CSS:-

Cascading style sheet

🡪It is language used to describe the style of document.

Usually write in three ways:-

1.inline

2.internal (in head part)

3. external file (.css)

🡪Basic syntax of CSS:- (internal)

**Internal External**

**<style>**

**Selector(tag){ Selector(tag){**

**Property:value; Property:value;**

**} }**

**</style>**

**🡪**to link css file with html we use **<link rel=”stylesheet” href=”css file path”>** in **head part** of html.

🡪Color Property:-

1.foreground= color

2.background-color

<!DOCTYPE html>

<html lang="en">

<head>

    <link rel="stylesheet" href="html1.css">

    <style>

        h1{

        color:red;

    }

    </style>

    </head>

<body>

    <h1>Hello</h1>

    <h1 style="color: blue;">HEllo</h1>

    <p>aditya das</p>

    <button>hello</button>

</body>

</html>



Color system:-

Color: rgb(0,0,0) for adding transparency in it 🡪 color:rgb(0,0,0,0-1)

Range – 0 to 255

p{

    color: yellow;

    background-color: blue;

}

button{

    color: magenta;

    font-family: fantasy;

}

body{

    background-color: black;

}

h3{

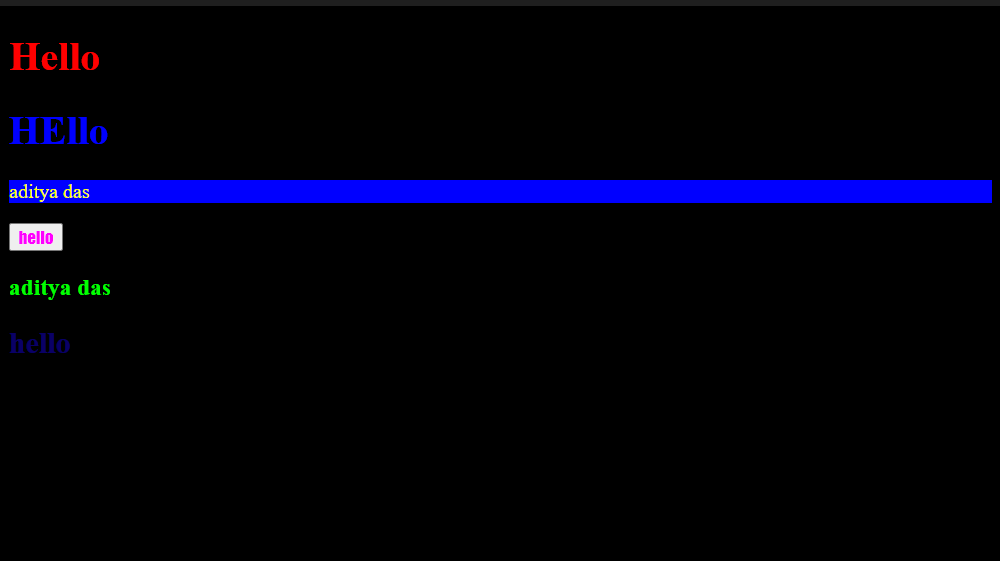
    color: rgb(0,255, 0);

}

h2{

    color: rgb(25,0, 255,0.4);

}



* For color combination we can vist [www.coolors.co](http://www.coolors.co)

Selectors in css:-

* Universal selector \*{} 🡪 \*
* Element selector tagname{}
* **Id selector #name{} 🡪# 🡪id given to tag as <tagname id=”name”> in css we call it as #name**
* **Class selector .name{} 🡪. class given to tag as <tagname class=”name”> in css we call it as .name**
* Priority:- inline>internal>External

\*{

    font-size: larger;

    background-color: rgb(38, 94, 118);

}

h1{

    color: rgb(234, 0, 0);

}

#h1{

    color: rgb(255, 132, 0);

}

.myclass{

    color: yellow;

}

<!DOCTYPE html>

<html lang="en">

<head>

    <link rel="stylesheet" href="html2.css">

    <title>Document</title>

</head>

<body>

    hello aditya das

    <h1>aditya das</h1>

    <h1 id="h1">adity das</h1>

    <h2 class="myclass">aditya das</h2>

    <h3 class="myclass">adity</h3>

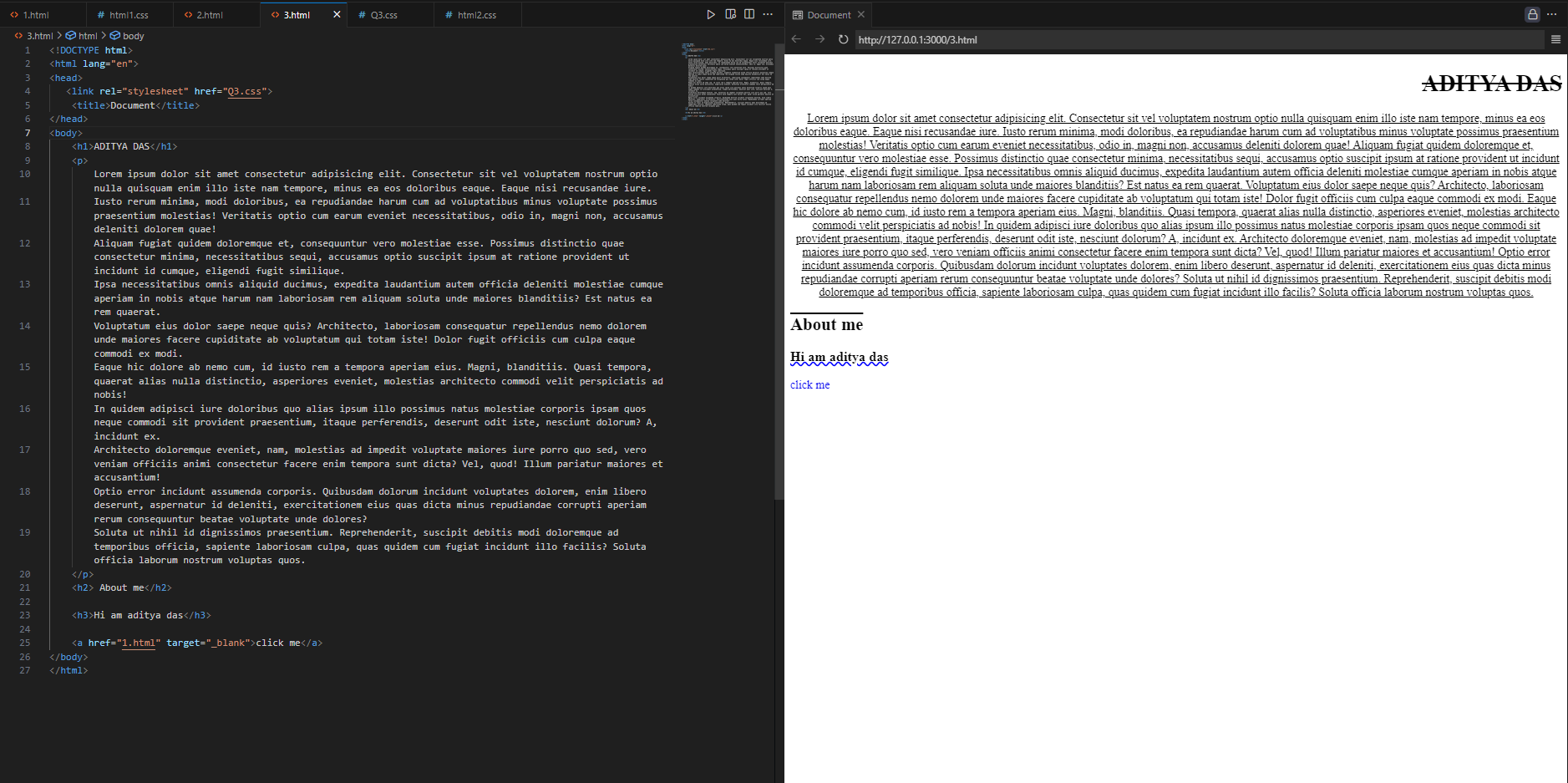
</body>

</html>



Text Property:-

* Text-align:left/right/center
* Text-decoration:underline/overline/line-through/none/wave
* We can refer to mdn website for see other properties of it



* Font-weight (usually done for light and dark text)

🡪normal/bold/bolder/lighter

* Range: 100 -900
* Font-family(usually done font style)

**UNITS in css:-**

* Pixels(px)
* 96px =1 inch or 2.5cm
* Font size: “x px”
* Line-height: “x px” (spaces between lines)
* Text-transform: uppercase/lowercase/capitalize/none

(to transform text in upper or lower or capitalized way)

**CODE:-**

#h1{

    font-weight: bolder;

}

#h2{

    font-weight: 200;

}

#h3{

    font-weight: 400;

    font-family:cursive;

    line-height: 1px;

}

#para{

    font-family:cursive;

    font-size: 20px;

}

#para1{

    text-transform: capitalize;

}

#para2{

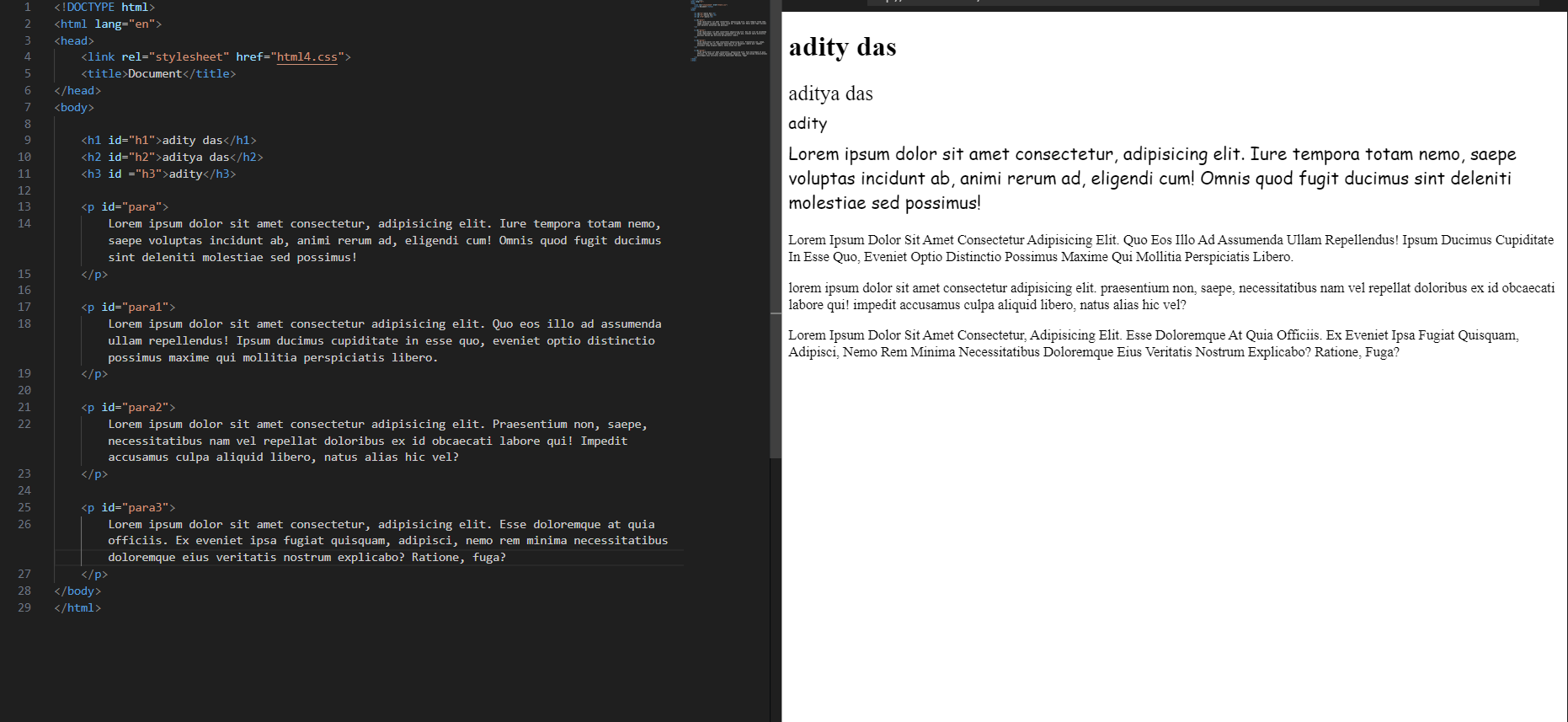
    text-transform: lowercase;

}

#para3{

    text-transform: capitalize;

}



**BOX MODEL IN CSS:-**

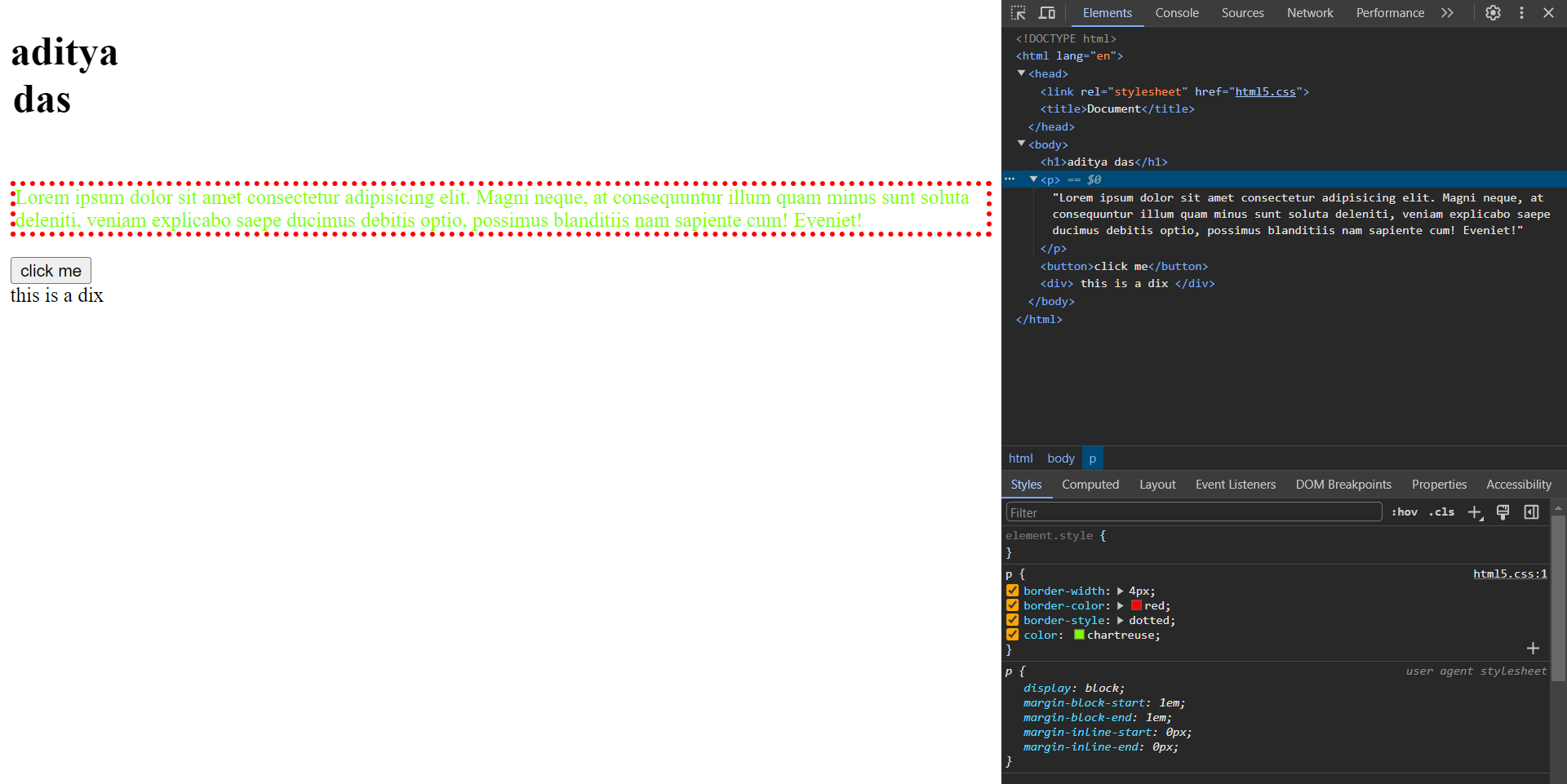
🡪Content:- It contains content part

🡪Padding:- space between border and content

🡪Border:- outter layer of Element

🡪Margin:- Space between two block elements

* Height: to set height in px
* Width: to set width in px
* Border-width: to set border width in px
* Border-style: solid/dotted/dashed
* Border-color: gives color to border



* Border-radius:- It helps to convert border edges to cicular type and written **px.**

CODE:-

p{

    border-width: 4px;

    border-color: red;

    border-style: outset;

    color: chartreuse;

    border-radius: 50px;

}

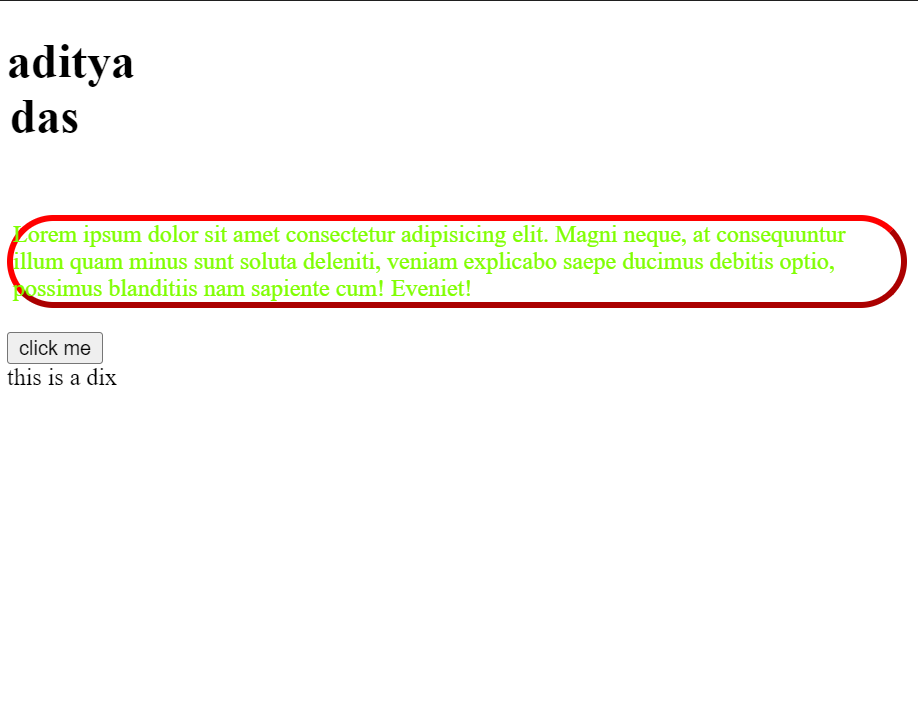
h1{

    height: 100px;

    text-align: center;

    width: 50px;

}



**Padding in css:-**

🡪space between content and border.

* Padding-left/right/top/bottom: px
* Padding: top px right px bottom px left px (clockwise)

**CODE:-**

#adi{

    border: 2px solid black;

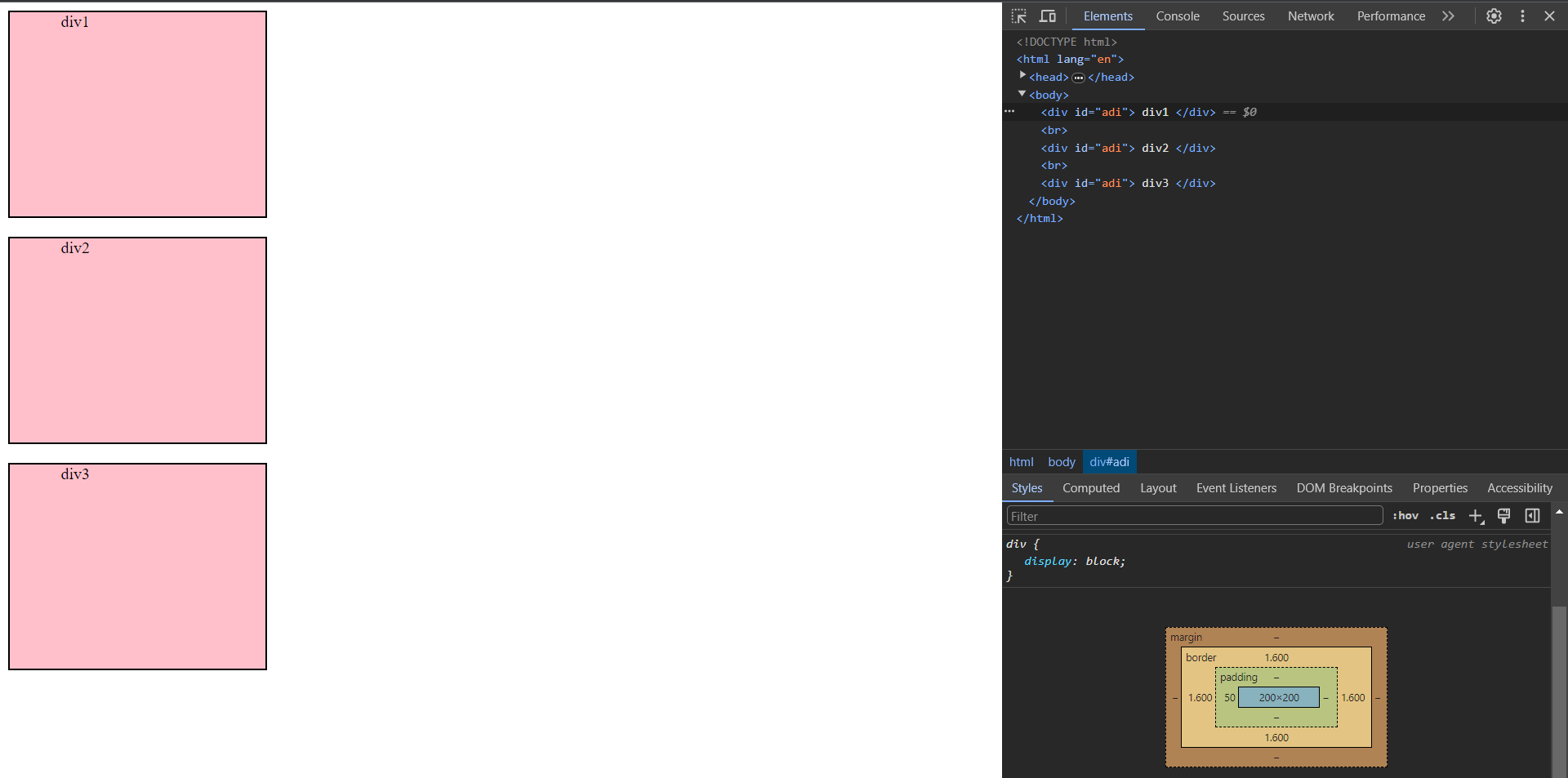
    height: 200px;

    width: 200px;

    background-color: pink;

    padding-left: 50px;

}



**Margin in CSS:-**

**🡪**Space between two box element.

* Margin-top/bottom/right/left: px
* Margin: 4 diections with px(clockwise)

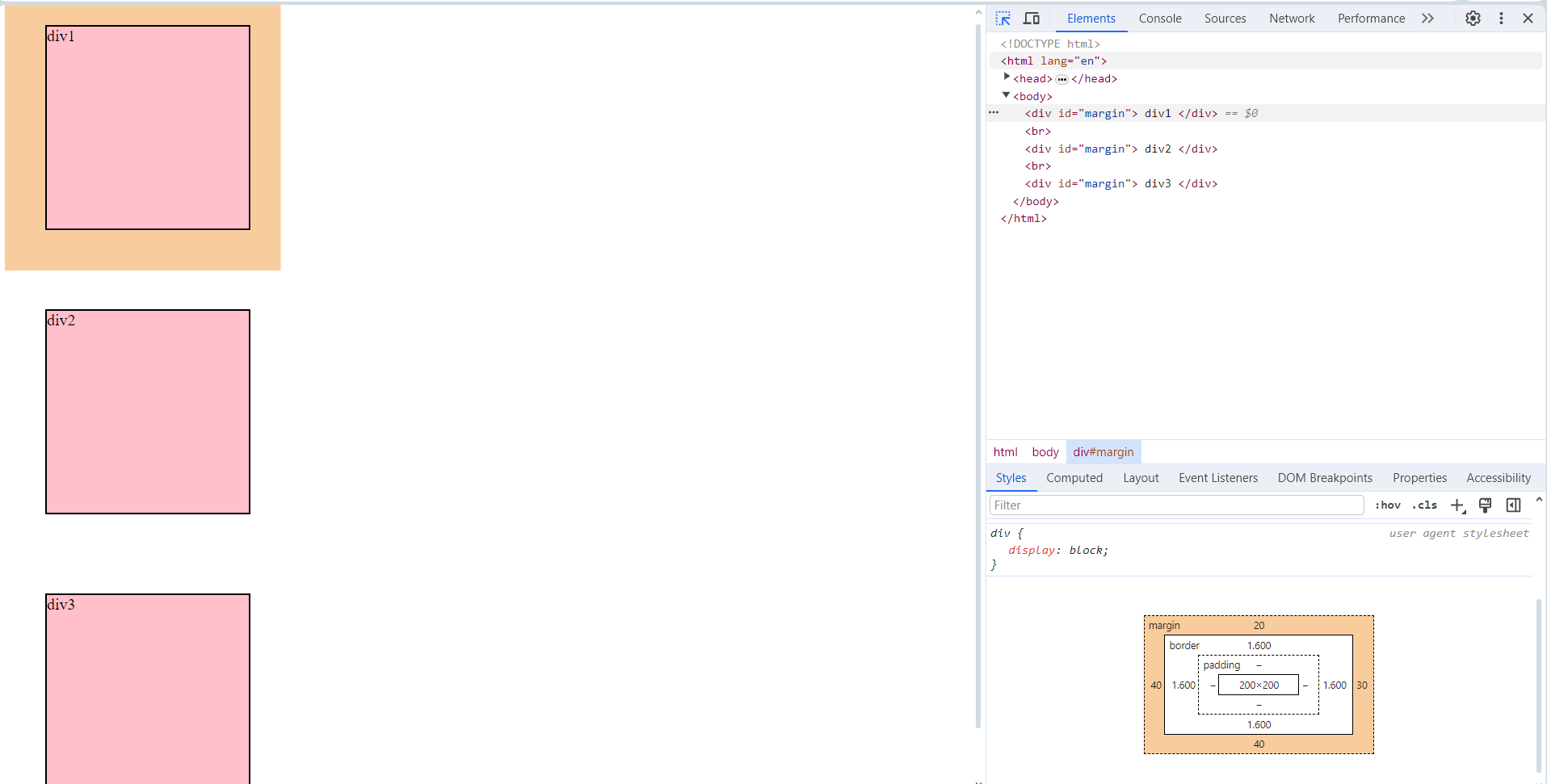
#margin{

    border: 2px solid black;

    height: 200px;

    width: 200px;

    background-color: pink;



    margin: 20px 30px 40px 40px;

}

🡪Output:-

Display Property:-

* Display: inline/block/inline-block/none

🡪inline elements takes space upto they need. E.g.<span>, <a> ,<button>

🡪block element takes whole space. E.g. <div>, <header>

🡪inline-block:- Used to add padding and margin to block elements

🡪none:- to remove the visibility and reserve the space for it

#div1{

    background-color: rgb(0, 255, 47);

}

#span1{

background-color: rgb(255, 0, 0);

}

#div2{

    display: inline;

    background-color: rgb(64, 255, 0);

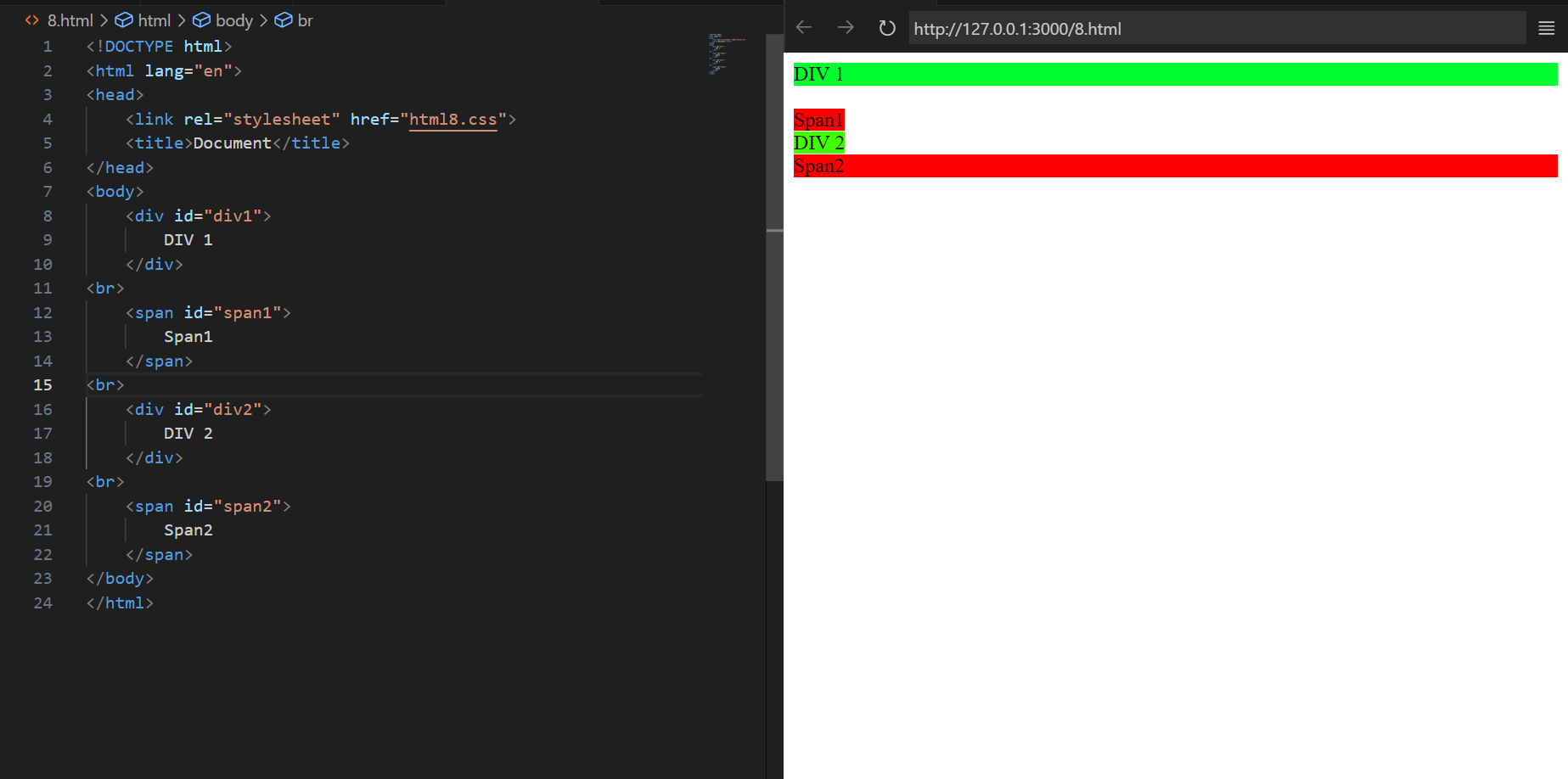
}

#span2{

    display: block;

background-color: rgb(255, 0, 0);

}



**Code:-**

<!DOCTYPE html>

<html lang="en">

<head>

    <link rel="stylesheet" href="html8.css">

    <title>Document</title>

</head>

<body>

    <div id="div1">

        DIV 1

    </div>

<br>

    <span id="span1">

        Span1

    </span>

<br>

    <div id="div2">

        DIV 2

    </div>

<br>

    <span id="span2">

        Span2

    </span>

<br>

<div id="box1">box1</div>

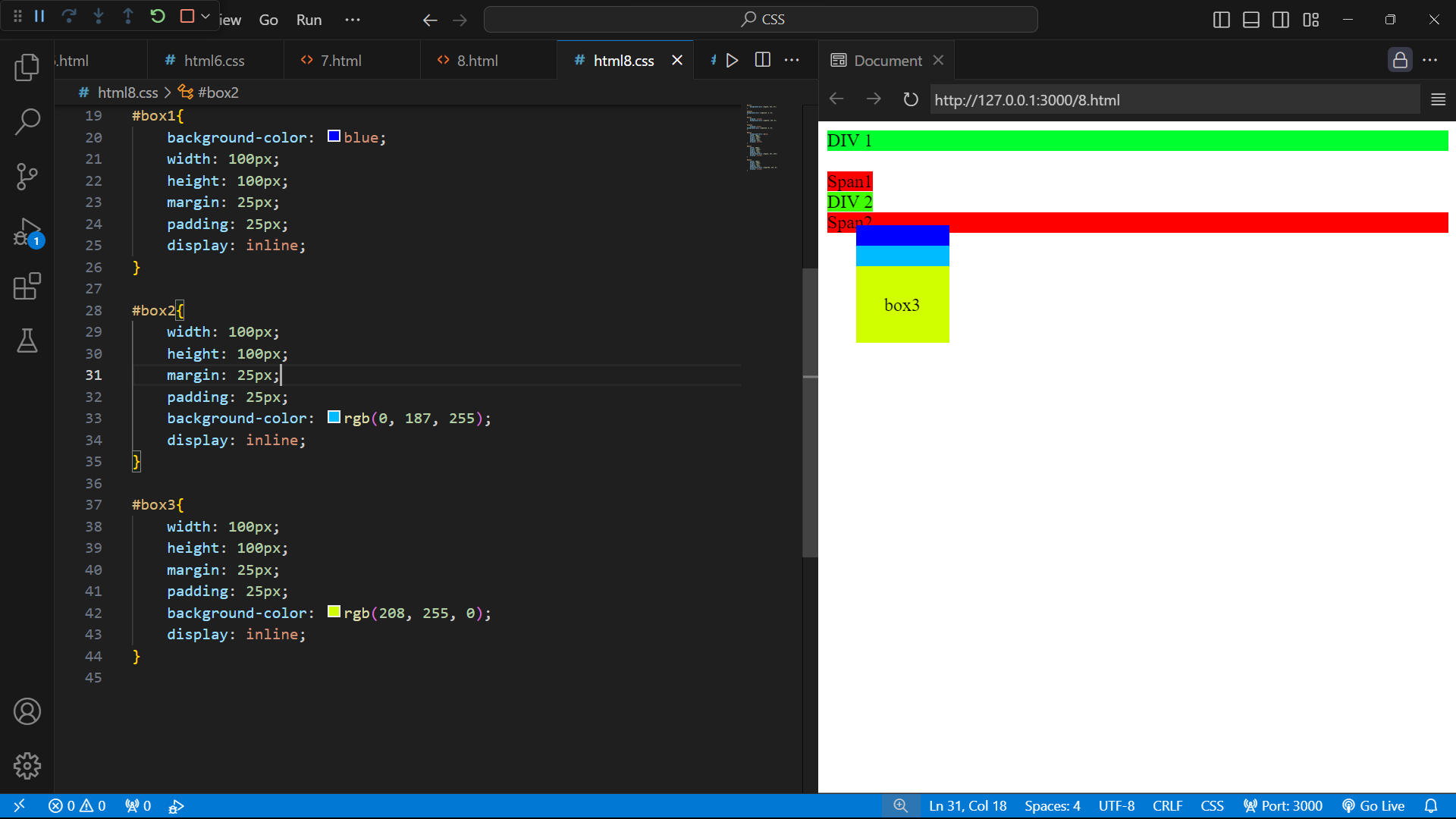
<br>

<div id="box2">box2</div>

<br>

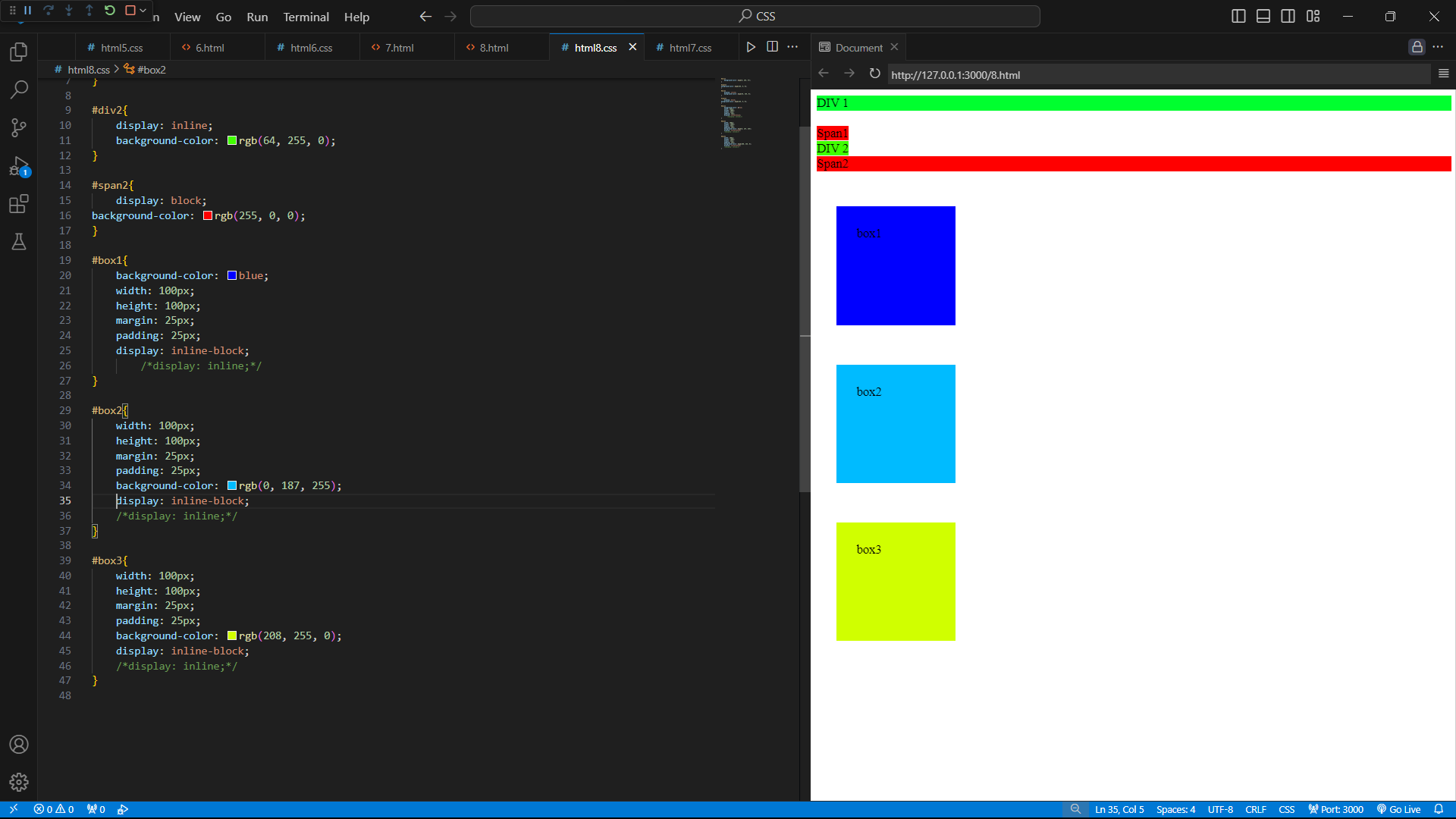
<div id="box3">box3</div>

</body>



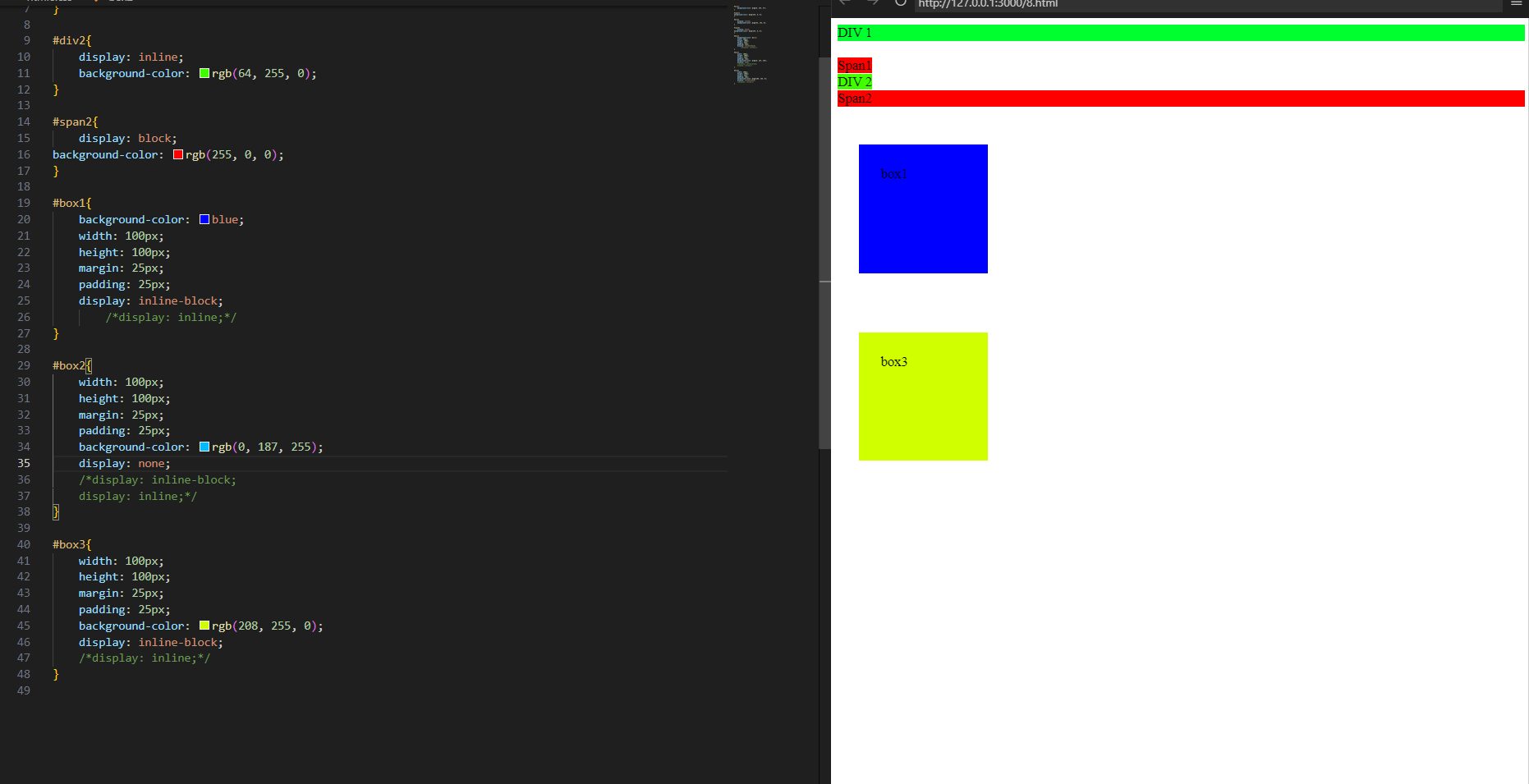
</html>

🡪while using diplay property



It reserves the space for none

property in it.



VISIBILITY (used to hide or display the element):-

* Visibility:none/visible

Alapha channel:-

* Color: rgba(r,g,b,a)

🡪a value ranges from 0.0 to 1.0 more the value high more visible color on color(font-end) or backgroundcolor.

Units in css:-

* %:-

🡪Define a size as relative to an element’s parent object. E.g nested html tags

e.g

🡪<div1><div2></div2></div1> (here div1 is parent and div2 is child) if we give height in % then its x% of div1 tag.

Code:-

<!DOCTYPE html>

<html lang="en">

<head>

    <link rel="stylesheet" href="html9.css">

    <title>Document</title>

</head>

<body>

    <div id="box1">

    <div id="box2">Box2</div>

    <br><br><br><br><br><br><br>

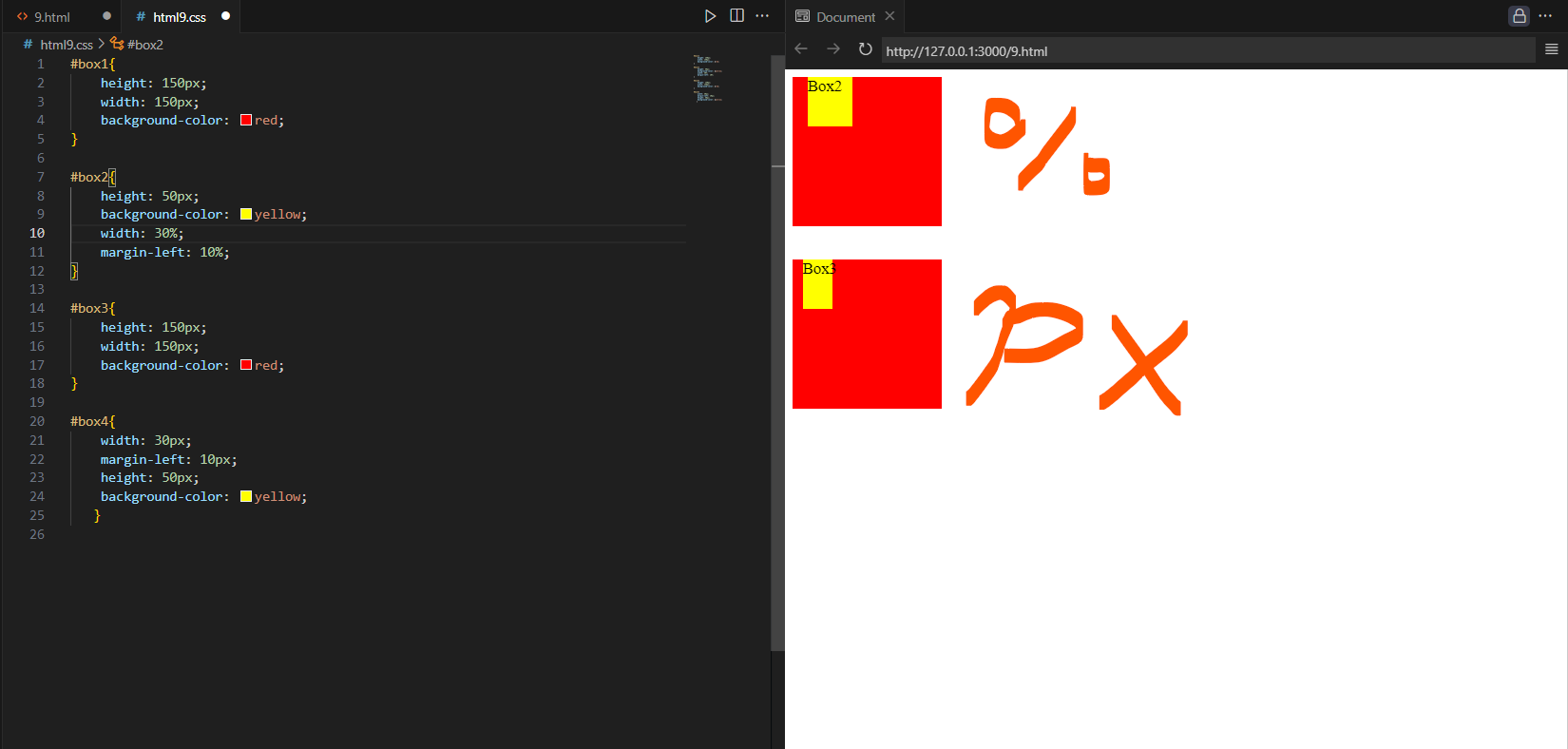
    <div id="box3">

        <div id="box4">Box3</div>

</div>

</body>

</html>



* em:-

🡪relative to parent font size to child size.

e.g. 100px=1em (only for height)

* rem (root element):-

🡪size is set as per root element means <html> tag.

#box7{

    height: 100px;

    width: 200px;

    background-color: rgb(225, 0, 207);

}

#box8{

    margin-left: 20px;

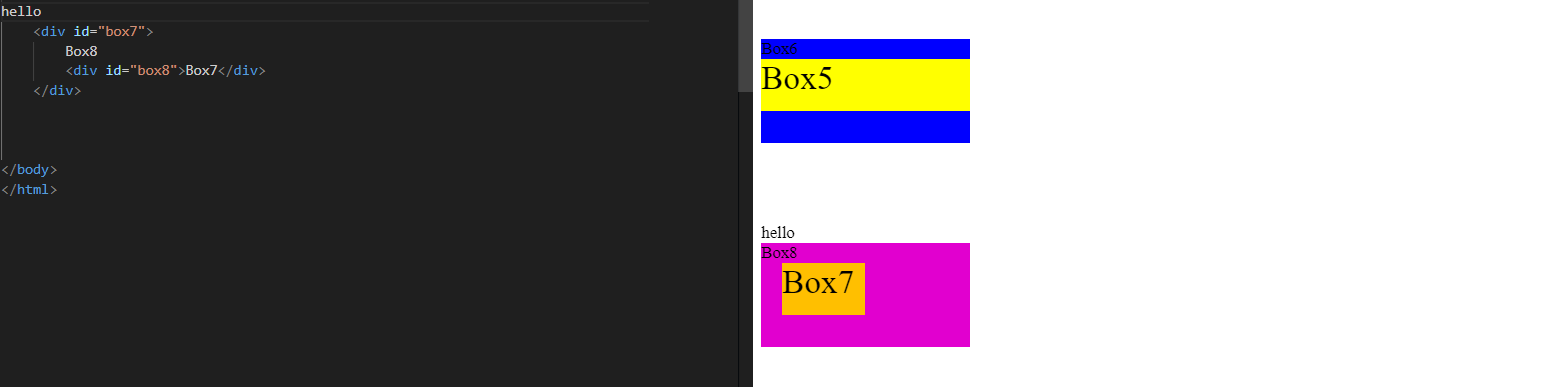
    font-size: 2em;

    height: 50px;

    background-color: rgb(255, 191, 0);

    width: 5rem; /\*16X5=80px here\*/

}



Position Property:-

* It tells how an elelment is positioned in a document.

🡪position: static/relative/absolute/fixed/sticky

* Static🡪its by default
* Relative🡪 When element relative to its self means its moves from its place.
* Top/boootom/left/right/z-index
* Absolute🡪ignoring the parent tag from it follow it relative to body tag.
* Fixed🡪positioned from browser.

Code:-

<!DOCTYPE html>

<html lang="en">

<head>

   <link rel="stylesheet" href="html10.css">

    <title>Document</title>

</head>

<body>

    <h1>Position</h1>

        <div id="box1">box1</div>

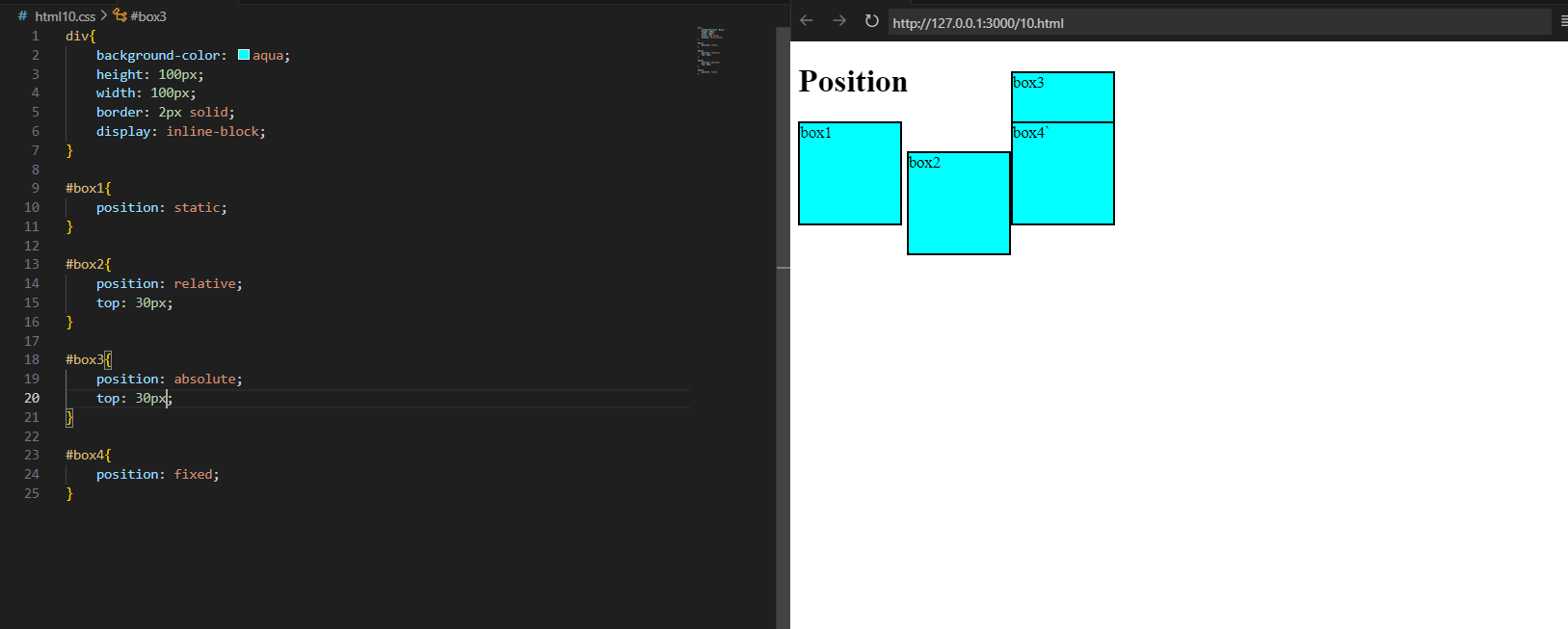
        <div id="box2">box2</div>

        <div id="box3">box3</div>

        <div id="box4">box4`</div>

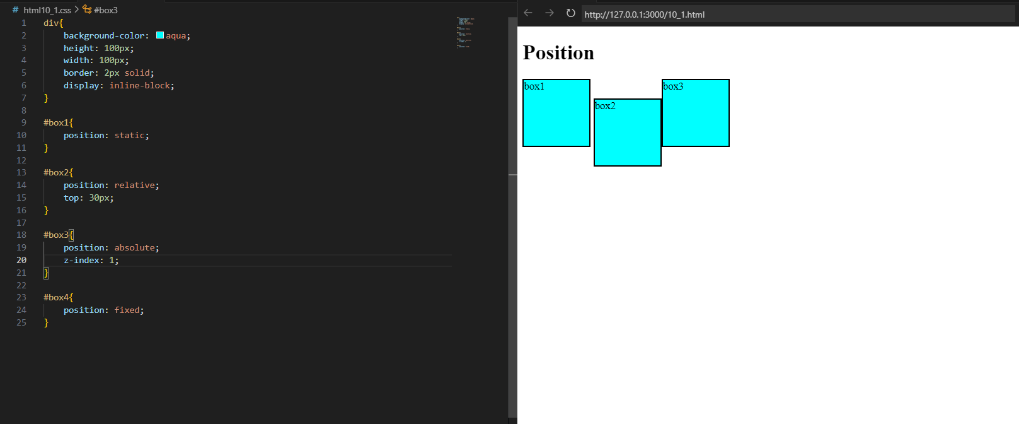
</body>

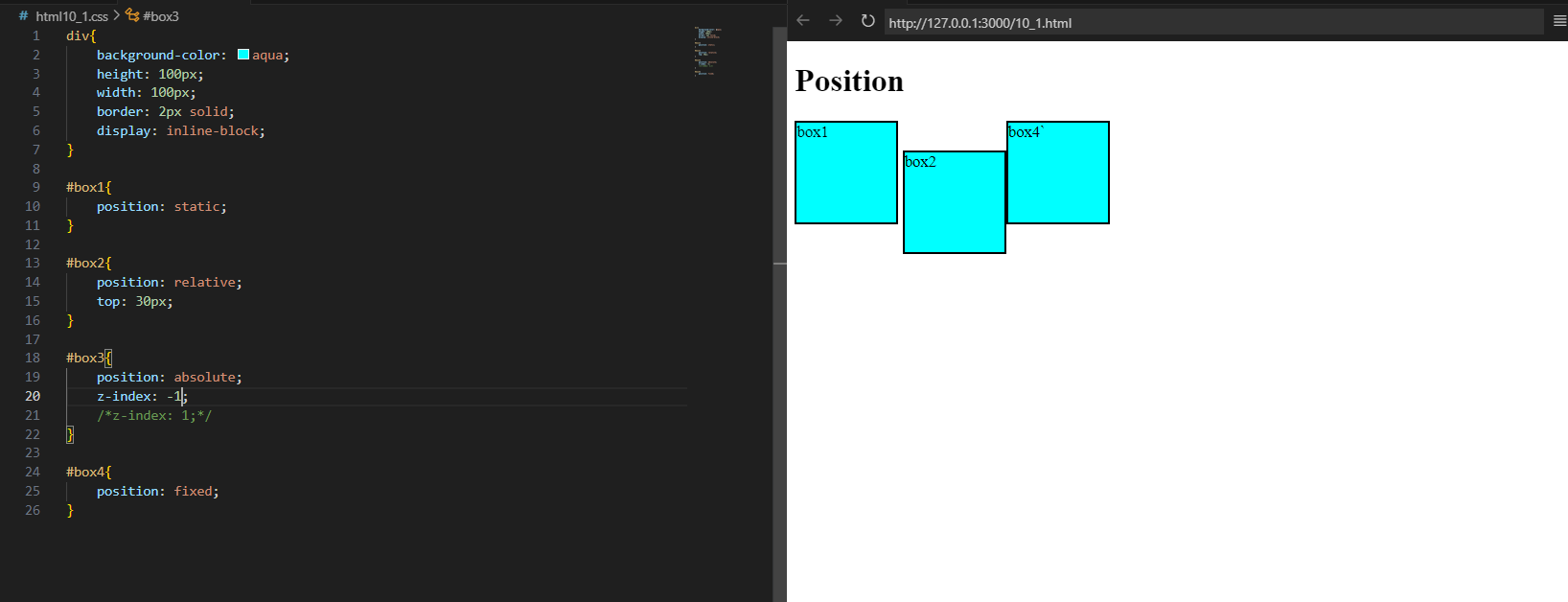
</html>



* When overlapping of elements we decide which element will come top and other will bottom, to do this we use z-index.

-2<-1<0<1<2<3<4



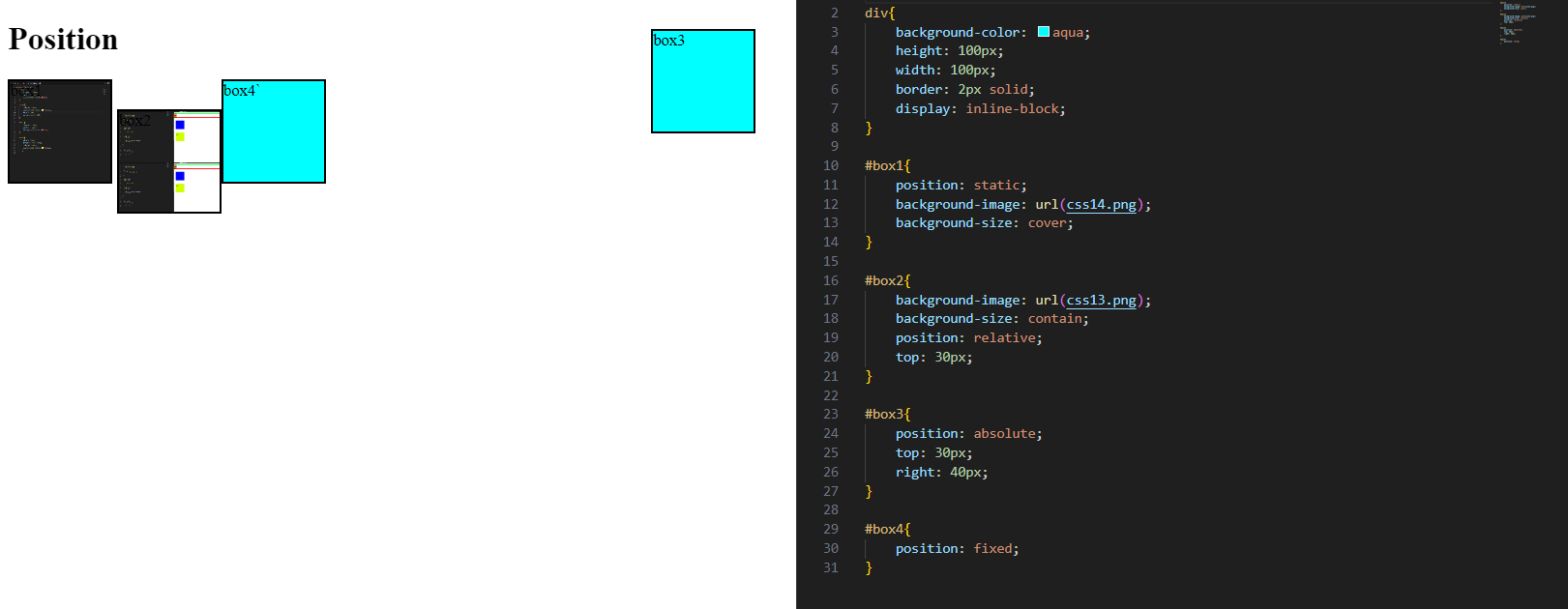


Background-Image:-

🡪to set background image in webpage.

🡪background-size: cover/contain/auto

* Cover=fits the image
* Contain=it enlarge the image
* Auto= It set by default



**FlexBox:-**

* It’s a one dimensional method for arranging items in **rows** or **columns.**

**🡪main axis and cross axis will perpendicular to each other**

**🡪**display: flex;

**🡪**To use flexbox property we need a container and items in it.

* Flexbox direction:-
* It set how flex items are placed in the flex container,along axis and direction.

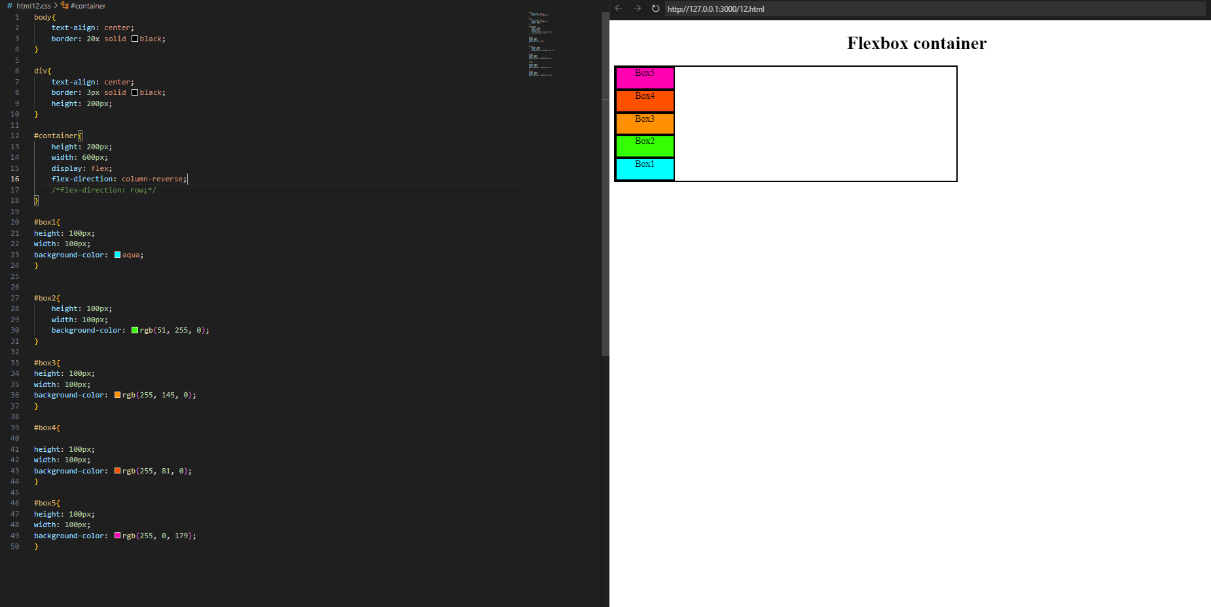
🡪flexbox-direction: row (default)/row-reverse/column/ column-reverse

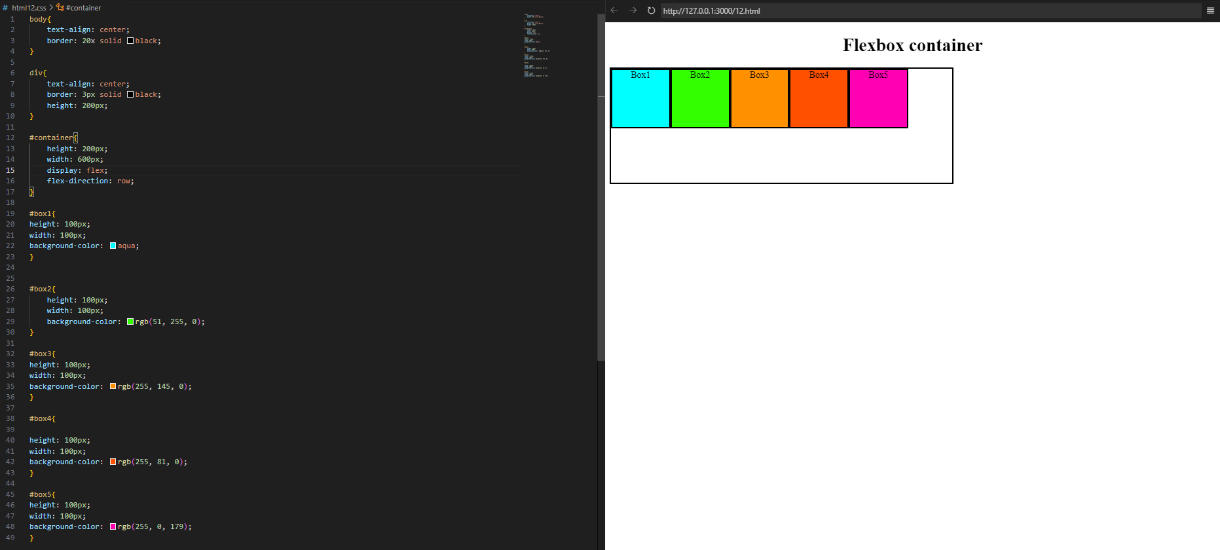
🡪row:left to right on left side

🡪row-reverse: reverse of it from right side

🡪column: top to bottom

🡪column-reverse: top to bottom but in reverse





* Flex Properties:-(for container){ display:flex;}
* **Justify-content**: flex-start(left side)/flex-end(right side)/centre (middle)/space-evenly (space between margin and border of containes)/space-between(space between elements)

🡪alignment along the main axis

Code:-

<!DOCTYPE html>

<html lang="en">

<head>

<link rel="stylesheet" href="html13.css">

    <title>Document</title>

</head>

<body>

    <h1>Flexbox container</h1>

    <div id="container">

        <div id="box1">Box1</div>

        <div id="box2">Box2</div>

        <div id="box3">Box3</div>

        <div id="box4">Box4</div>

        <div id="box5">Box5</div>

    </div>

</body>

</html>



* **Flex-wrap**: nowrap/wrap/wrap-reverse (takes to end)
* **Align-items**: alignment along the cross axis of items

🡪 align-item: center/front-end/front-start/top/bottom

* **Align-content**: alignment of space between & around the content moves all content to respective direction.

🡪 align-content: center/ front-end/front-start /top/bottom

**MEDIA QUERIES:-**

🡪Help to create a website responsive means it fit in all aspect of ratio of screens.

* To see website in different devices with different screen sizes.
* @media (width: px){

Tagname(changes){

change which we want in css}

}

🡪(min-width:px) 🡪used to tell minimum width for show changes

🡪(max-width:px) 🡪used to tell maximum width for show changes

🡪(width:px)🡪used to tell exact width for show changes

**Code:-** div{

    height: 50px;

    width: 50px;

    background-color: pink;

}

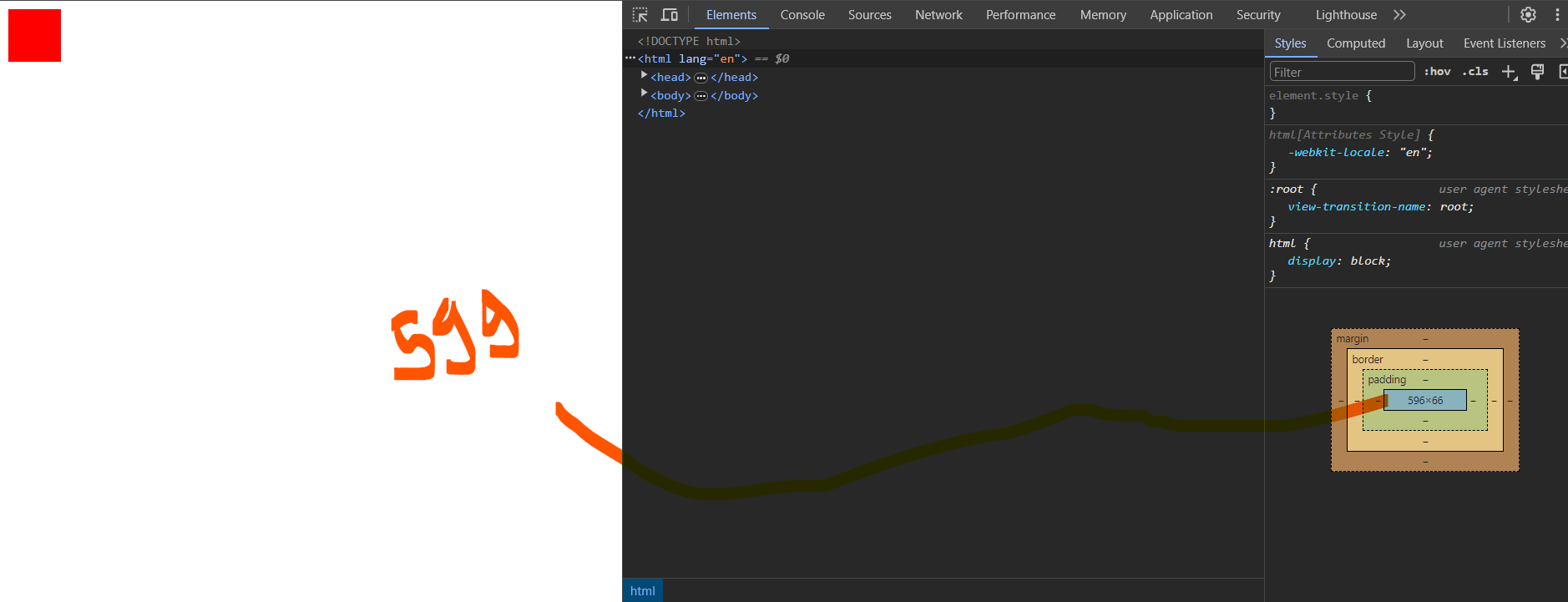
@media (max-width:600px) {

    div{

        background-color: red;

    }

}



* We can can define range for width as

🡪(min-width:px) and (max-width:px)

**CODE:-**

div{

    height: 50px;

    width: 50px;

    background-color: pink;

}

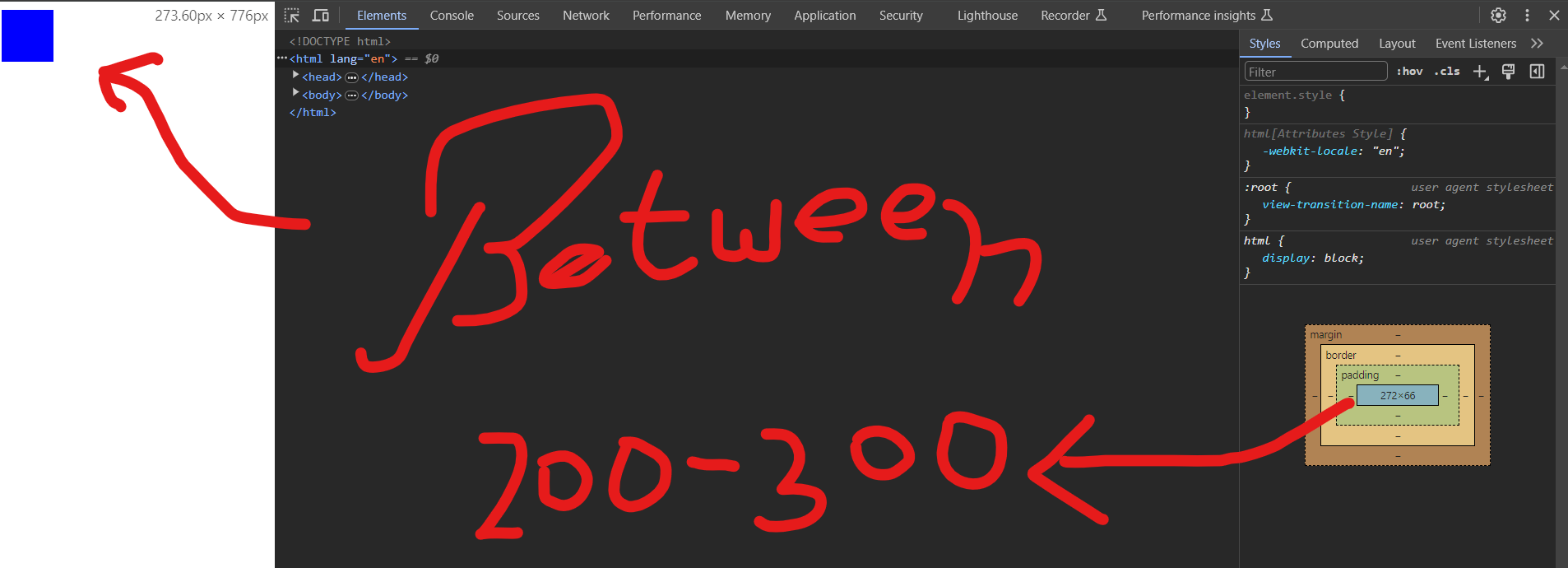
@media(min-width:200px) and (max-width:300px){

    div{

        background-color: blue;

    }

}

****

**ADVANCED CSS (15)**

Transitions property:-

* It means converting a element to one state to another state.
* To do this we need to declare a psedoclass

🡪syntax

**Element\_name:action{**

**Changes using css**

**}**

Action= hover/focus/active/visited/focus-within/focus-visible/target

🡪hover=means go to the element

🡪active=click on element

(To search about more action we can go to www.mdn.com)

div{

    height: 100px;

    width: 100px;

    background-color: aqua;

    border: 3px solid black;

}

div:hover{

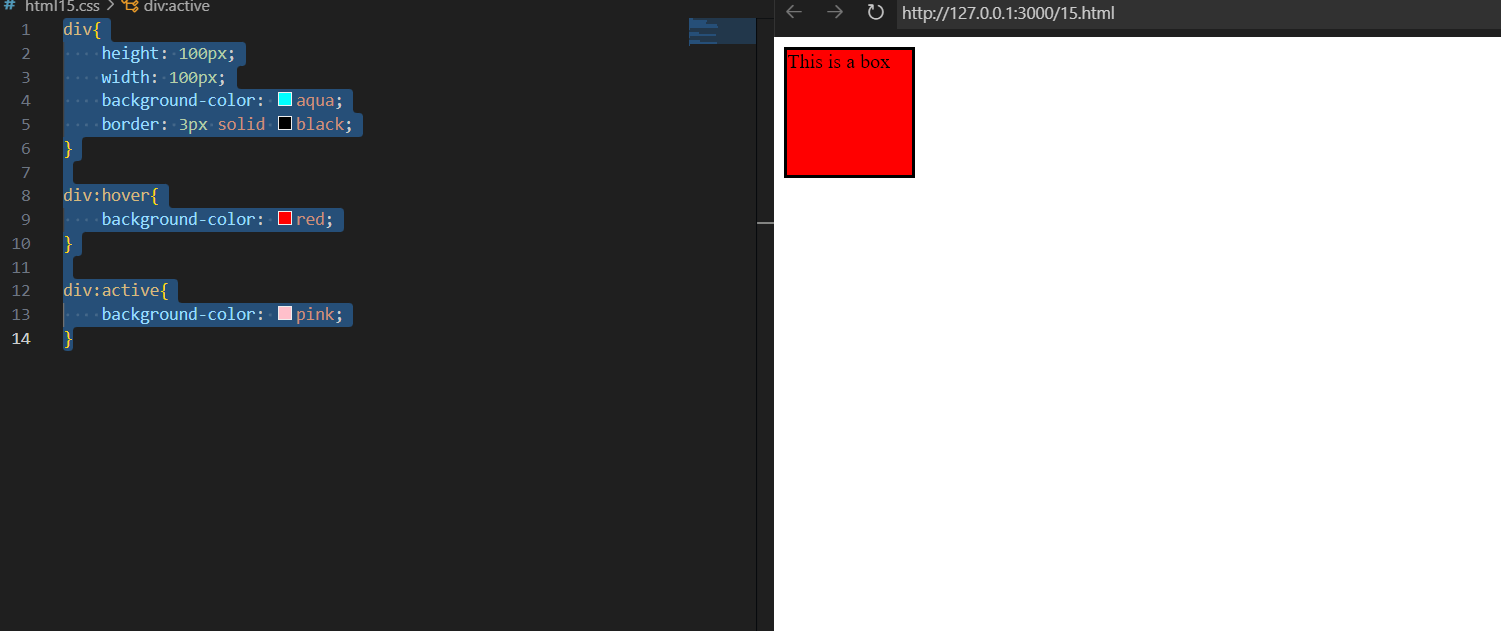
    background-color: red;

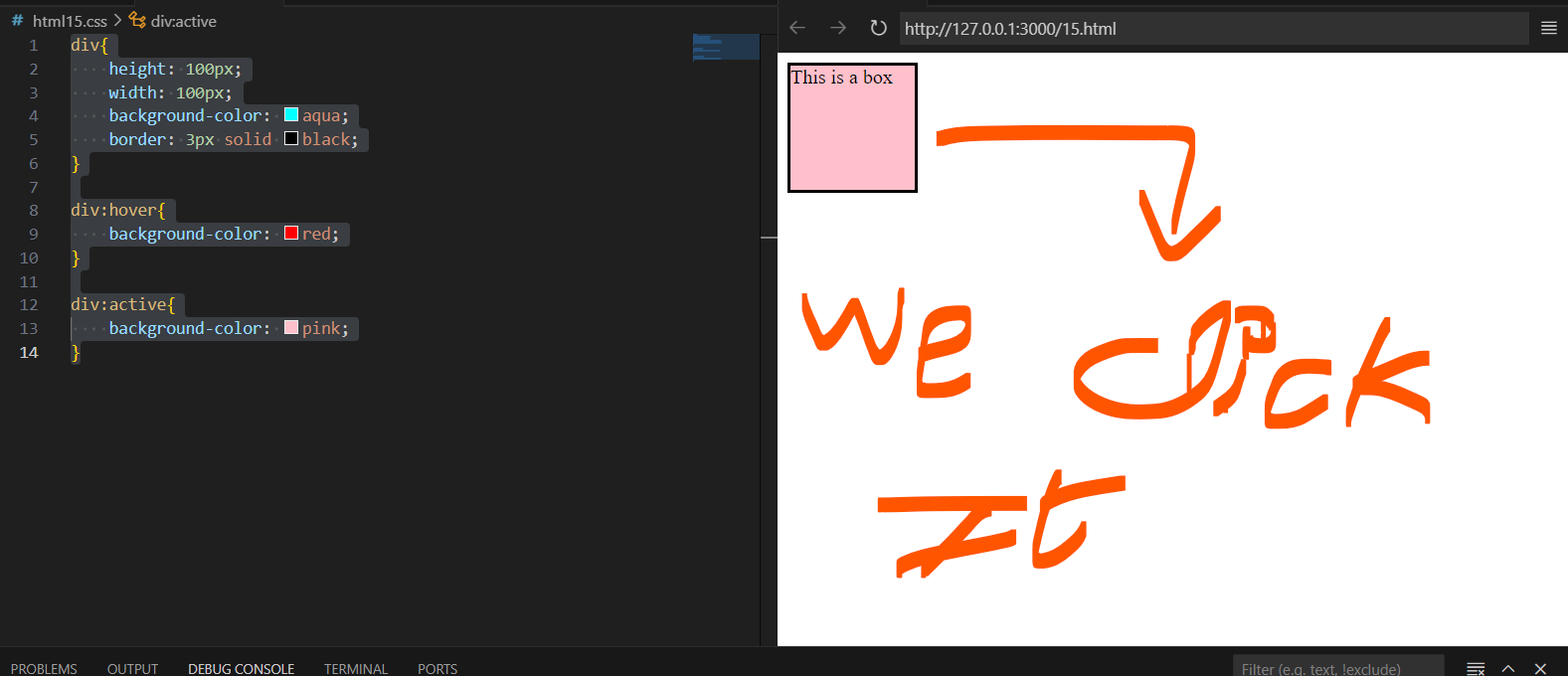
}

div:active{

    background-color: pink;

}

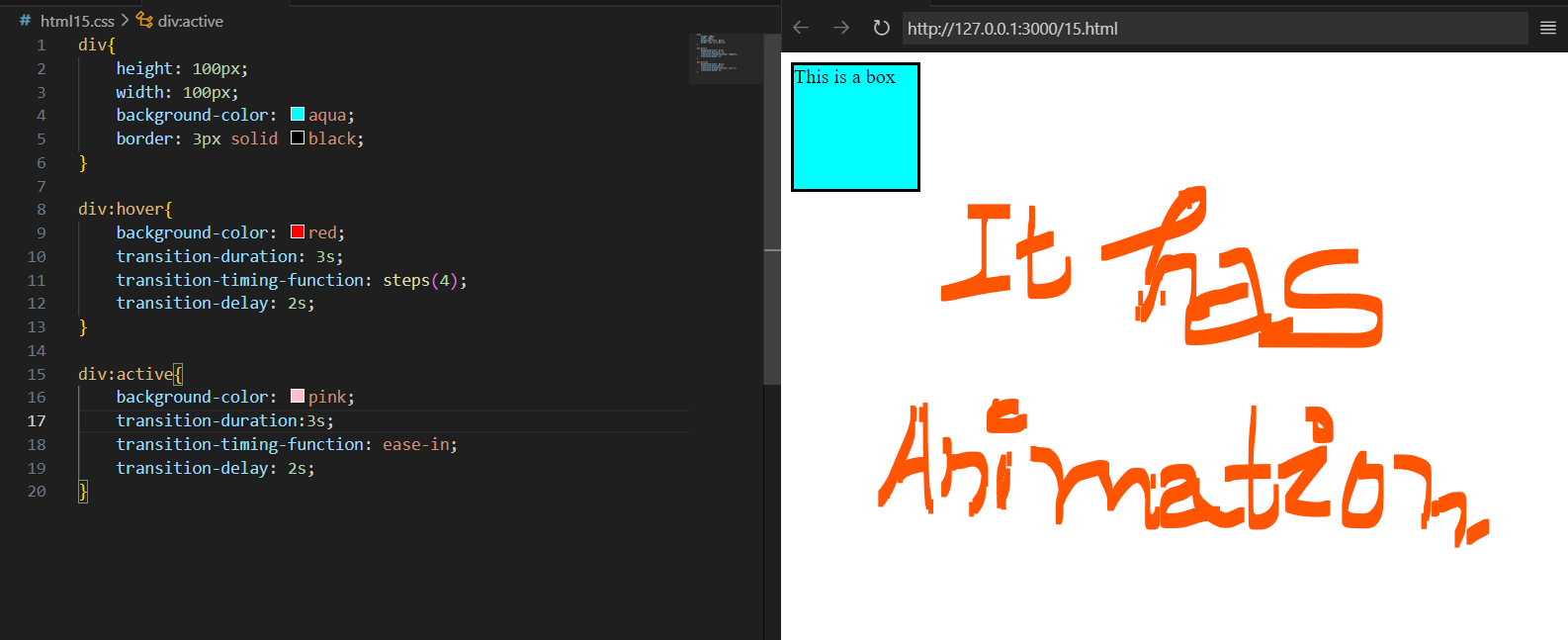




* Transition property:property you want to transition(font-size,width,etc.)

🡪Main

* Transition-duration: 2s/4ms..
* Transition-timing-function: ease-in/ease-out/linear/steps ..
* Transition-delay:2s/4ms..



Transition shorthand technique:-

**Transition: property-name duration timing-function delay**

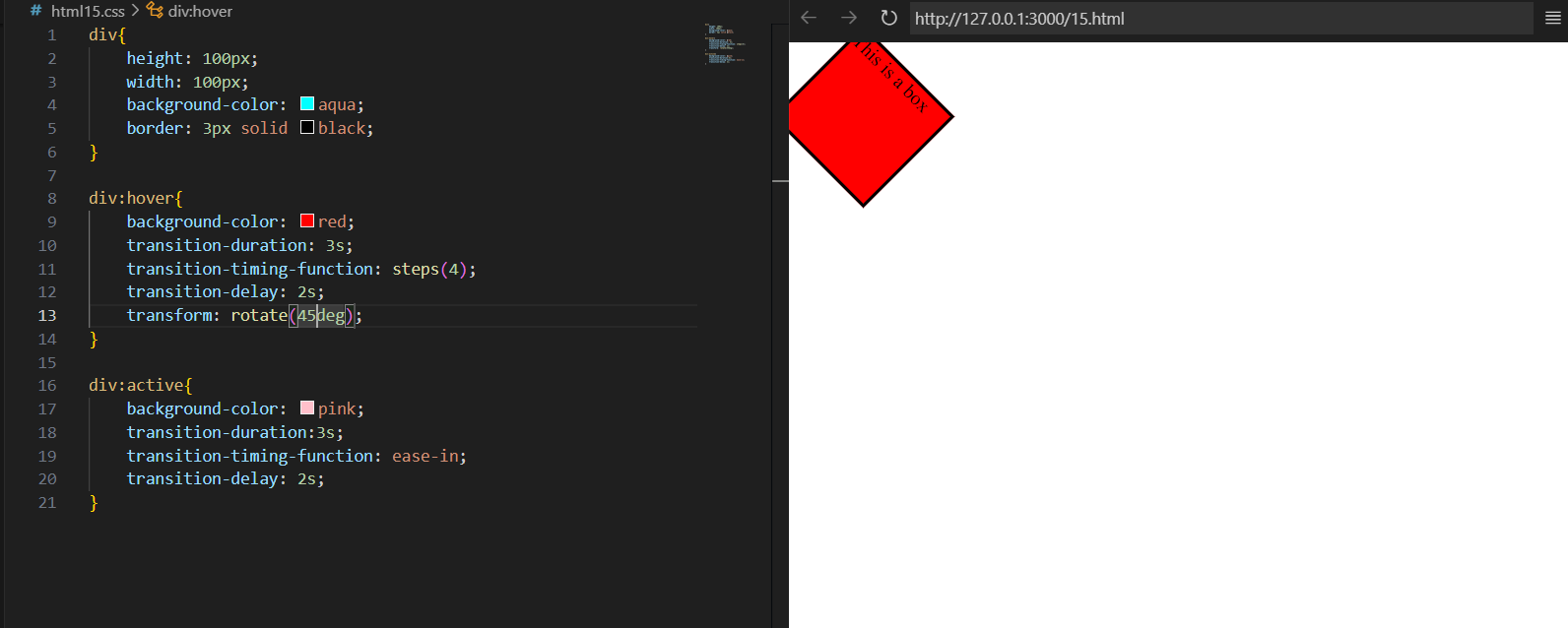
* Property name: background-color,font-family,etc
* Duration: 2 sec
* Timing-function: ease-in-out,etc
* Delay: 2s,4ms

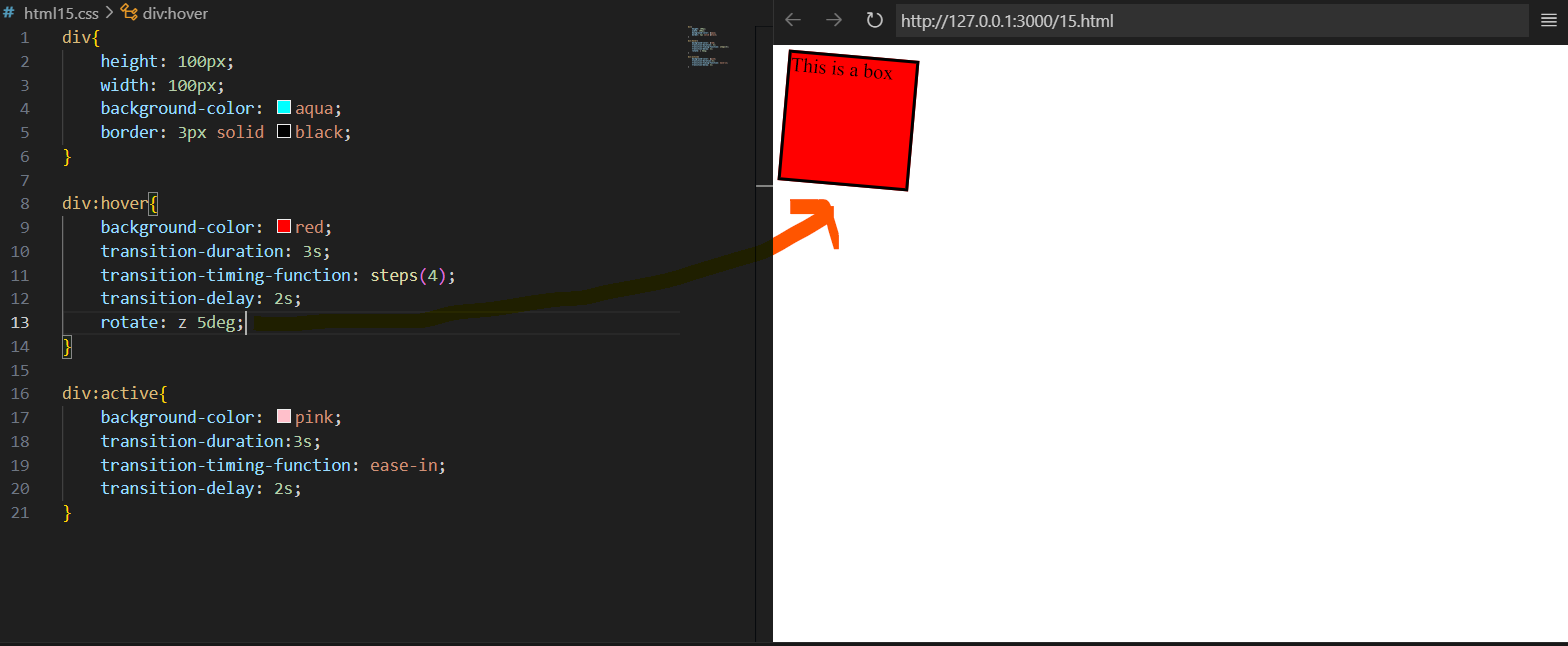
**CSS Transform:-**

🡪used to apply 2d and 3d transformation to an element

* **Transform:rotate(n deg);**

**🡪Rotate: x/y/z ndeg; (where n is number)**





* **Scale:-**

🡪used to increase or decrease the size

🡪syntax:

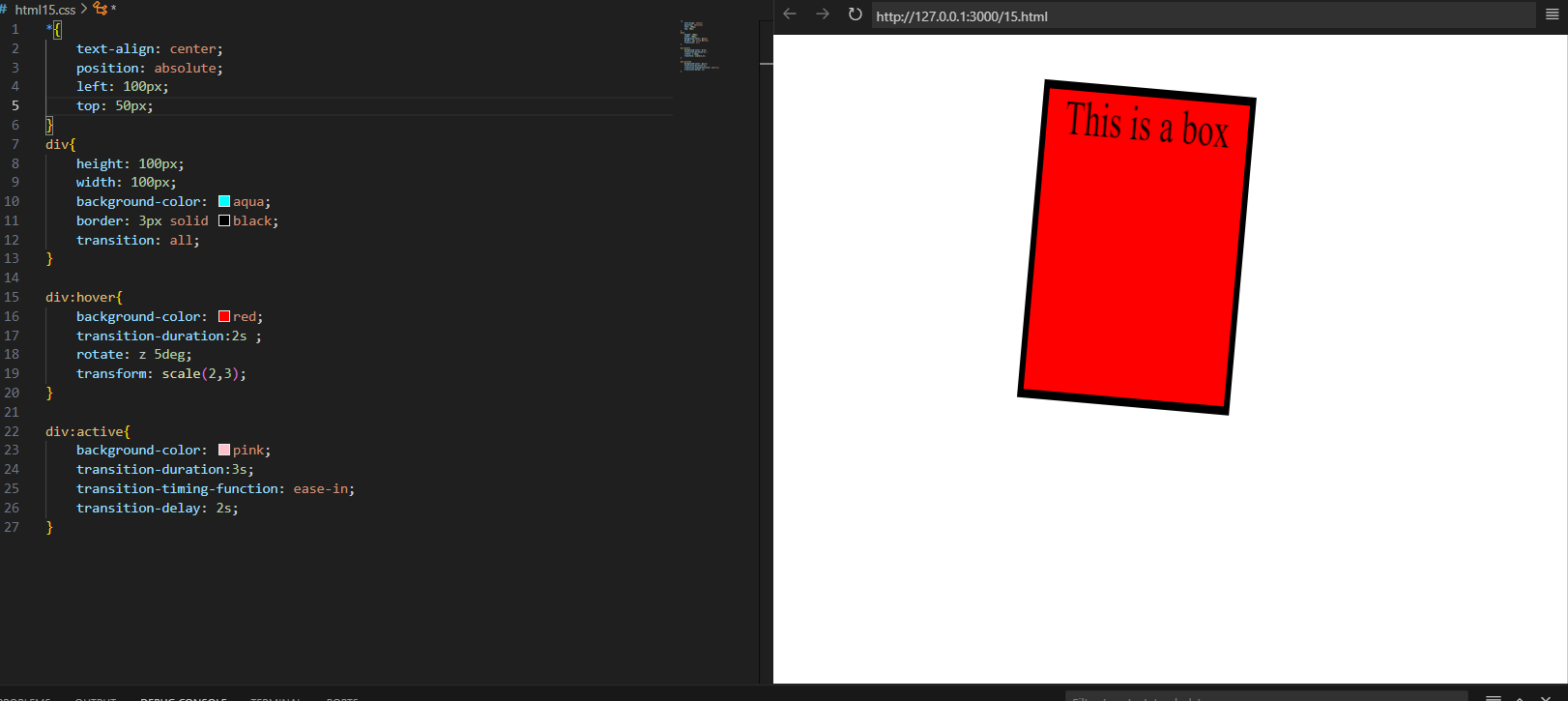
Transform:scale (x,y)

Or

Transform:scale(x)

Or

Transform:scale(y)



* **Translate:-**

🡪moving element from one place to another place like a car.

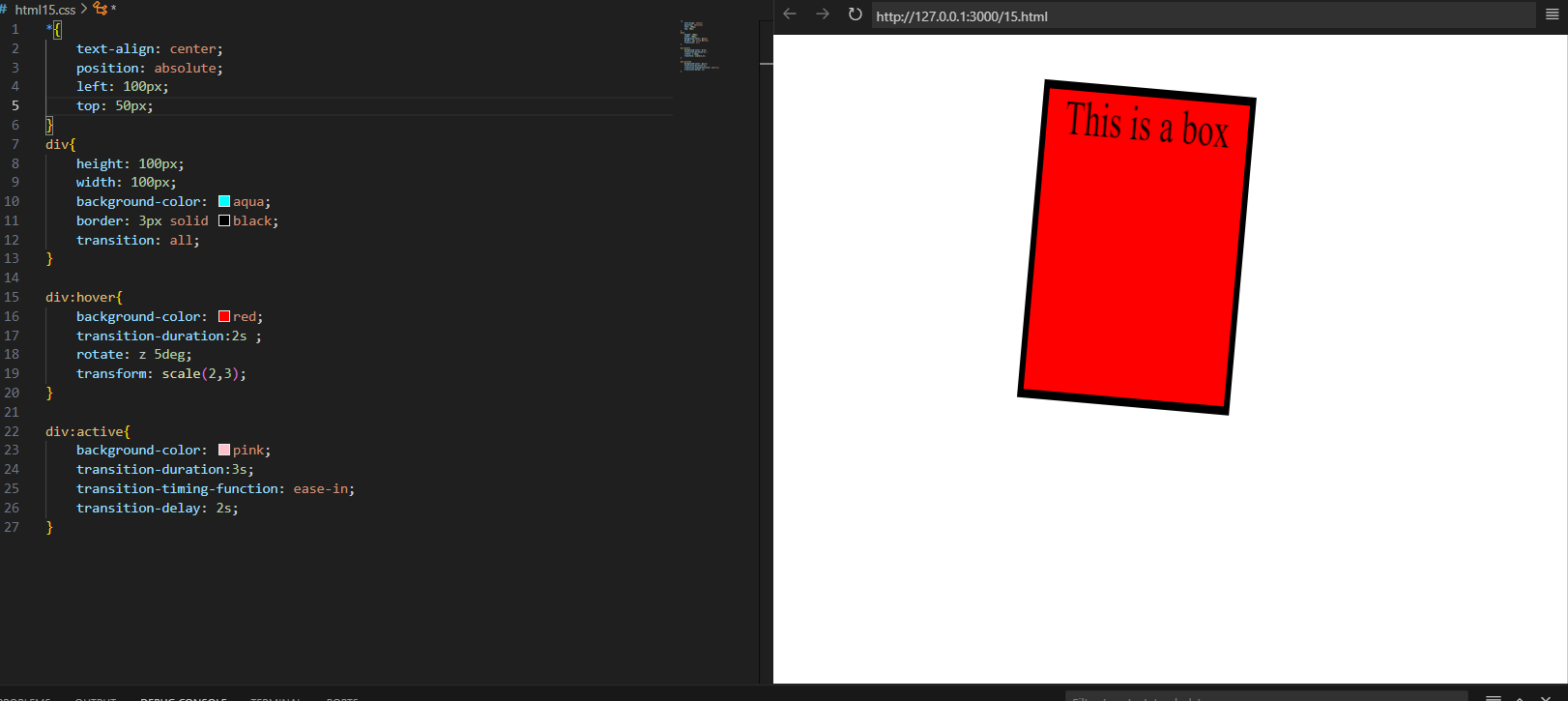
🡪syntax:-

**transform:translatex(npx);**

**transform:translatey(npx);**

**transform:translate(x px,y px);**

* Y value= negative (down)
* X= value= negative(left side)



* **Skew:-**

🡪It provides crazy or funny animations to elements

**🡪syntax:-**

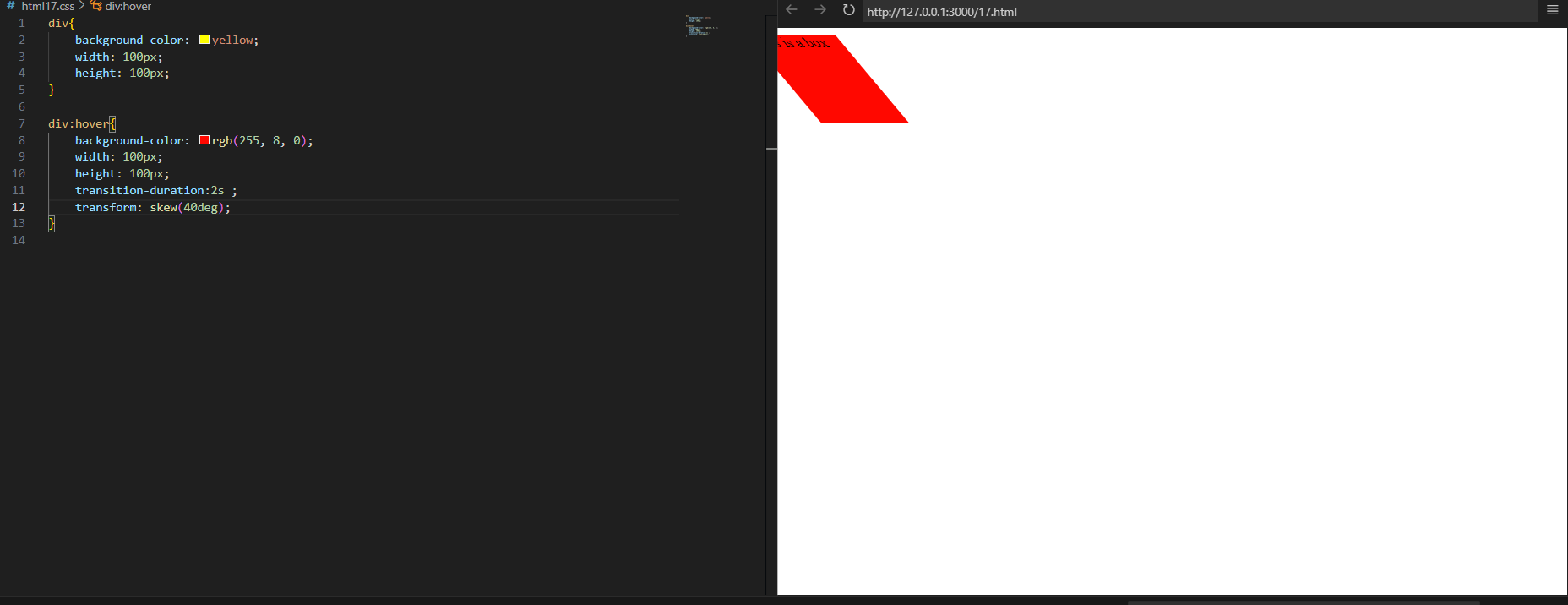
**transform: skew(x deg,y deg)**

**or**

**transform: skewx(n deg)**

**or**

**transform: skewy(n deg)**

****

**ANIMATION IN CSS:-**

🡪Used to add animation in CSS

Syntax:-

Step1:-

**@keyframe animation\_name{**

**from {property of css}**

**to{ property of css }**

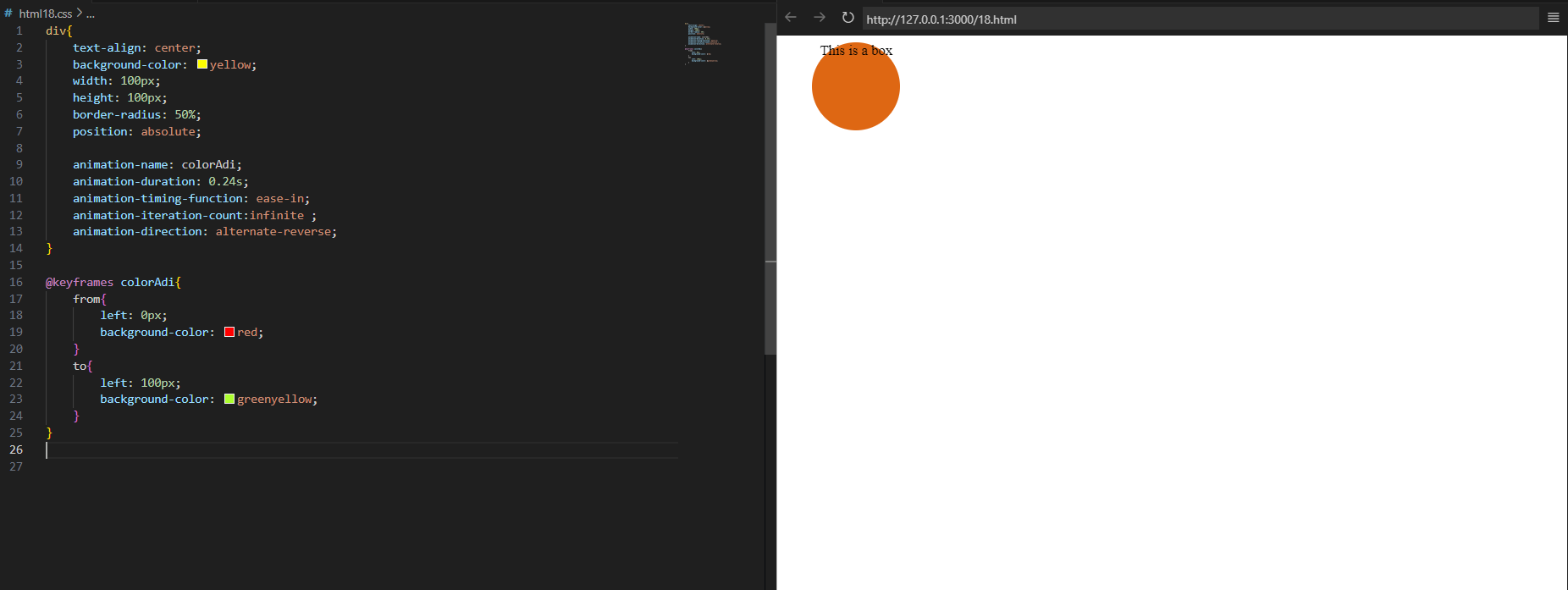
**}**

**Step2:-**

🡪declare animation property in element tag where animation is being applied

* Animation property:-

1. Animation-name
2. Animation-duration
3. Animation-timing-function
4. Animation-delay
5. Animation-iteration-count:-how many times my names will done.
6. Animation-direction:normal/reverse/alternate/alternate re

****

**🡪short hand to write this:-**

**Animation: name duration timing-function delay iteration-count direction**

**Percentage in animation:-**

🡪we can also declare keyframe as:-

Syntax:-

**@keyframe animation\_name{**

**0% {property of css}**

**50%{ property of css }**

**100%{ property of css }**

**}**