Web programming I

Section 4

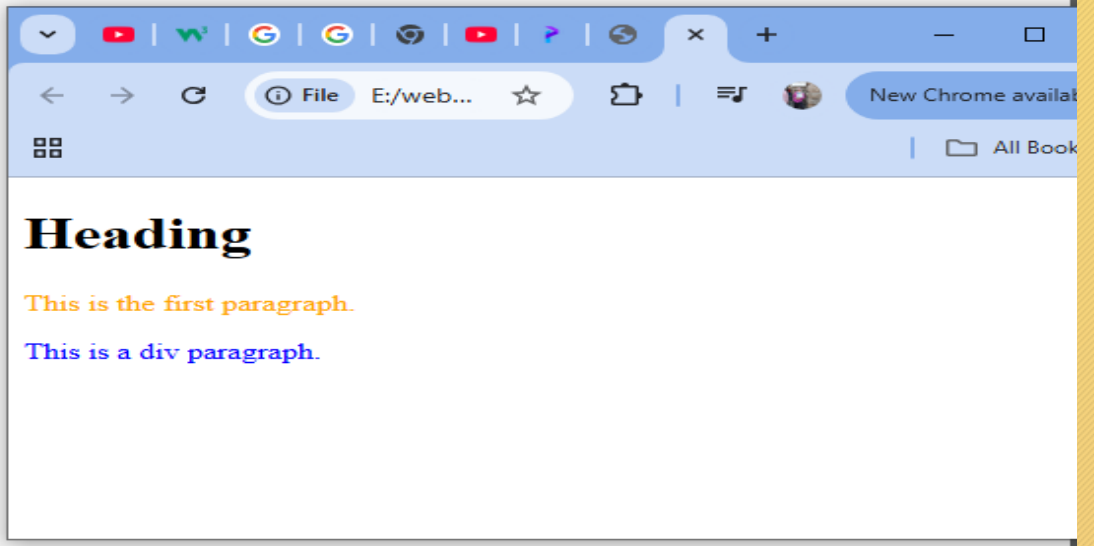
Eng/ Mona AbdelAziz

Selectors

- selectors are used to "find" (or select) the HTML elements you want to style Selectors can be divided into five categories
- simple selectors(type-class-id-group-universal)
- Combinator selectors.
- Attribute selectors.
- Pseudo-class selectors.
- Pseudo-elements selectors.

Descendant combinator (space)

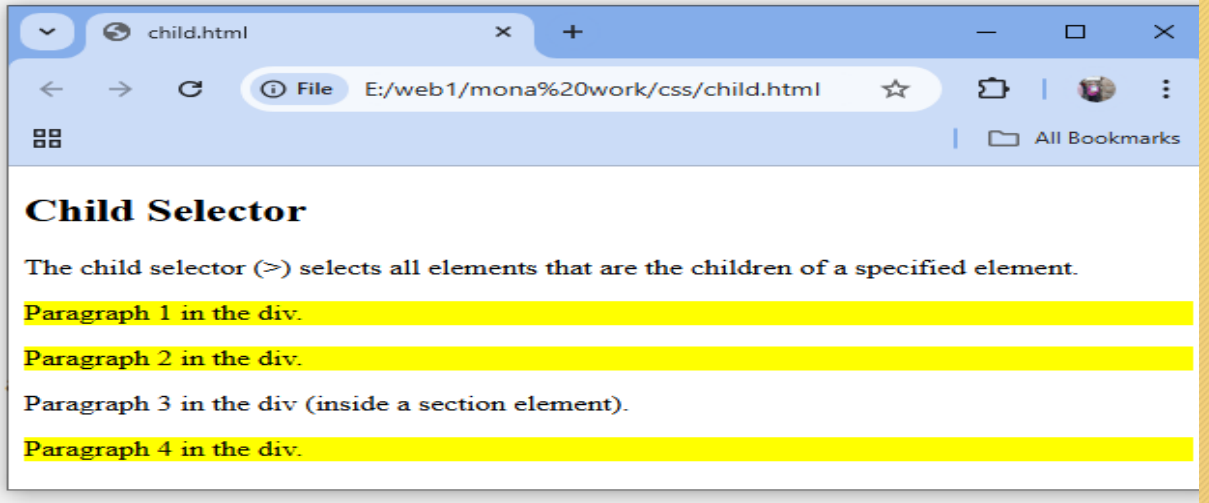
```
1   <!DOCTYPE html>
2   <html>
3   <head>
4   <style>
5   div p {
6       color: blue;
7   }
8
9   p {
10      color: orange;
11  }
12  </style>
13  </head>
14  <body>
15      <h1>Heading</h1>
16      <p>This is the first paragraph.</p>
17      <div>
18              <p>This is a div paragraph.</p>
19          </div>
20  </body>
21  </html>
```



child selector (>)

- selects all elements that are the children of a specified element.

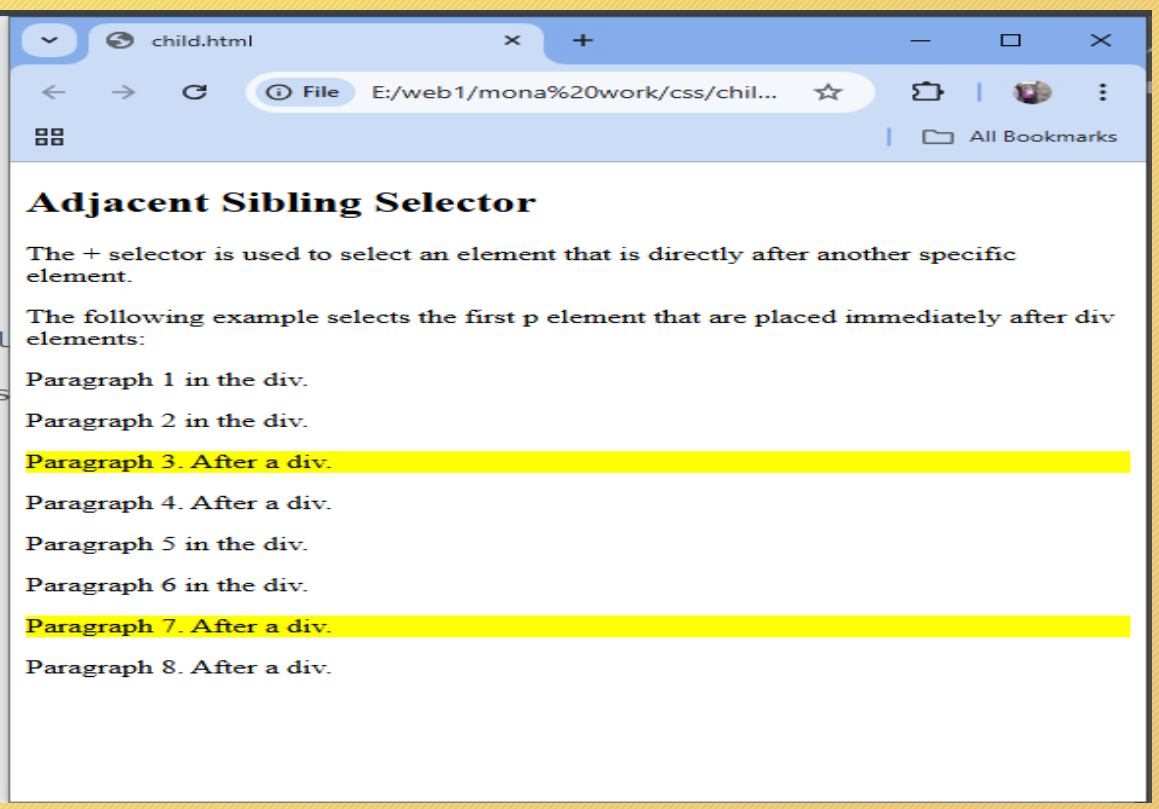
```
<!DOCTYPE html>
<html>
<head>
<style>
div > p {
  background-color: yellow;
}
</style>
</head>
<body>

<h2>Child Selector</h2>

<p>The child selector (>) selects
</p>
<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
  <section>
    <!-- not Child but Descendant -->
    <p>Paragraph 3 in the div (inside a section element).</p>
  </section>
  <p>Paragraph 4 in the div.</p>
</div>
</body>
</html>
```



Next Sibling Combinator (+)

- used to select an element that is directly after another specific element

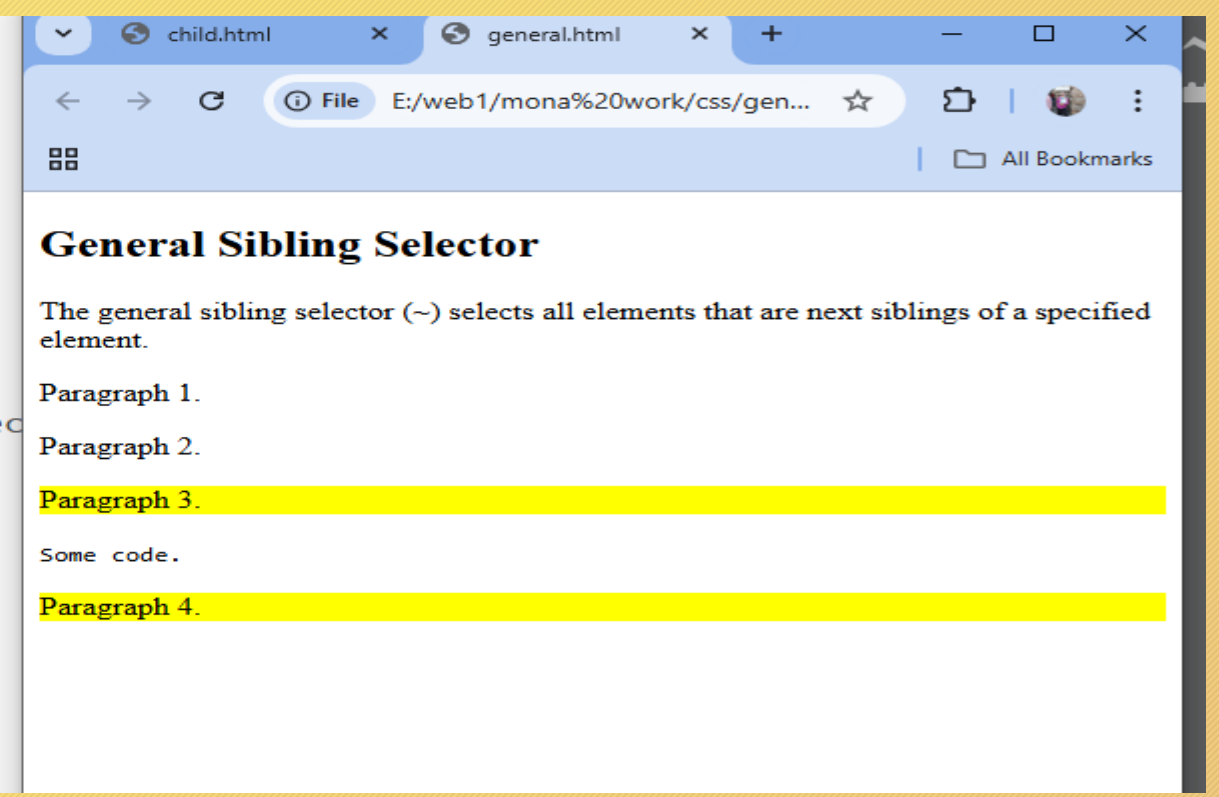
```
<!DOCTYPE html>
<html>
<head>
<style>
div + p {
  background-color: yellow;
}
</style>
</head>
<body>
<h2>Adjacent Sibling Selector</h2>
<p>The + selector is used to select an el
element.</p>
<p>The following example selects the firs
elements:</p>
<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
</div>
<p>Paragraph 3. After a div.</p>
<p>Paragraph 4. After a div.</p>
<div>
  <p>Paragraph 5 in the div.</p>
  <p>Paragraph 6 in the div.</p>
</div>
<p>Paragraph 7. After a div.</p>
<p>Paragraph 8. After a div.</p>
</body>
</html>
```



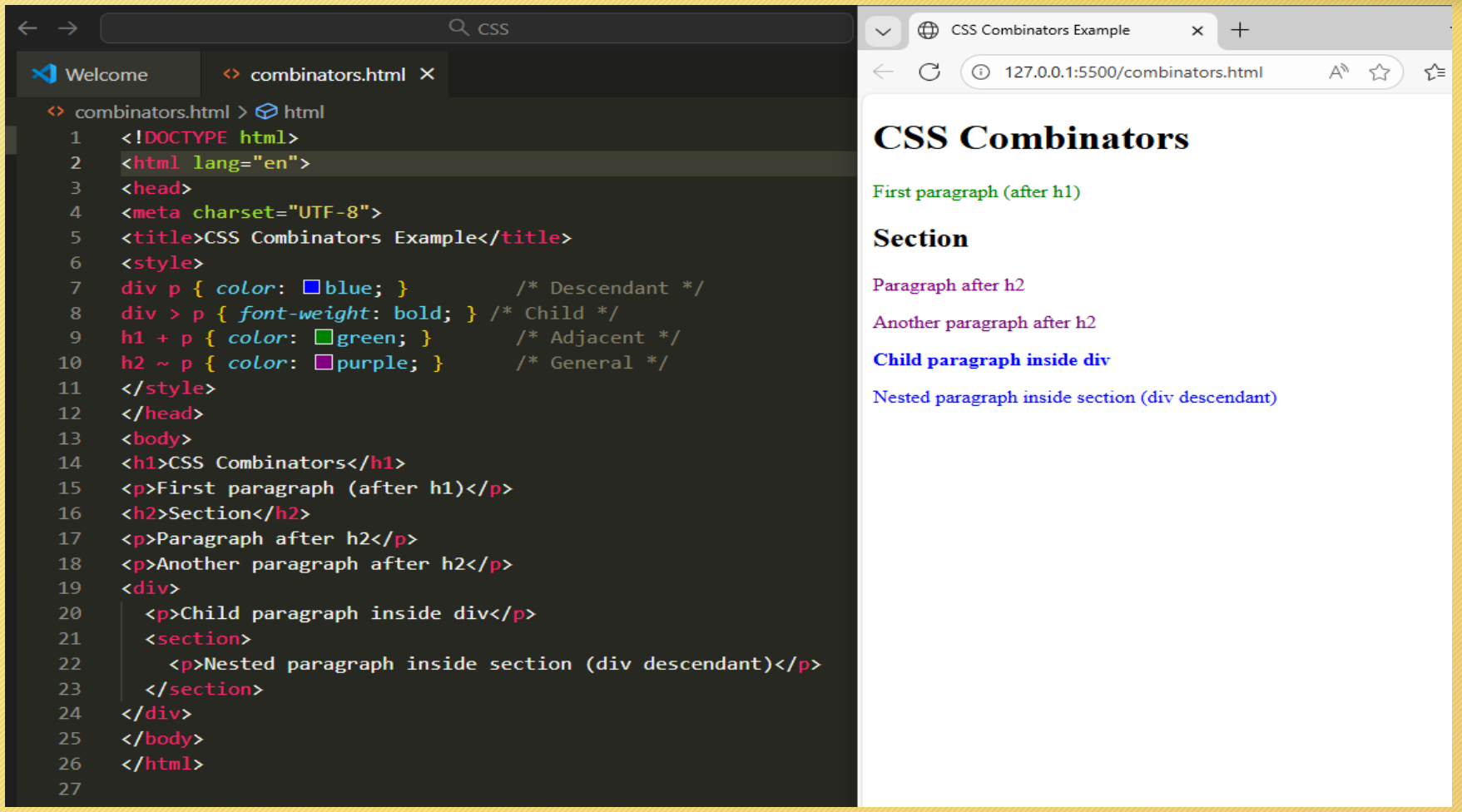
general-sibling Combinator (~)

- selects all elements that are next siblings of a specified element.

```
<!DOCTYPE html>
<html>
<head>
<style>
div ~ p {
  background-color: yellow;
}
</style>
</head>
<body>
<h2>General Sibling Selector</h2>
<p>The general sibling selector (~) selec
specified element.</p>
<p>Paragraph 1.</p>
<div>
  <p>Paragraph 2.</p>
</div>
<p>Paragraph 3.</p>
<code>Some code.</code>
<p>Paragraph 4.</p>
</body>
</html>
```



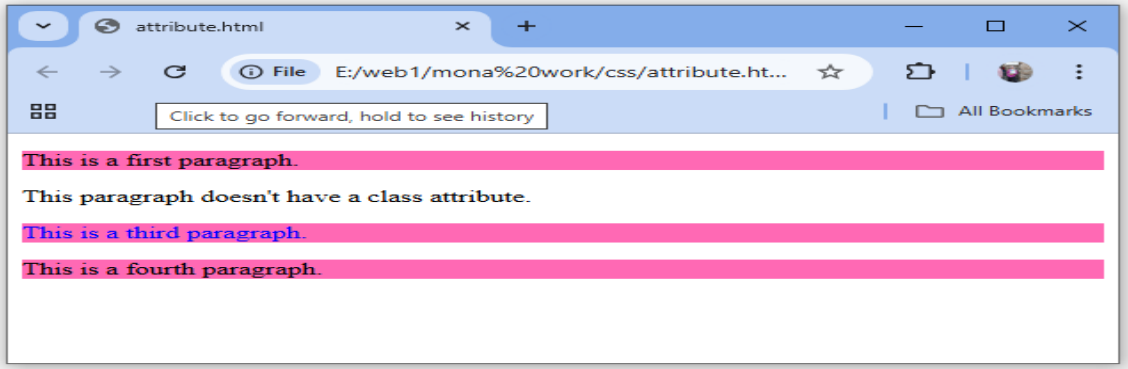
All types Example



Attribute Selector

- The [attribute] selector is used to select elements with a specified attributeallow you to style HTML elements **based on their attributes or attribute values.**
- Syntax : **element[Attribute]**

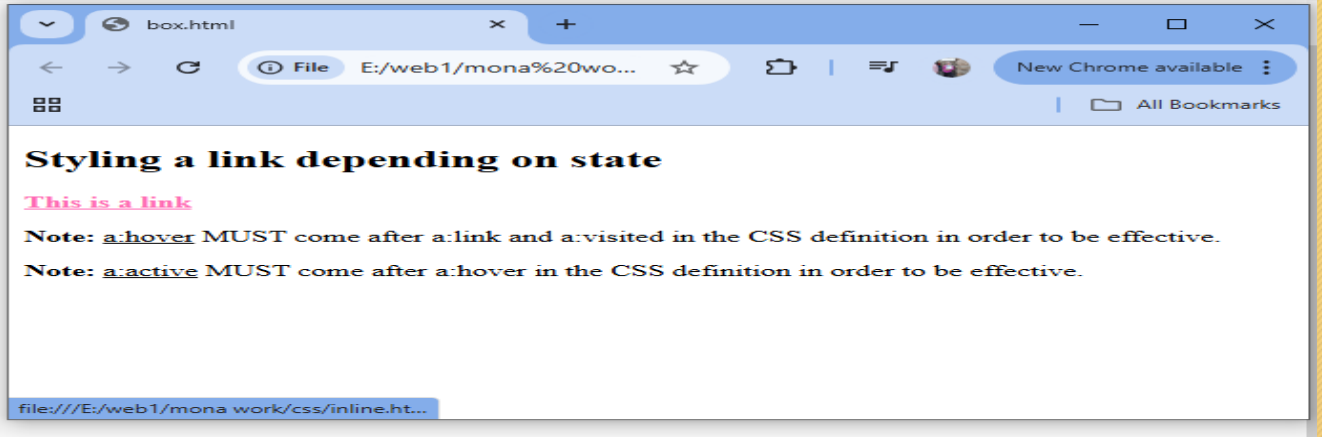
```
<!DOCTYPE html>
<html>
<head>
<style>
p[class] {
    background-color: hotpink;
}

p[class="third"] {
    color: blue;
}
</style>
</head>
<body>

 <body>
        <p class="first">This is a first paragraph.</p>
        <p>This paragraph doesn't have a class attribute.</p>
        <p class="third">This is a third paragraph.</p>
        <p class="fourth">This is a fourth paragraph.</p>
</body>

</body>
</html>
```



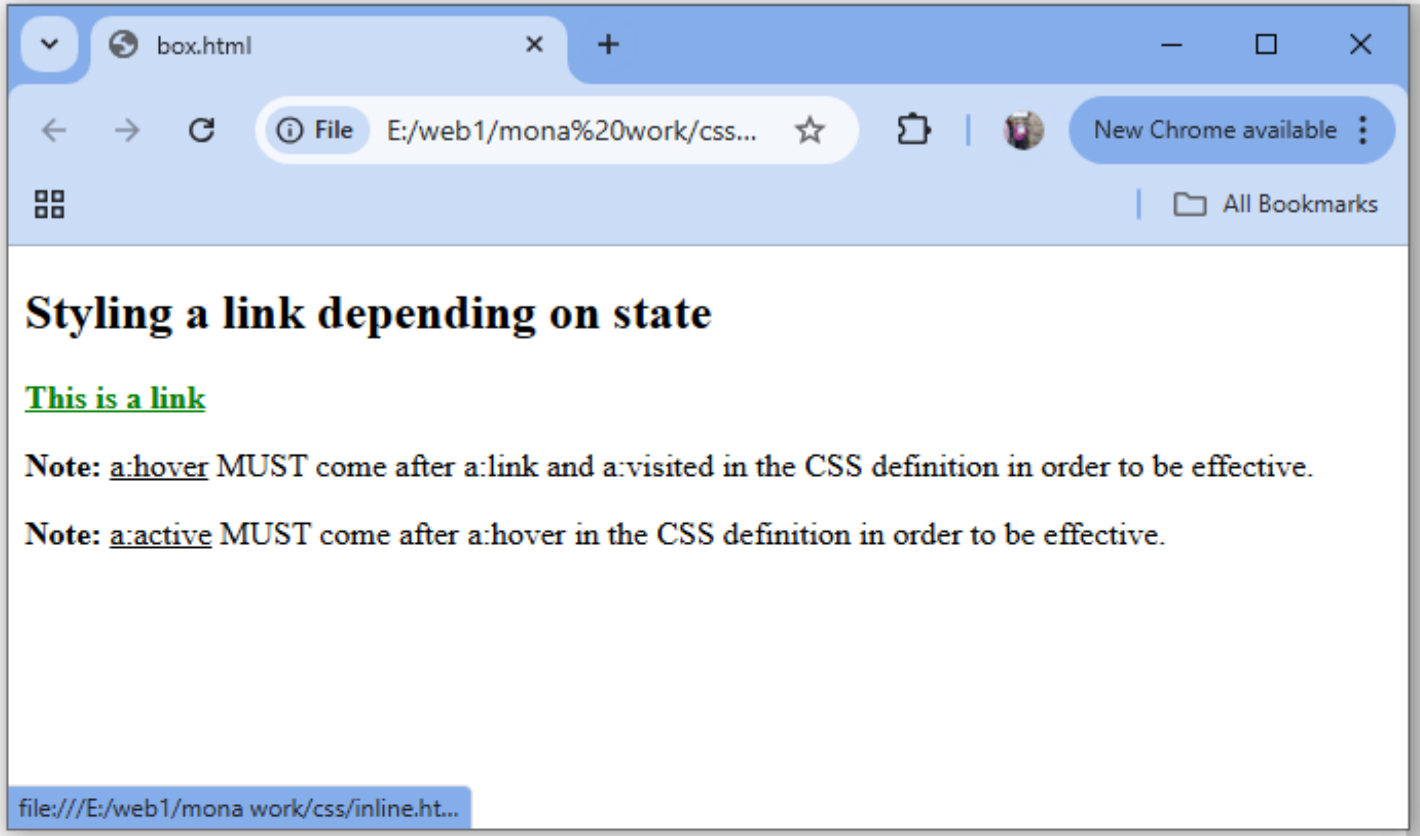
CSS Pseudo-Classes Selectors

A pseudo-class is used to define a special state of an element. Pseudo-classes are keywords that can be added to selectors to style elements **based on their state, position, or interaction** (like when you hover, click, or focus).

```
<head>
<style>
/* unvisited link */
a:link {
  color: red;
}
/* visited link */
a:visited {
  color: green;
}
/* mouse over link */
a:hover {
  color: hotpink;
}
/* selected link */
a:active {
  color: blue;
}
</style>
</head>
<body>
<h2>Styling a link depending on state</h2>
<p><b><a href="inline.html" target="_blank">This is a link</a></b></p>
<p><b>Note:</b> <u>a:hover</u> MUST come after a:link and a:visited in the CSS definition in order to be effective.</p>
<p><b>Note:</b> <u>a:active</u> MUST come after a:hover in the CSS definition in order to be effective.</p>
</body>
</html>
```

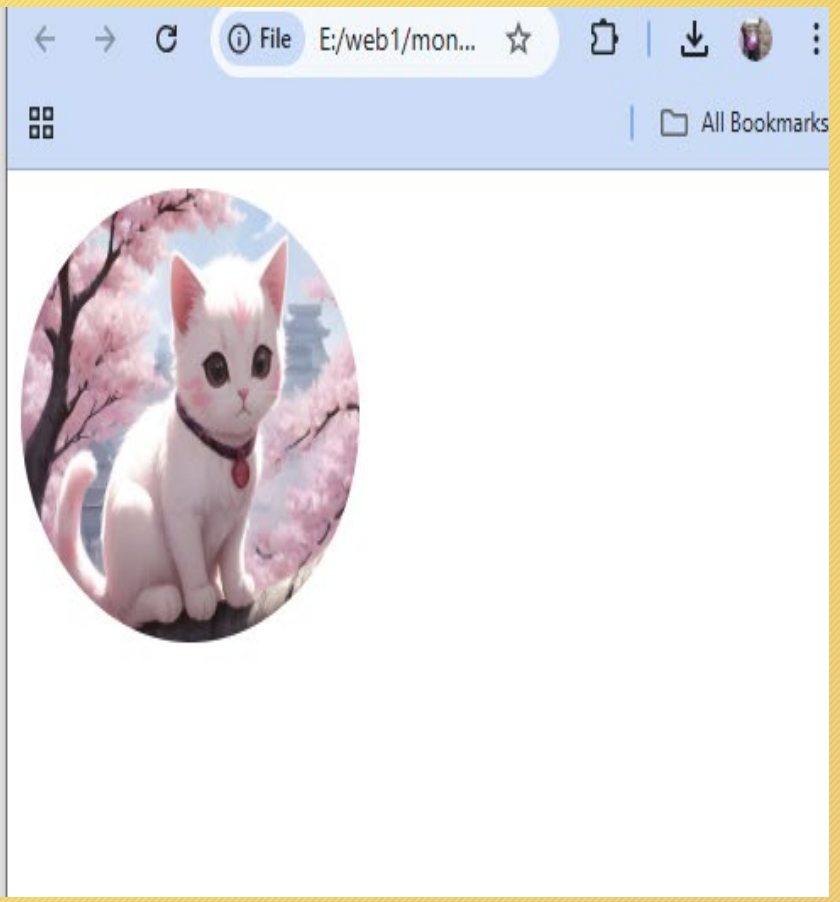


```
<!DOCTYPE html>
<html>
<head>
<style>
    a:active {
  color: blue;
}
    a:hover {
  color: hotpink;
            }
/* unvisited link */
a:link {
  color: red;
}
/* visited link */
a:visited {
  color: green;
}
/* mouse over link */

/* selected link */

</style>
</head>
<body>
<h2>Styling a link depending on state</h2>
<p><b><a href="inline.html" target="_blank">This is a link</a></b></p>
<p><b>Note:</b> <u>a:hover</u> MUST come after a:link and a:visited in the CSS definition in order to be effective.</p>
<p><b>Note:</b> <u>a:active</u> MUST come after a:hover in the CSS definition in order to
```



Example2

```html
<!DOCTYPE html>
<html>
<head>
<style>
    img:hover{
        width: 200px;
        height: 200px;
        border-radius: 70%;
    }
</style>
</head>
<body>
<img src="cat.jpg">
</body>
</html>
```



Pseudo-elements selectors.

CSS pseudo-elements selectors select the specific part of HTML elements.

- Style the first letter or line, of an element
- Insert content before or after an element
- Style the markers of list items
- Style the viewbox behind a dialog box

Syntax

element: specifies the HTML element
pseudo-element: specifies the specific part of the element that we want to target
Pseudo-element keywords are added to the selectors and preceded by a double colon **(::)**

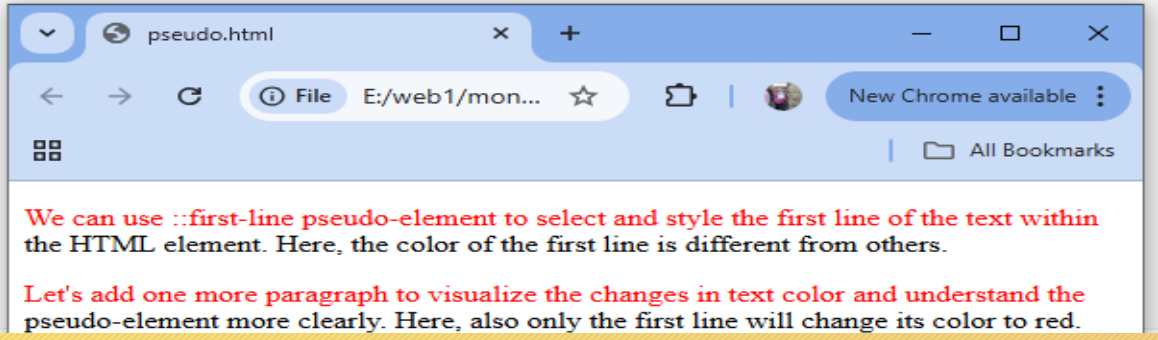
```
element::pseudo-element {
  /* CSS styles */
}
```

Types of Pseudo-Elements

- **::first-line:** selects the first line of text within a block-element
- **::first-letter:** selects the first letter of text
- **::before:** inserts content before the actual content of element
- **::after:** inserts content after the actual content of element
- **::marker:** selects the marker of list elements
- **::selection:** styles the user selected part of HTML elements

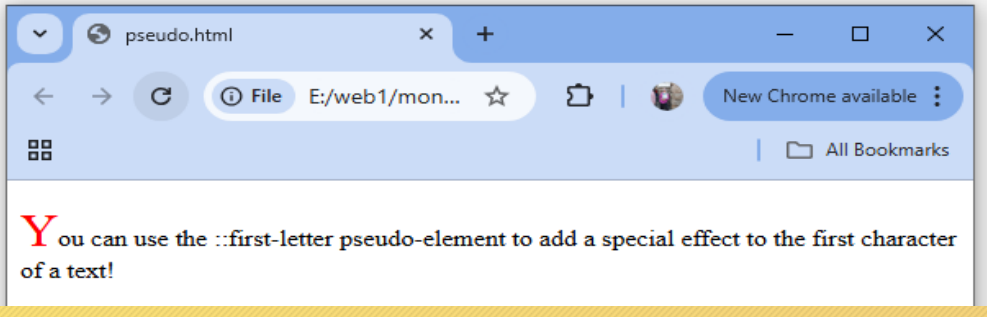
CSS first-line Pseudo- Element

```
<!DOCTYPE html>
<html>
<head>
<style>
p::first-line {
    color: red;
}
</style>
</head>
<body>
<p>
        We can use ::first-line pseudo-element to select and style
        the first line of the text within the HTML element. Here, the color
        of the first line is different from others.
    </p>
    <p>
        Let's add one more paragraph to visualize the changes in text color
        and understand the pseudo-element more clearly. Here, also only the first
        line will change its color to red.
    </p>
</body>
</html>
```



CSS first-letter Pseudo-Element

```
<!DOCTYPE html>
<html>
<head>
<style>
p::first-letter {
  color: #ff0000;
  font-size: xx-large;
}
</style>
</head>
<body>

<p>You can use the ::first-letter pseudo-element to add a special effect to the first character of a text!</p>

</body>
</html>
```



Before pseudo element

```
<!DOCTYPE html>
<html>
<head>
<style>
 h1::before {
  content: url(icon.png);
   }
</style>
 </head>
<body>
<h1>The general sibling selector selects all elements that
are siblings of a specified element</h1>
</body>
</html>
```

after Pseudo-element

Debug Help

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <style>
5   h3::after {
6
7    content: url(icon.png);
8     }
9
10 </style>
11  </head>
12 <body>
13 <h3>The general sibling selector selects all elements that
   are siblings of a specified elementThe general sibling
   selector selects all elements that are siblings of a
   specified element.The general sibling selector selects all
   elements that are siblings of a specified elementThe
   general sibling selector selects all elements that are
   siblings of a specified element</h3>
14 </body>
```

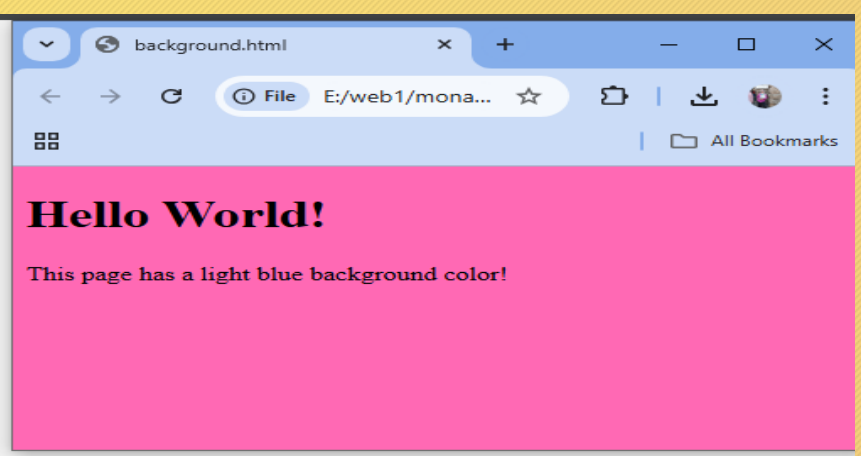
CSS Backgrounds

- The CSS background properties are used to add background effects for elements.

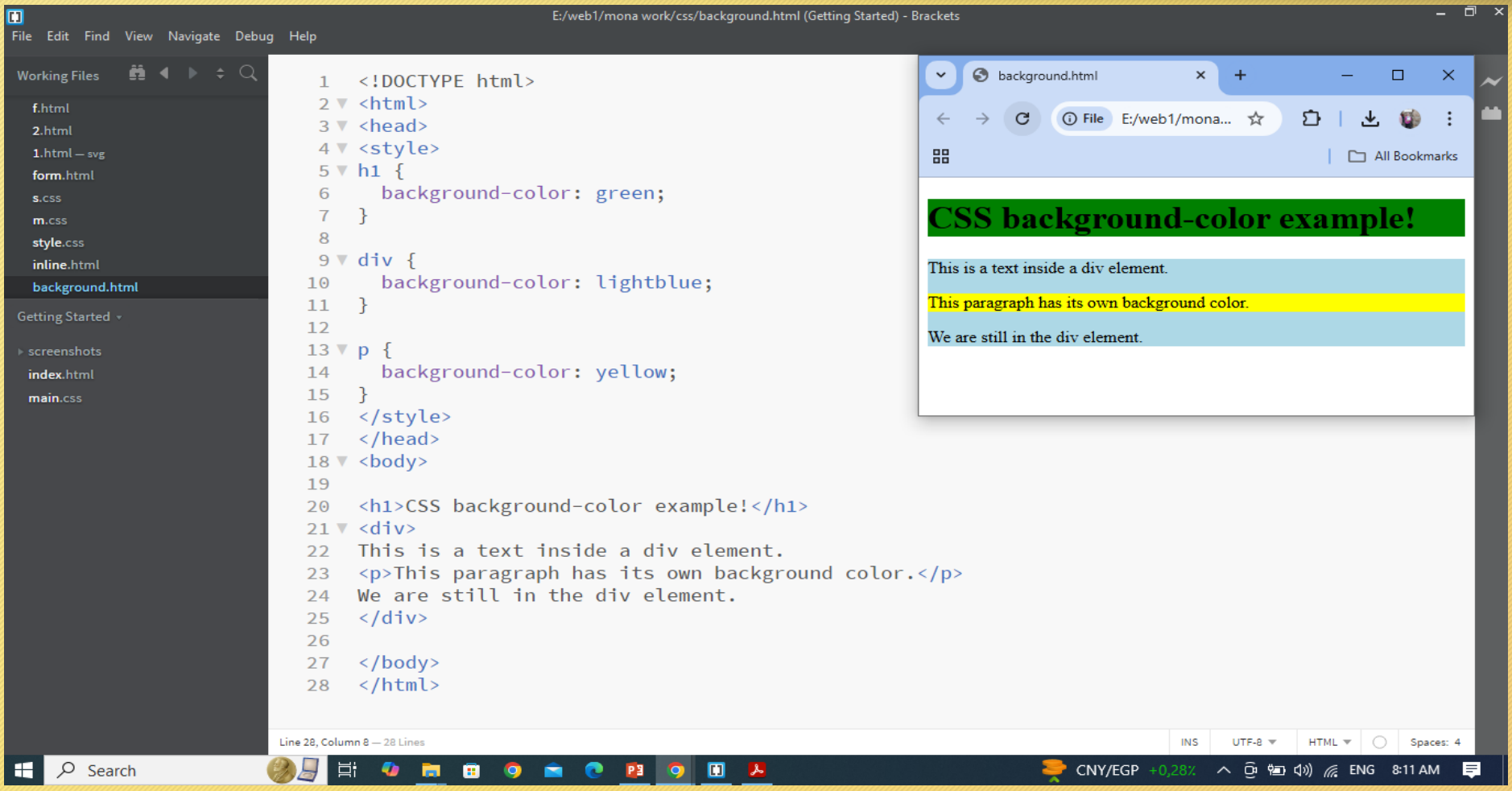
CSS background-color

- The background-color property specifies the background color of an element.

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color:hotpink;
}
</style>
</head>
<body>

<h1>Hello World!</h1>

<p>This page has hotpink background color!</p>

</body>
</html>
```



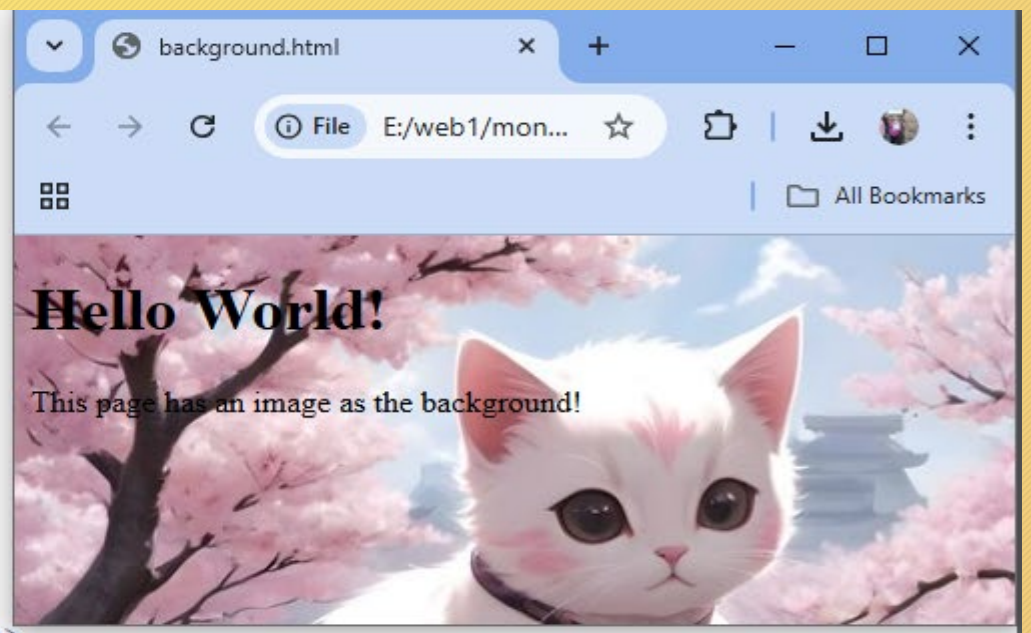
Other elements



CSS Background Image

- specifies an image to use as the background of an element.

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image: url("cat.jpg");
   background-size: cover;
}
</style>
</head>
<body>

<h1>Hello World!</h1>

<p>This page has an image as the background!</p>

</body>
</html>
```



Background Attachment

- property controls whether the background image scrolls or remains fixed with the page's content
- Scroll:allows the background image to scroll with the page (default value)
- Fixed: stops the background image from scrolling with the page
- Initial:sets the property value to default
- Local:allows the background image to scroll with the element's content
- Inherit: inherits the property value from its parent element

CSS Gradient

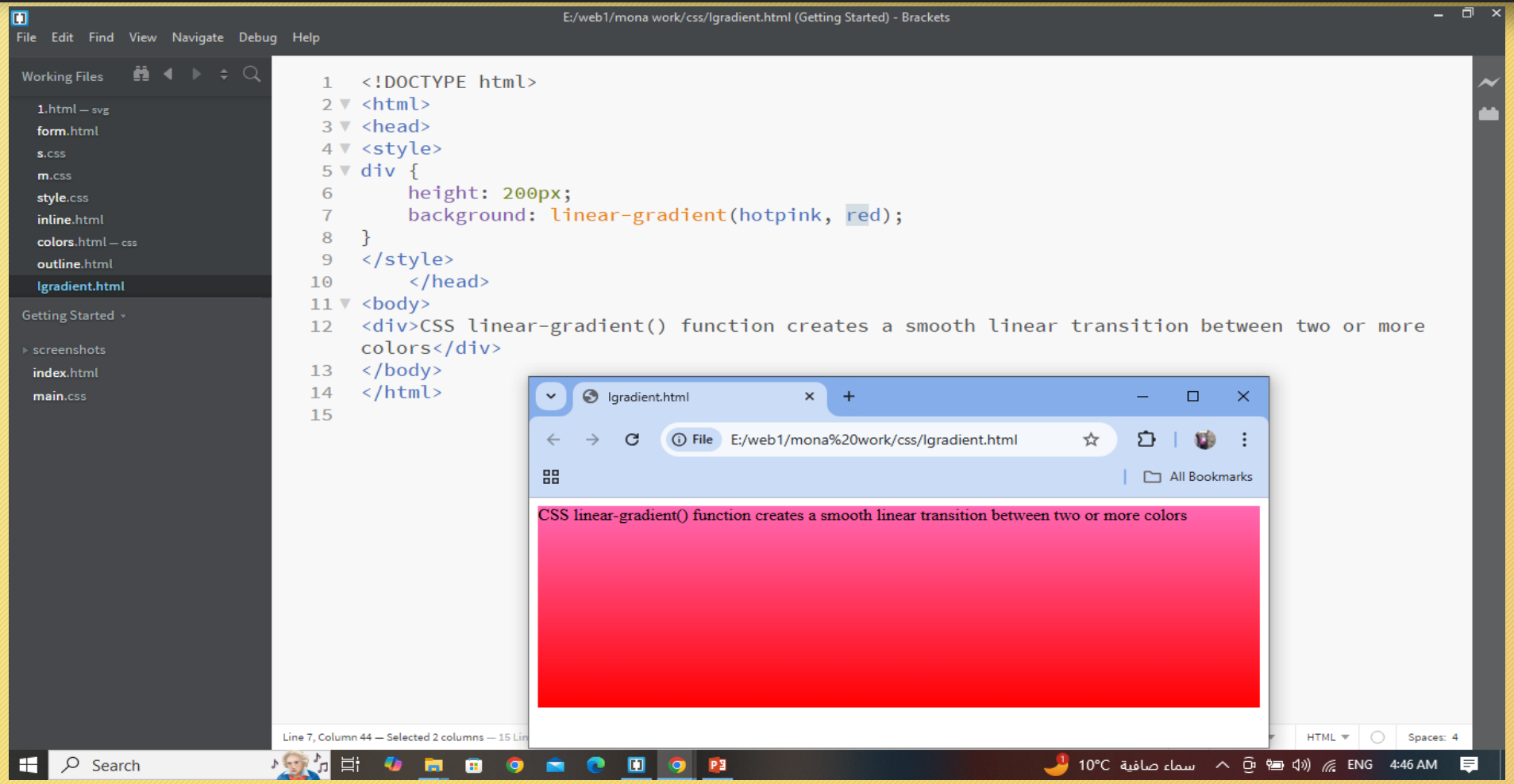
- CSS gradients are the smooth transitions between two or more colors. They are used to decorate the background, borders, and other elements on a webpage.
- There are three types of gradients in CSS,

❑Linear Gradients

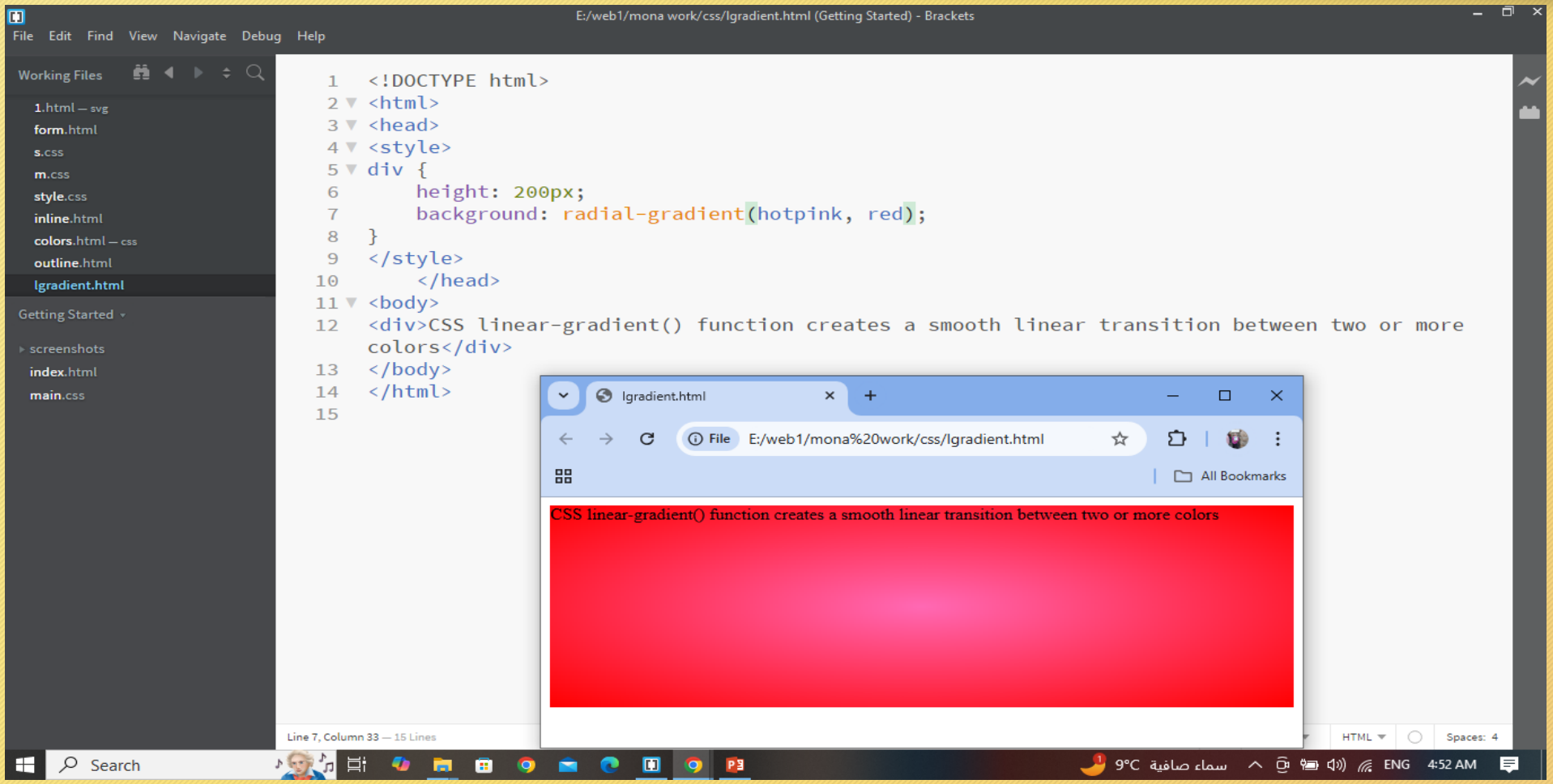
❑Radial Gradient

❑Conic Gradient

CSS linear gradient



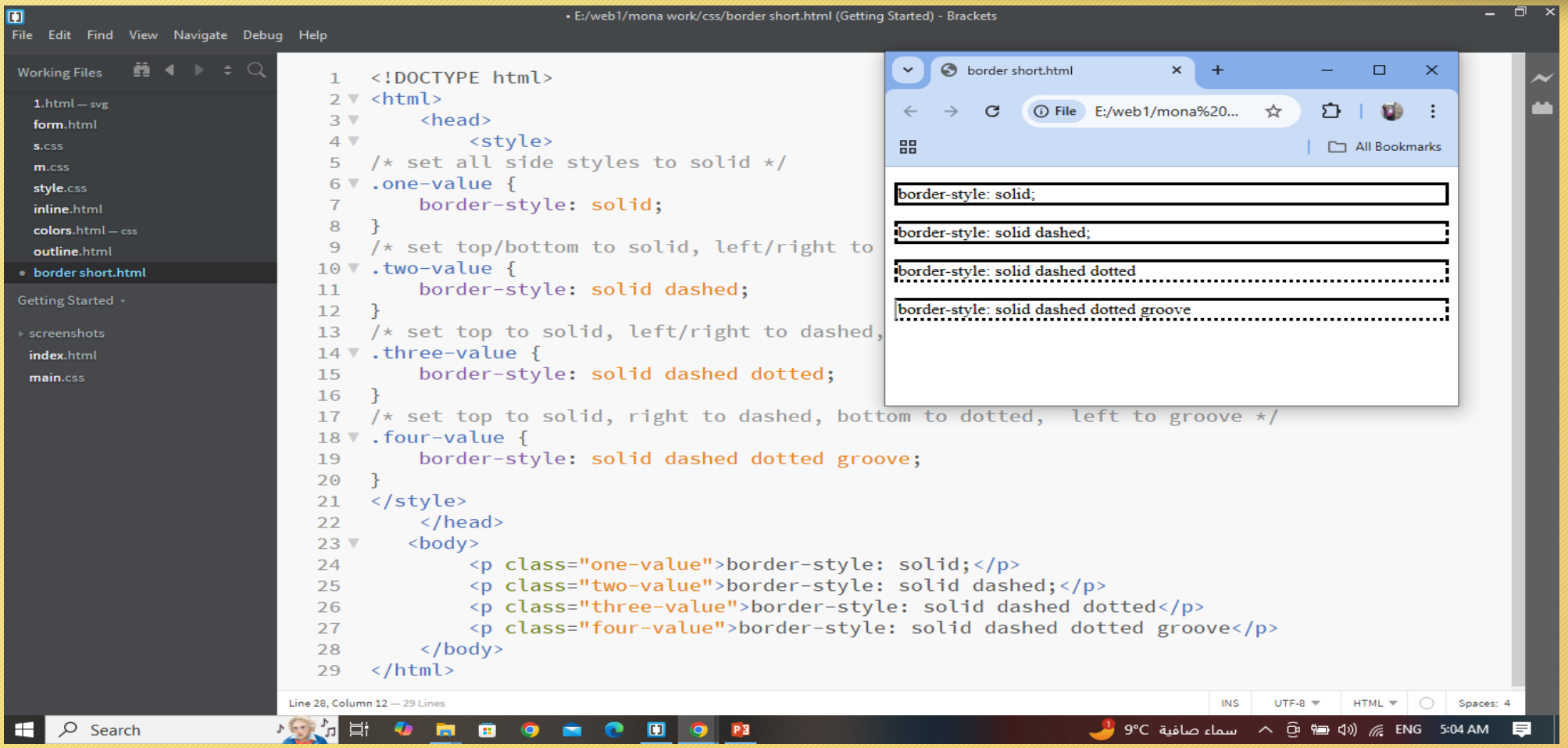
CSS radial gradient



Borders-style values

- `solid` : creates a border with a single solid line
- `dotted` : creates a border with a series of dots
- `dashed` : creates a border with a series of dashes
- `double` : creates a border with two parallel lines
- `groove` : creates a border with a carved-in 3D effect
- `ridge` : creates a border with a raised 3D effect
- `inset` : creates a border with a pushed-in 3D effect
- `outset` : creates a border with a popped-out 3D effect
- `none` : no border is displayed
- `hidden` : the border is hidden

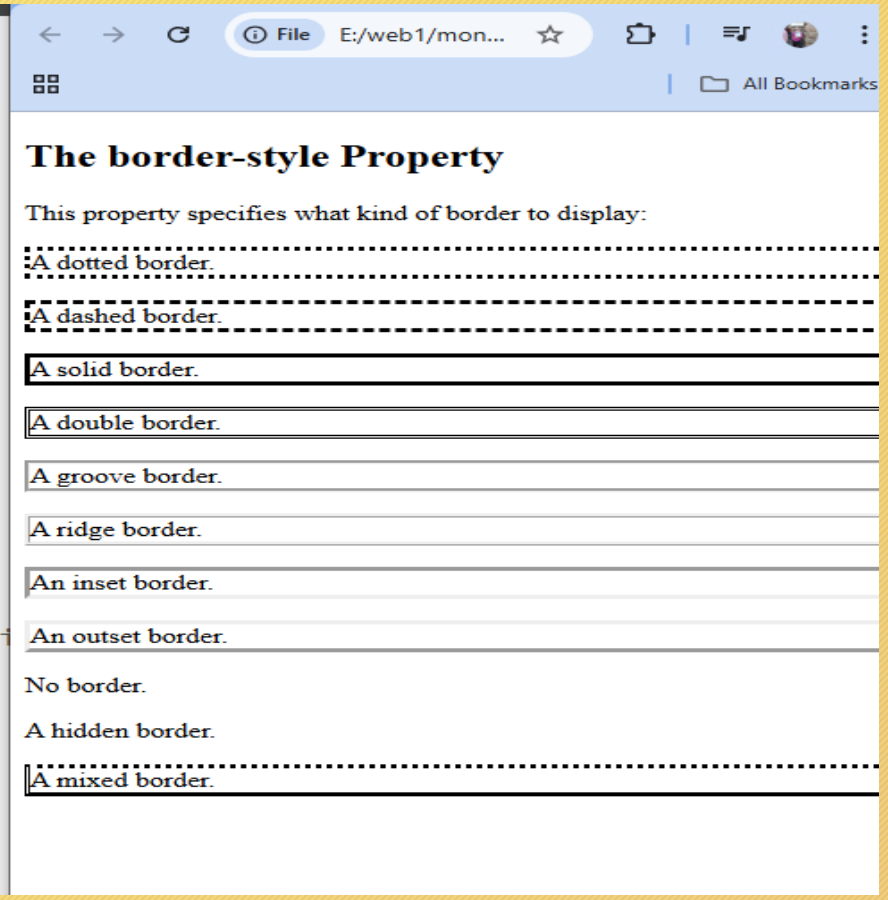
CSS border-style property as a Shorthand Property



Borders

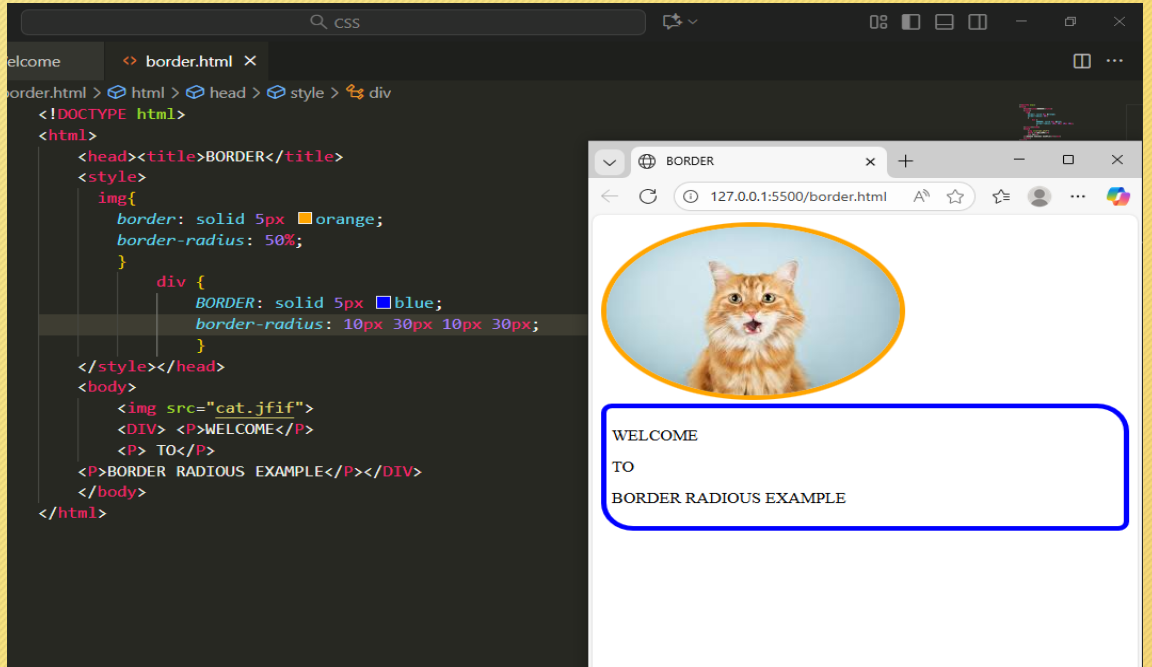
- allow you to specify the style, width, and color of an element's border.

```
<!DOCTYPE html>
<html>
<head>
<style>
.dotted {border-style: dotted;}
.dashed {border-style: dashed;}
.solid {border-style: solid;}
.double {border-style: double;}
.groove {border-style: groove;}
.ridge {border-style: ridge;}
.inset {border-style: inset;}
.outset {border-style: outset;}
.none {border-style: none;}
.hidden {border-style: hidden;}
.mix {border-style: dotted dashed solid double;}
</style>
</head>
<body>

<h2>The border-style Property</h2>
<p>This property specifies what kind of border to di

<p class="dotted">A dotted border.</p>
<p class="dashed">A dashed border.</p>
<p class="solid">A solid border.</p>
<p class="double">A double border.</p>
<p class="groove">A groove border.</p>
<p class="ridge">A ridge border.</p>
<p class="inset">An inset border.</p>
```



Border-radious

- defines how **rounded the corners** of an element are
- Syntax: border-radious: value
- values can be (px in pixels-percentage %)



Box shadow

- The box-shadow property adds a **shadow effect** around an element’s box. You can control its **position, blur, spread, and color.**
- syntax

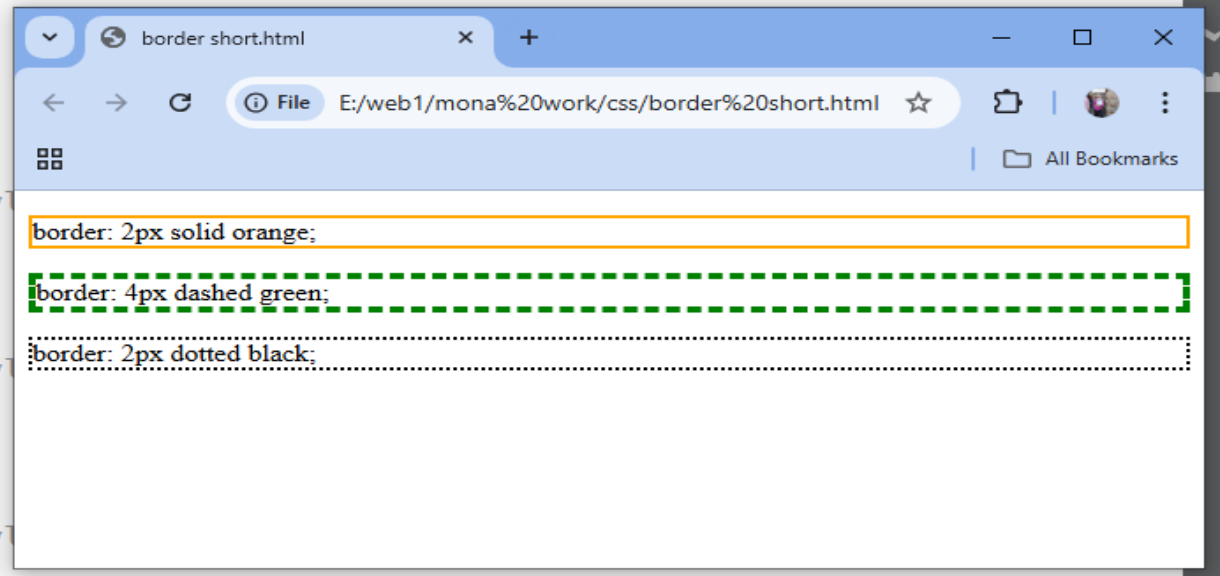
box-shadow: offset-x offset-y blur-radius spread-radius color;

- Offset-x horizontal position right left
- offset-y vertical position up down
- blur

CSS border Syntax

```
border: border-width border-style border-color;
```

```
<!DOCTYPE html>
<html>
    <head>
        <style>
.border-solid {
    /* border-width | border-styl
    border: 2px solid orange;
}

.border-dashed {
    /* border-width | border-styl
    border: 4px dashed green;
}

.border-dotted {
    /* border-width | border-styl
    border: 2px dotted black;
}
}
</style>
    </head>
   <body>
        <p class="border-solid">border: 2px solid orange;</p>
        <p class="border-dashed">border: 4px dashed green;</p>
        <p class="border-dotted">border: 2px dotted black;</p>
    </body>
</html>
```



Border image

```
<!DOCTYPE html>
<html>
    <head>
        <style>
            h1{
                /* adds a solid border, necessary for border image */
                border: solid;
                /* image url | slice width | border-width */
                border-image: url("flower.jpg")40/5px;
            }
</style>
    </head>
   <body>
        <h1> border image</h1>
    </body>
</html>
```