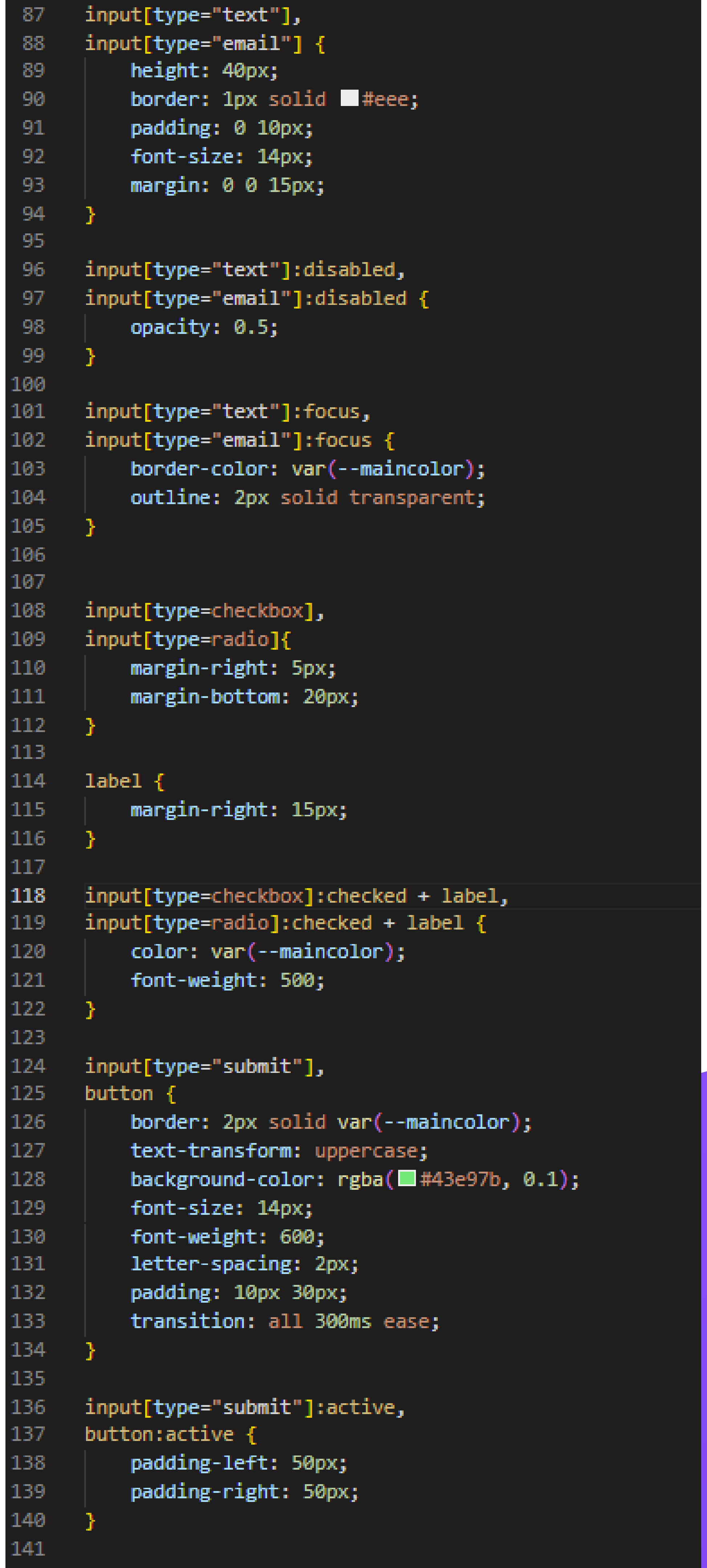
```html
<ul class="lists">
    <li>First Item</li>
    <li>Second Item</li>
    <li>Third Item</li>
    <li>Fourth Item</li>
</ul>
```



```css
// child
.lists li:first-child{
    color: red;
}

.lists li:last-child{
    color: green;
}

.lists li:nth-child(3) {
    color: orange;
}

```



```html
<form action="#" method="post">
    <div>
        <input type="text" placeholder="Full name">
    </div>
    <div>
        <input type="email" placeholder="Your email" required>
    </div>
    <div>
        <input type="email" placeholder="Confirm your emial" disabled>
    </div>
    <div>
        <input type="radio" name="gender" id="male"><label for="male">Male</label>
        <input type="radio" name="gender" id="female"><label for="female">Female</label>
    </div>
    <div>
        <button type="submit">Send</button>
    </div>
</form>
```



```css
input[type="text"],
input[type="email"] {
    height: 40px;
    border: 1px solid #eee;
    padding: 0 10px;
    font-size: 14px;
    margin: 0 0 15px;
}

input[type="text"]:disabled,
input[type="email"]:disabled {
    opacity: 0.5;
}

input[type="text"]:focus,
input[type="email"]:focus {
    border-color: var(--maincolor);
    outline: 2px solid transparent;
}


input[type=checkbox],
input[type=radio]{
    margin-right: 5px;
    margin-bottom: 20px;
}

label {
    margin-right: 15px;
}

input[type=checkbox]:checked + label,
input[type=radio]:checked + label {
    color: var(--maincolor);
    font-weight: 500;
}

input[type="submit"],
button {
    border: 2px solid var(--maincolor);
    text-transform: uppercase;
    background-color: rgba(#43e97b, 0.1);
    font-size: 14px;
    font-weight: 600;
    letter-spacing: 2px;
    padding: 10px 30px;
    transition: all 300ms ease;
}

input[type="submit"]:active,
button:active {
    padding-left: 50px;
    padding-right: 50px;
}

```

Pseudo-classes (HTML Form)

input:focus

- selects input on focus state

input:checked (checkbox/radio)

- selects the checked element

input:disabled / input:enabled

- selects the disabled/enabled element

input/button:active

- selects the active element

input:read-only

- selects the read-only element

input:required

- selects the required element

Pseudo-classes (others)

:lang

- The CSS :lang pseudo-class allows to specify the special rules for different languages

:root

- The CSS :root pseudo-class matches the document’s root element (<html> element):

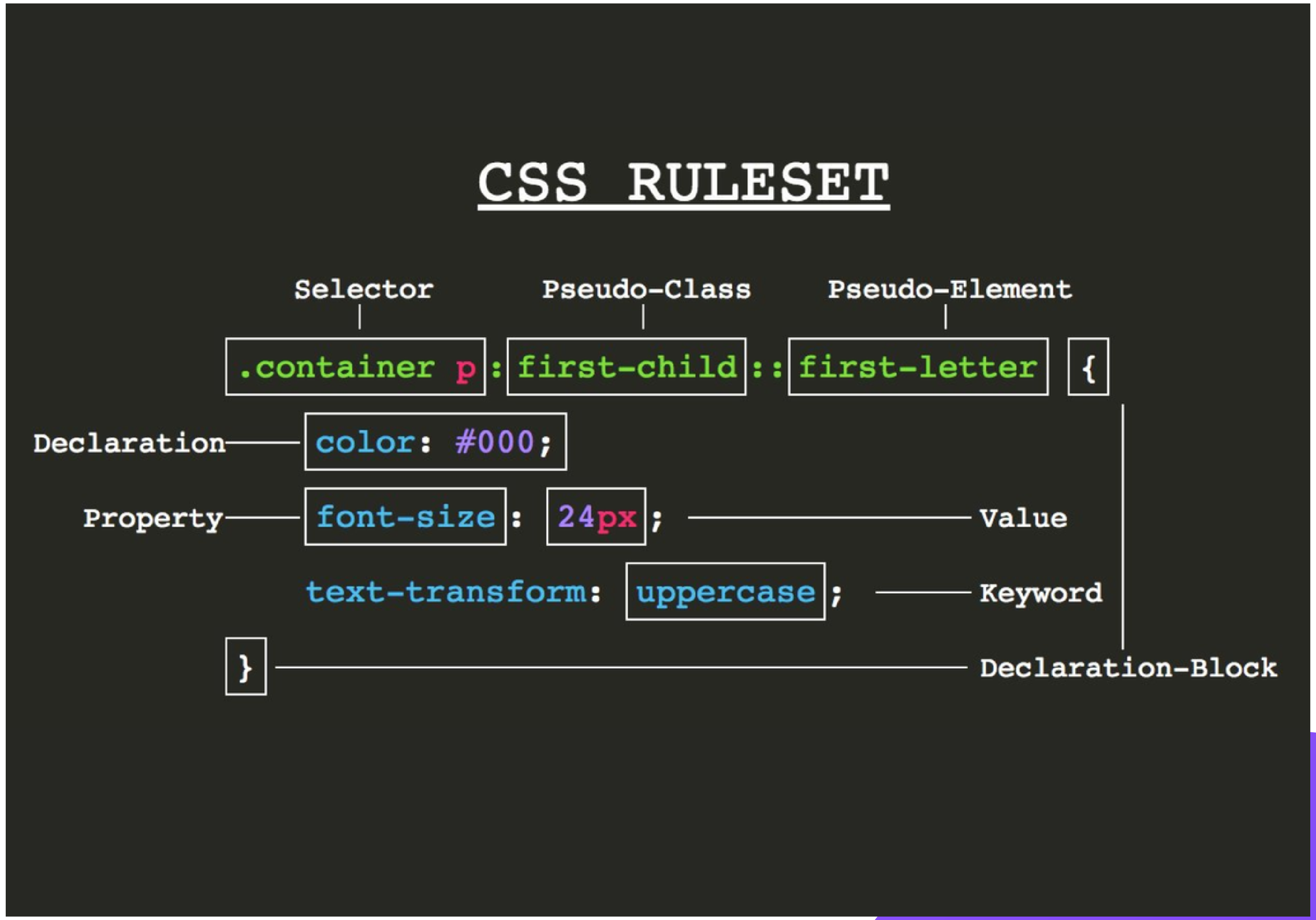
What is a pseudo-element?

Pseudo-elements behave in a similar way. However, they act as if you had added a whole new HTML element into the markup, rather than applying a class to existing elements. Pseudo-elements start with a double colon ::.

Pseudo-elements act as if you had added a whole new element to the DOM, and enable you to style that.

Declaration *(selector:state)*

::pseudo-element-name



CSS RULESET

```
.container p:first-child::first-letter {
    color: #000;
    font-size: 24px;
    text-transform: uppercase;
}
```

Pseudo-elements

selector::first-line

- To add the special effect to your first line of the text use the ***:first-line*** pseudo-element.

font/color/background/decoration/letter-spacing etc.

selector::first-letter

- To add the special effect to your first character of the text use the :first-letter pseudo-element.

font/color/background/decoration/letter-spacing etc.

selector::before

- The :before pseudo-element places content before the content of an element.

selector::after

- The :after pseudo-element places content after the content of an element.

selector::selection

- The :selection pseudo-element matches the portion of an element that is selected by a user.

color/background/cursor/outline