



@gregwhitworth



"Sorcerer's Apprentice - Fantasia" 1940
© Disney•Pixar. All rights reserved.



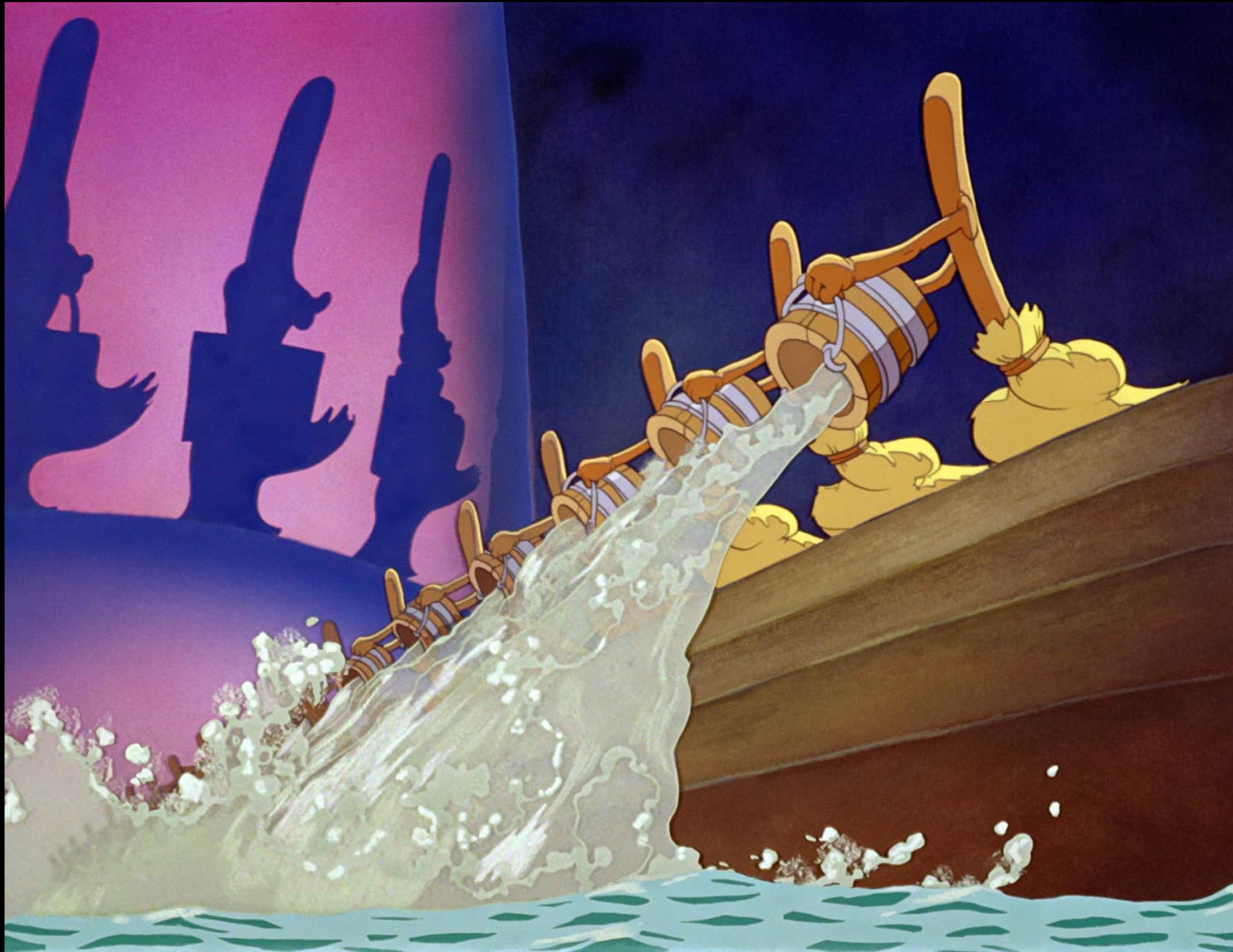
"Sorcerer's Apprentice - Fantasia" 1940
© Disney•Pixar. All rights reserved.



"Sorcerer's Apprentice - Fantasia" 1940
© Disney•Pixar. All rights reserved.



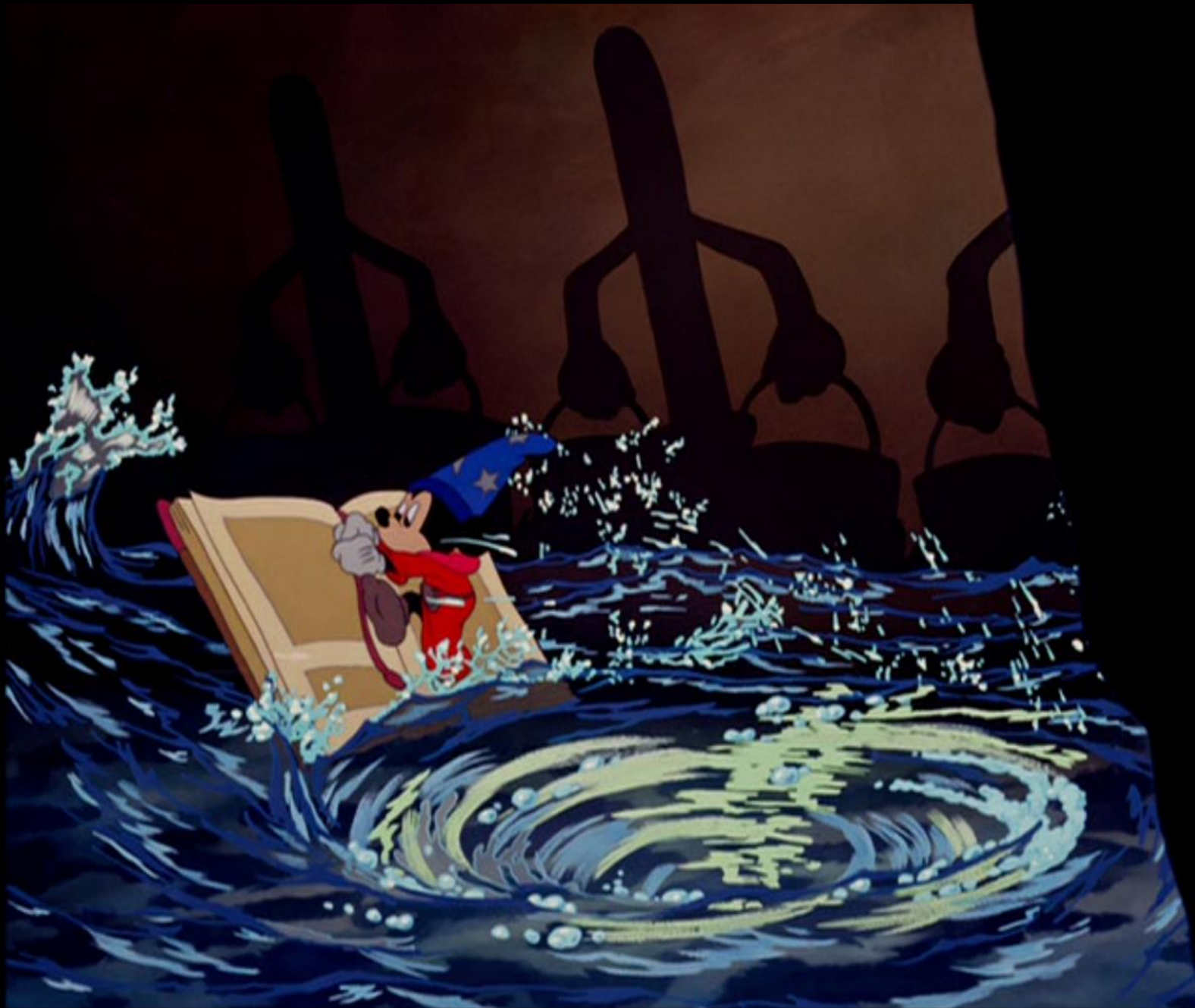
"Sorcerer's Apprentice - Fantasia" 1940
© Disney•Pixar. All rights reserved.



"Sorcerer's Apprentice - Fantasia" 1940
© Disney•Pixar. All rights reserved.



"Sorcerer's Apprentice - Fantasia" 1940
© Disney•Pixar. All rights reserved.



"Sorcerer's Apprentice - Fantasia" 1940
© Disney•Pixar. All rights reserved.



"Sorcerer's Apprentice - Fantasia" 1940
© Disney•Pixar. All rights reserved.

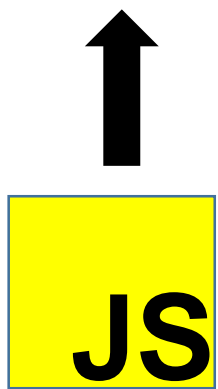
}

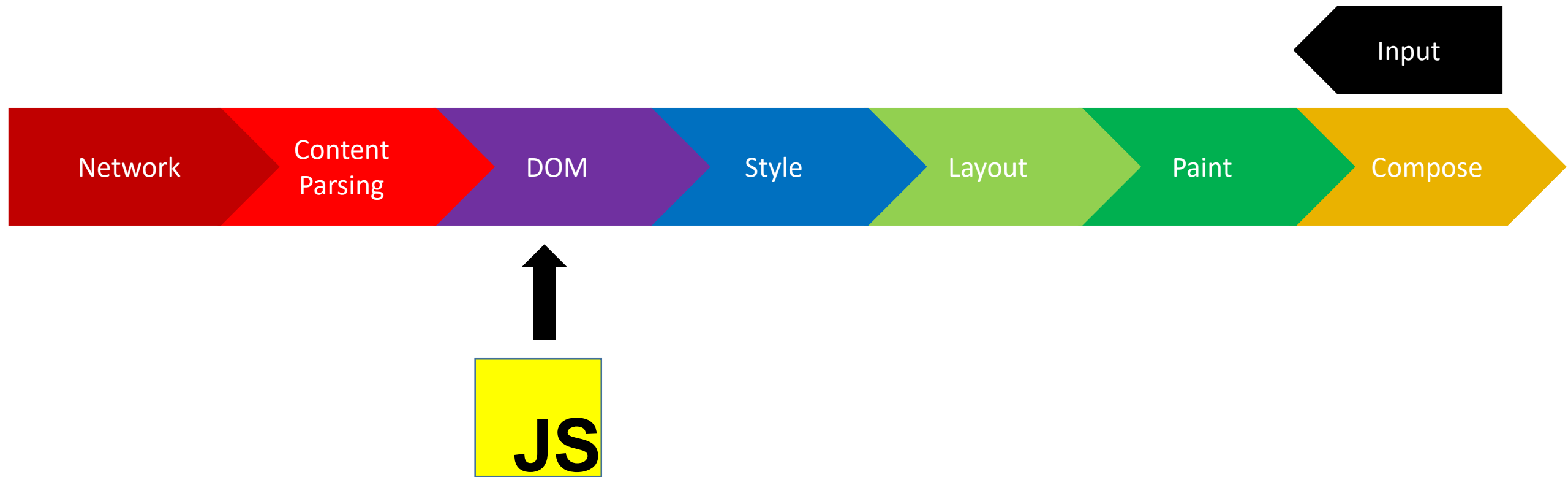


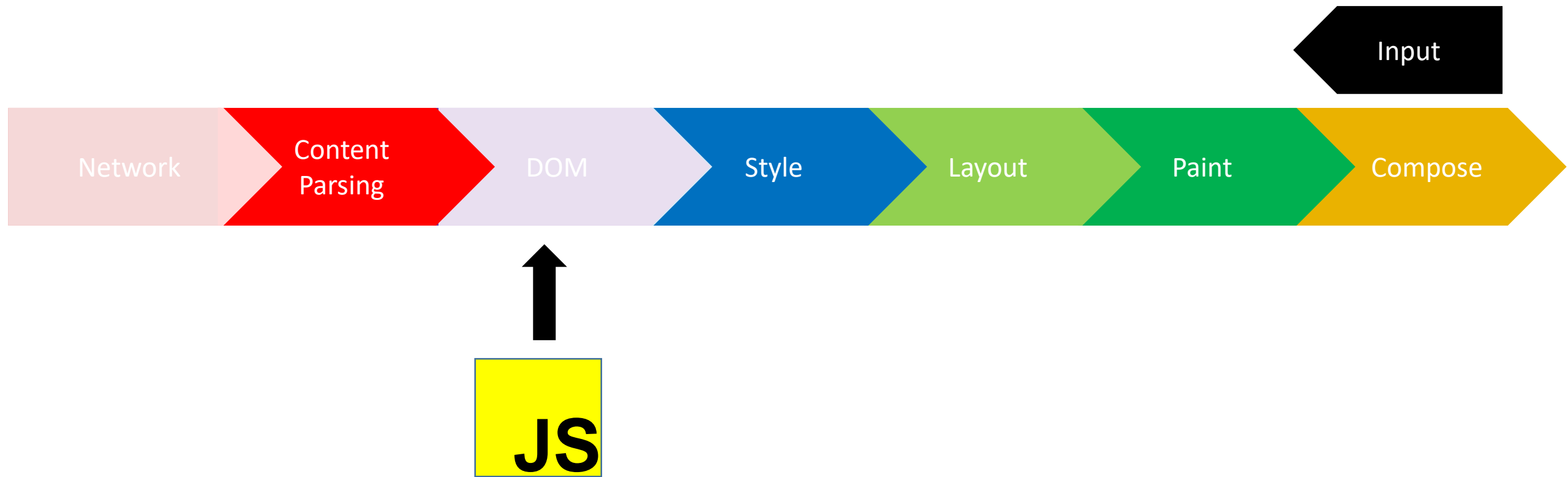
{the pipeline}











{parsing}

Network

Content
Parsing

DOM

Style

Layout

Paint

Compose

YOU
ARE
HERE

I found a selector!

```
.fancy-button {  
    background: green;  
}
```

I found an opening curly brace!

```
.fancy-button {  
    background: green;  
}
```

I found a property

```
.fancy-button {  
  background: green;  
}
```

I found a colon!

```
.fancy-button {  
    background: green;  
}
```

I found a value!

```
.fancy-button {  
    background: green;  
}
```

I found a semi-colon!

```
.fancy-button {  
    background: green;  
}
```

I found a closing curly brace!

```
.fancy-button {  
    background: green;  
}
```



```
.fancy-button {  
  background: green;  
  border: 3px solid red;  
  font-size: 1em;  
}
```

Selector

.fancy-button

```
.fancy-button {  
  background: green;  
  border: 3px solid red;  
  font-size: 1em;  
}
```

Selector

.fancy-button

```
.fancy-button {  
  background: green;  
  border: 3px solid red;  
  font-size: 1em;  
}
```

Selector	Property	Value
.fancy-button	background-color	rgb(0,255,0)

```
.fancy-button {  
  background: green;  
  border: 3px solid red;  
  font-size: 1em;  
}
```

Selector	Property	Value
.fancy-button	background-color	rgb(0,255,0)
.fancy-button	border-size	3px
.fancy-button	border-color	rgb(255,0,0)
.fancy-button	border-style	solid

```
.fancy-button {  
  background: green;  
  border: 3px solid red;  
  font-size: 1em;  
}
```

Selector	Property	Value
.fancy-button	background-color	rgb(0,255,0)
.fancy-button	border-size	3px
.fancy-button	border-color	rgb(255,0,0)
.fancy-button	border-style	solid
.fancy-button	font-size	1em

{style}

Network

Content
Parsing

DOM

Style

Layout

Paint

Compose

YOU
ARE
HERE

{style:terminology}

Value Types

Declared: What the author wrote

width: 506.456789132456489787px

Value Types

Specified: Declared + defaults

width: auto

Value Types

Computed: Resolution of specified

font-size: 16px

Value Types

Used: Result of layout calculations

width: 350px

Value Types

Actual: UA specific adjustments

width: 506.456789132456489787px

{style:computation}

The User Agent should have
everything resolved except
what **requires** layout

Before

width: 1in;

width: 50%;

line-height: calc(10px + 2em);

border-color: currentColor;

height: 50vh;

After

width: 96px;

width: 50%;

line-height: 42px;

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

height: 540px;


```
{style:cascade}
```

Specificity

A score awarded to a selector based on the count of tags, classes, id's, !important, & attribute selectors used

Specificity

Inline styles win out over all selectors unless an !important was used

Specificity

- !Important
- Style Attribute
- ID
- Classes, Psuedo Classes, attr
- Elements & Psuedo Elements

li

0 0 0 0 1

li

0 0 0 0 1

li.foo

0 0 0 1 1

li	0	0	0	0	1
li.foo	0	0	0	1	1
#comment li.foo.bar	0	0	1	2	1

li	0	0	0	0	1
li.foo	0	0	0	1	1
#comment li.foo.bar	0	0	1	2	1
<li style="color: red">	0	1	0	0	0

li	0	0	0	0	1
li.foo	0	0	0	1	1
#comment li.foo.bar	0	0	1	2	1
<li style="color: red">	0	1	0	0	0
color: red !important	1	0	0	0	0

“Keep your CSS selectors short”

by Harry Roberts

bit.ly/css-short-selectors

Style Origins

An origin is what determines where the stylesheet is placed within the cascade

3 currently specified origins

User Agent: Renders CSS

Author : Web Developer

User : Person using site

Ordering in the cascade

1. User
2. Author
3. User Agent

```

.fancy-button {
  background: green;
  border: 3px solid red;
  font-size: 1em;
}

```

```

div .fancy-button {
  background: yellow;
}

```

Origin	Selector	Property	Value	Specificity
Author	.fancy-button	background-color	rgb(0,255,0)	0,0,0,1,0
Author	.fancy-button	border-style	Solid	0,0,0,1,0
Author	.fancy-button	border-size	3px	0,0,0,1,0
Author	.fancy-button	border-color	rgb(255,0,0)	0,0,0,1,0
Author	.fancy-button	font-size	16px	0,0,0,1,0
Author	div .fancy-button	background-color	rgb(255,255,0)	0,0,0,1,1
User	*	font-size	25px	1,0,0,0,0

```

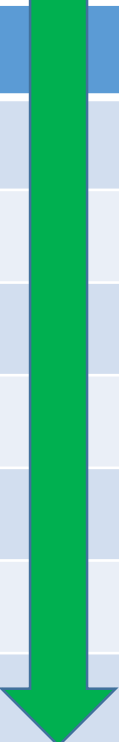
.fancy-button {
  background: green;
  border: 3px solid red;
  font-size: 1em;
}

```

```

div .fancy-button {
  background: yellow;
}

```



Origin	Selector	Property	Value	Specificity
User	*	font-size	25px	1,0,0,0,0
Author	.fancy-button	background-color	rgb(0,255,0)	0,0,0,1,0
Author	.fancy-button	border-style	Solid	0,0,0,1,0
Author	.fancy-button	border-size	3px	0,0,0,1,0
Author	.fancy-button	border-color	rgb(255,0,0)	0,0,0,1,0
Author	.fancy-button	font-size	16px	0,0,0,1,0
Author	div .fancy-button	background-color	rgb(255,255,0)	0,0,0,1,1

```

.fancy-button {
  background: green;
  border: 3px solid red;
  font-size: 1em;
}

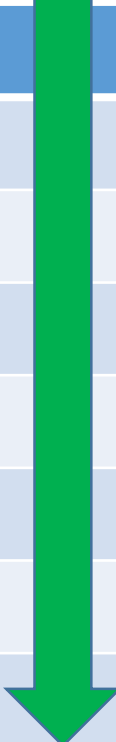
```

```

div .fancy-button {
  background: yellow;
}

```

Origin	Selector	Property	Value	Specificity
User	*	font-size	25px	1,0,0,0,0
Author	div .fancy-button	background-color	rgb(255,255,0)	0,0,0,1,1
Author	.fancy-button	background-color	rgb(0,255,0)	0,0,0,1,0
Author	.fancy-button	border-style	Solid	0,0,0,1,0
Author	.fancy-button	border-size	3px	0,0,0,1,0
Author	.fancy-button	border-color	rgb(255,0,0)	0,0,0,1,0
Author	.fancy-button	font-size	16px	0,0,0,1,0




```

.fancy-button {
  background: green;
  border: 3px solid red;
  font-size: 1em;
}

```

```

div .fancy-button {
  background: yellow;
}

```

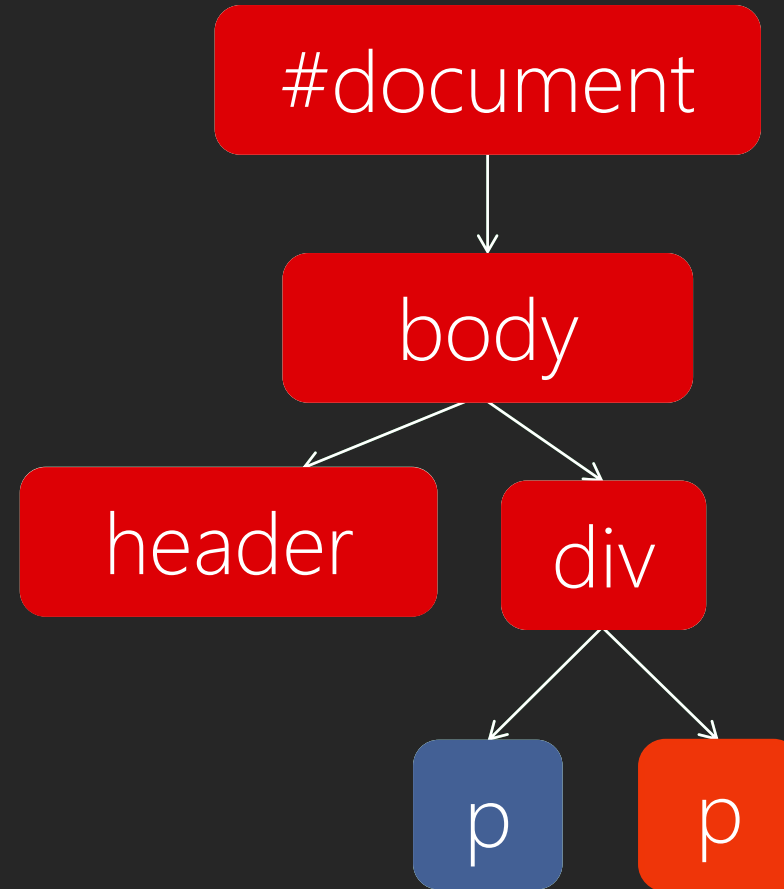
Origin	Selector	Property	Value	Specificity
User	*	font-size	25px	1,0,0,0,0
Author	div .fancy-button	background-color	rgb(255,255,0)	0,0,0,1,1
Author	.fancy-button	background-color	rgb(0,255,0)	0,0,0,1,0
Author	.fancy-button	border-style	Solid	0,0,0,1,0
Author	.fancy-button	border-size	3px	0,0,0,1,0
Author	.fancy-button	border-color	rgb(255,0,0)	0,0,0,1,0
Author	.fancy-button	font-size	16px	0,0,0,1,0



.fancy-button

Property	Value
border-style	Solid
border-size	3px
border-color	rgb(255,0,0)
background-color	rgb(255,255,0)
font-size	25px

```
* {  
  background: red  
}  
  
div p {  
  background: blue  
}  
  
p:last-of-type {  
  background: orange  
}
```



{layout}

Network

Content
Parsing

DOM

Style

Layout

Paint

Compose

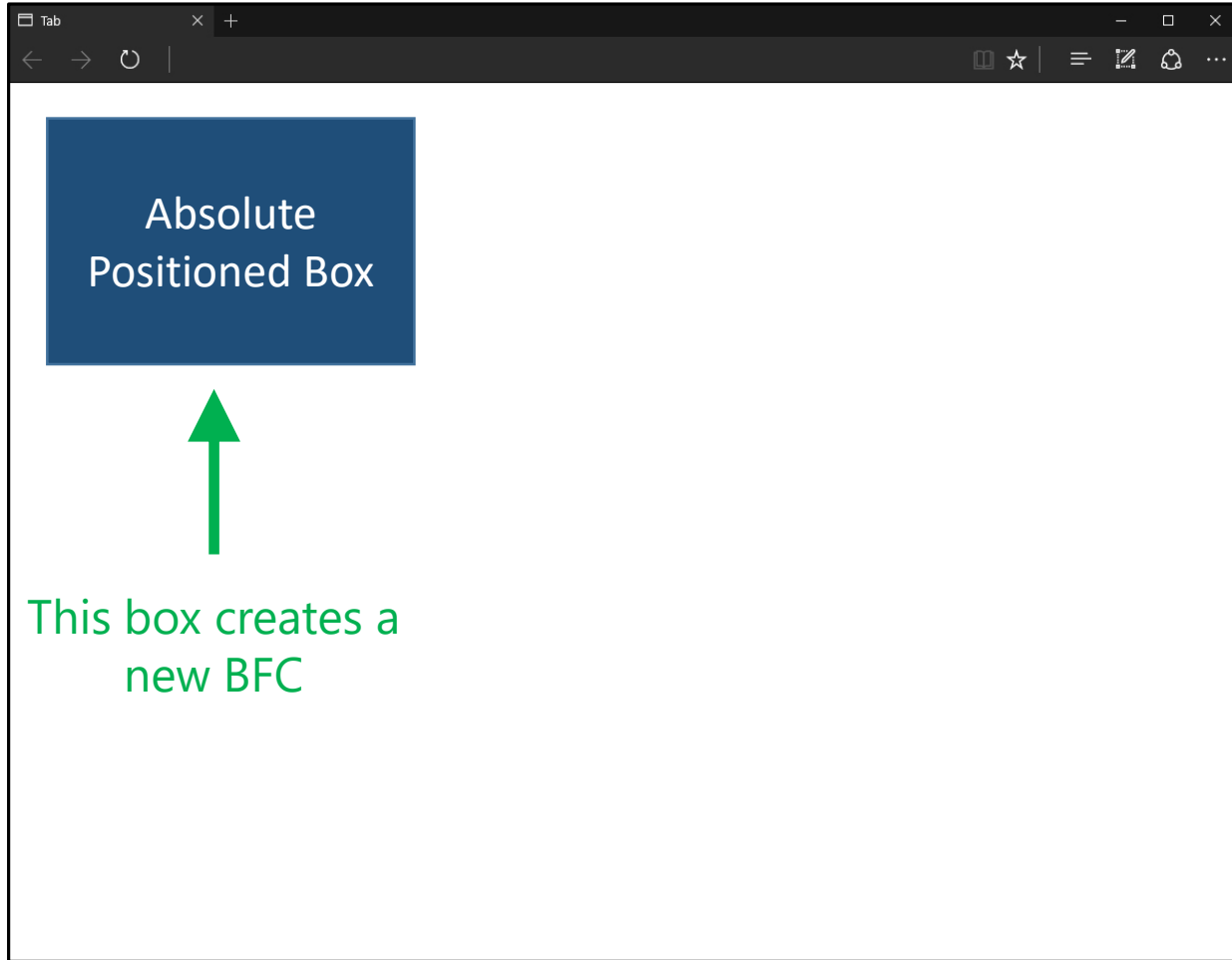
YOU
ARE
HERE

{layout:terminology}

Block formatting context (BFC)

Floats, absolutely positioned elements, block containers (such as inline-blocks, table-cells, and table-captions) that are not block boxes, and block boxes with 'overflow' other than 'visible' (except when that value has been propagated to the viewport) establish new block formatting contexts for their contents.

Block formatting context (BFC)



Containing block

The ancestor block that styles are resolved against

W3C CSS 2.2 Definition of Containing Block

<http://bit.ly/containing-block>

Containing Block

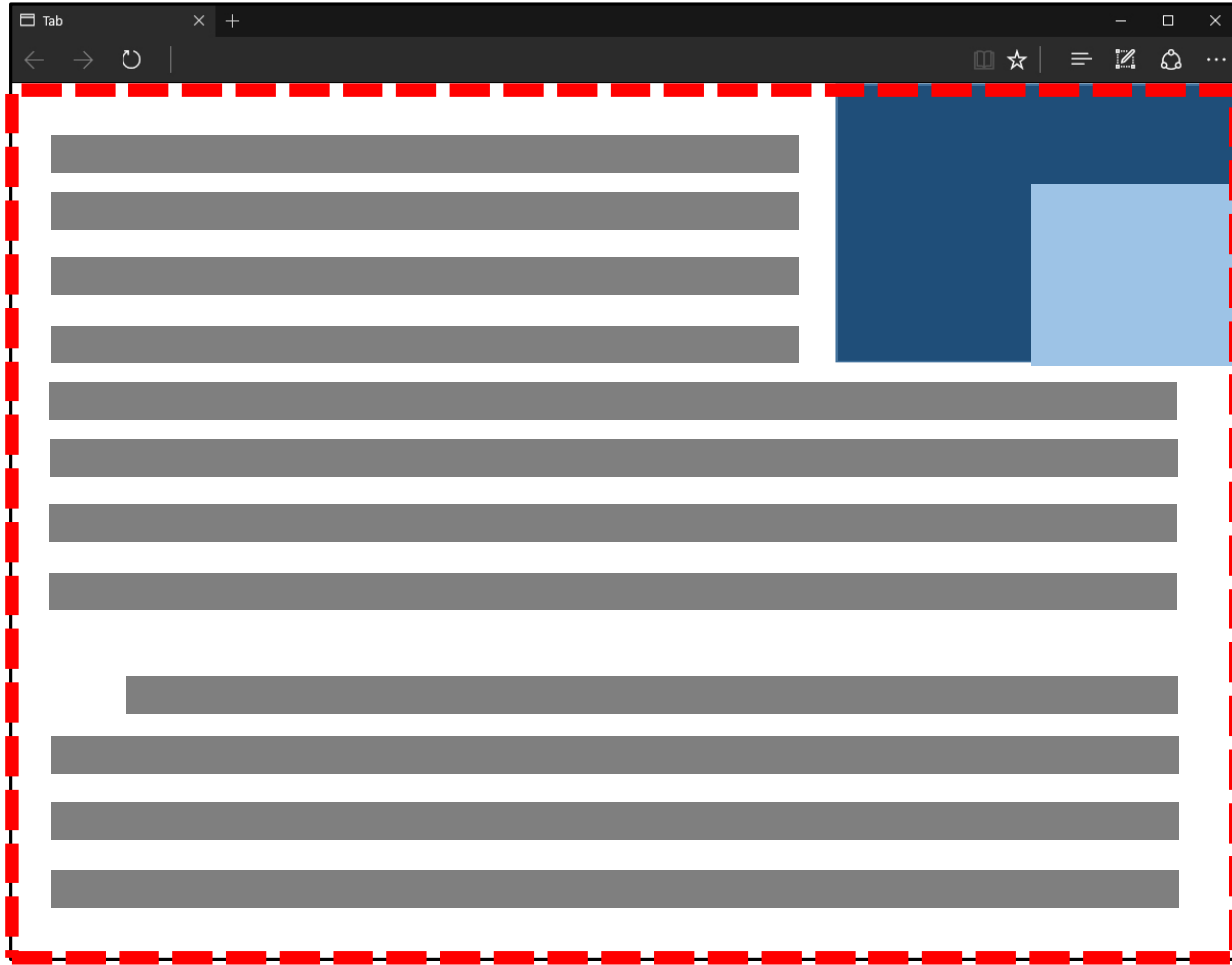


position: fixed

Initial Containing block (ICB)

A rectangle that is computed based on the top-most box or page box.

Initial Containing Block (ICB)



The red dashed box is the ICB

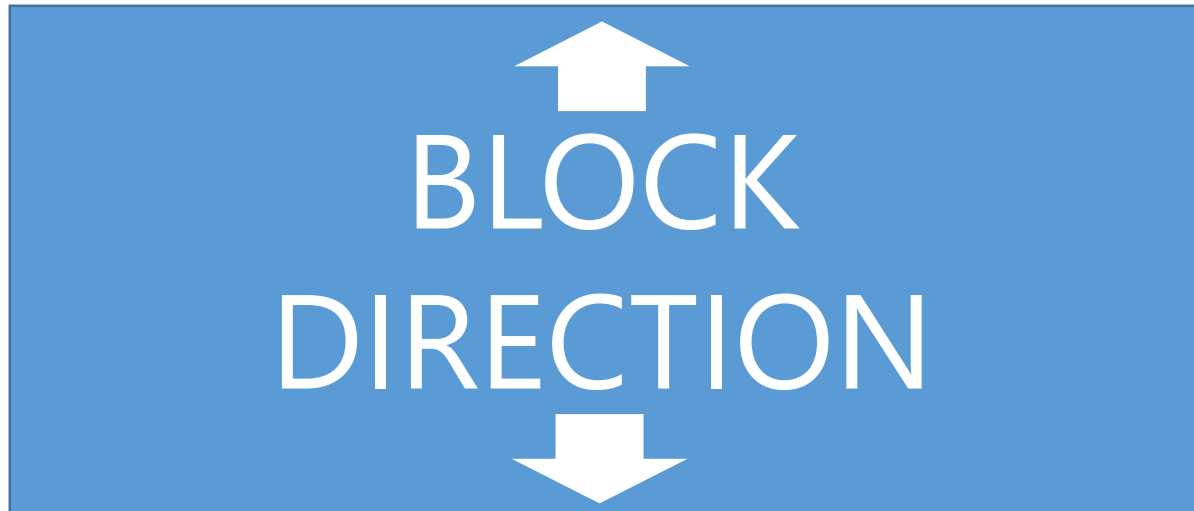
LTR/RTL Writing Mode



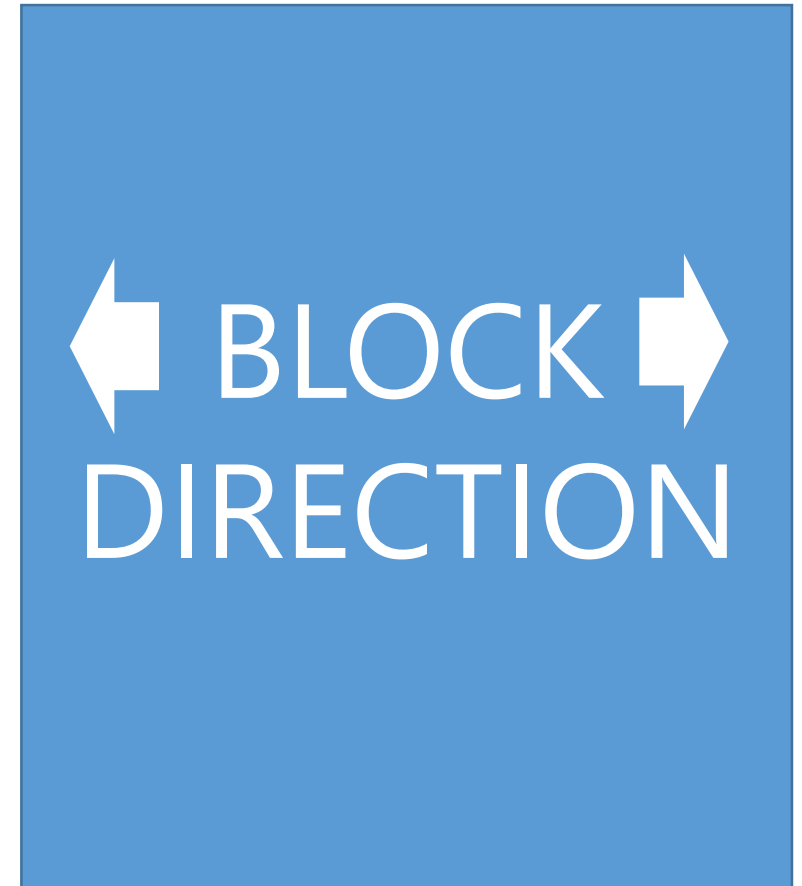
Vertical Writing Mode



LTR/RTL Writing Mode



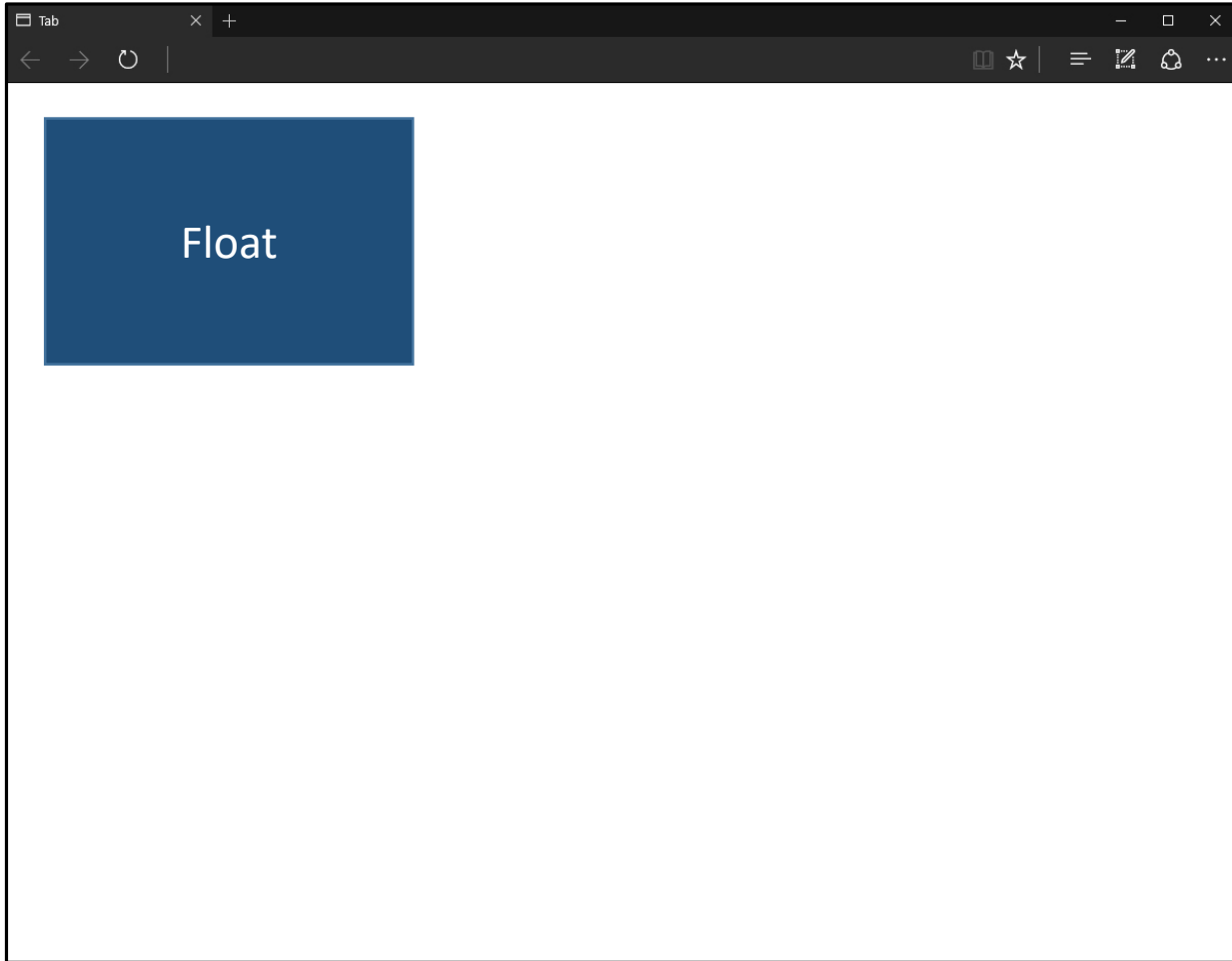
Vertical Writing Mode



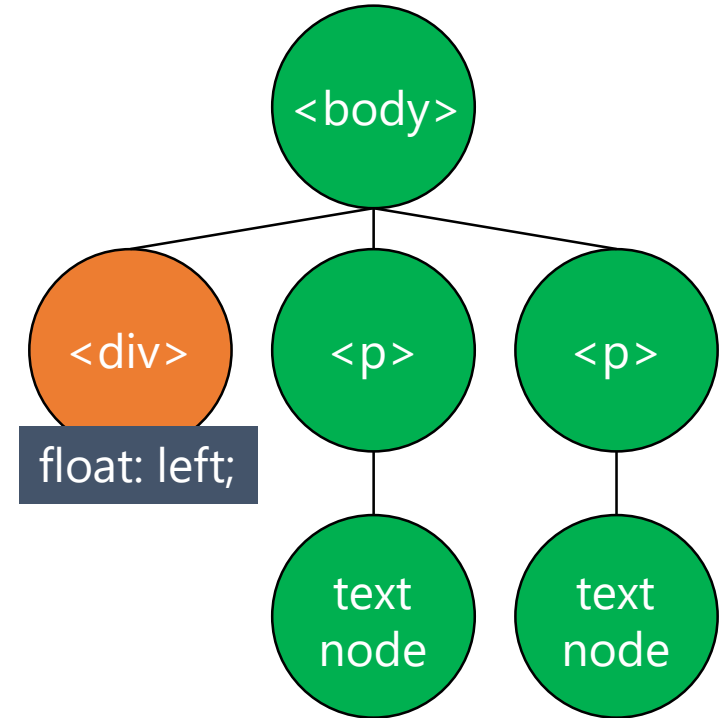
{layout:flow}

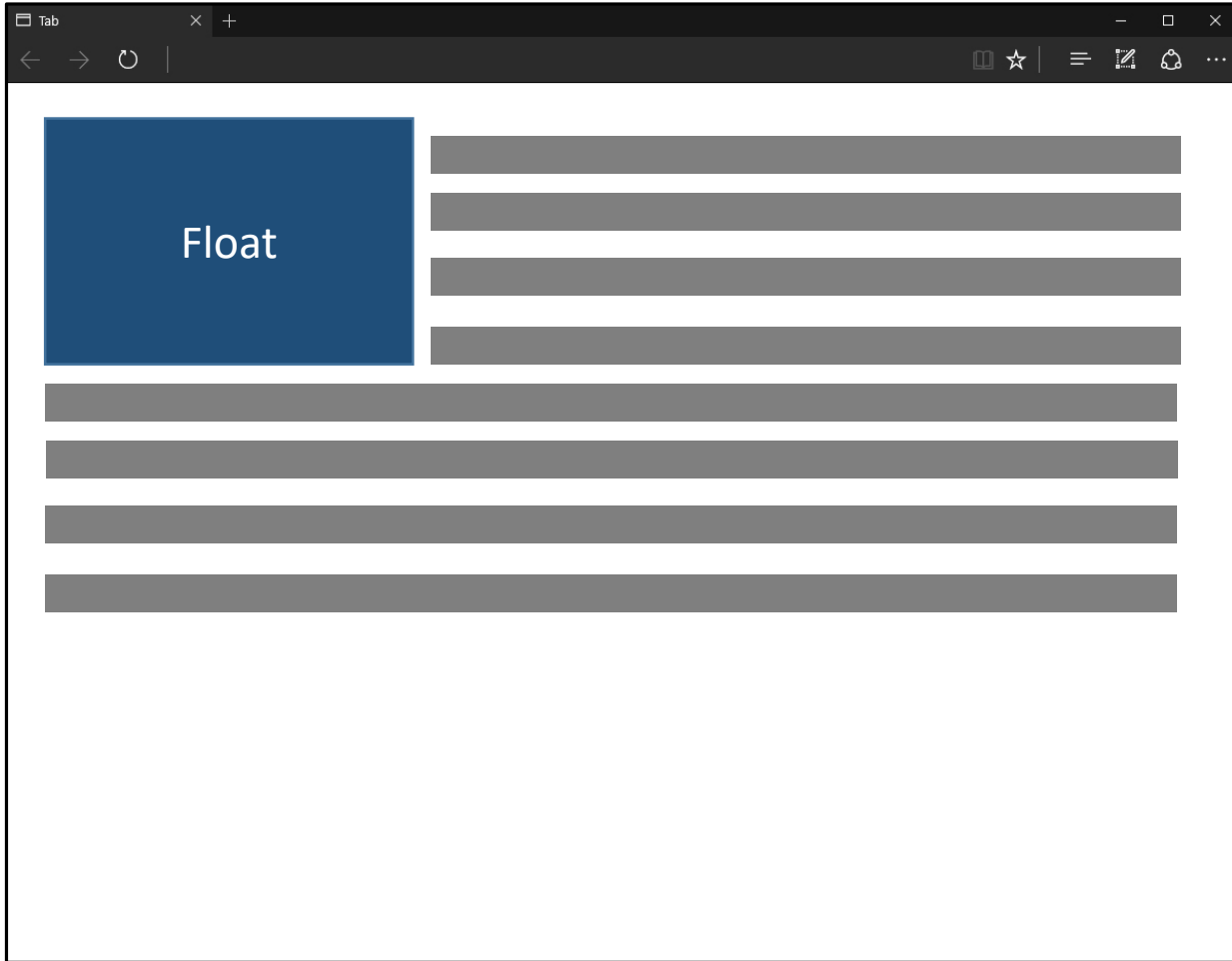
The flow

You move in document order
laying out each child in the
direction of the boxes
writing mode

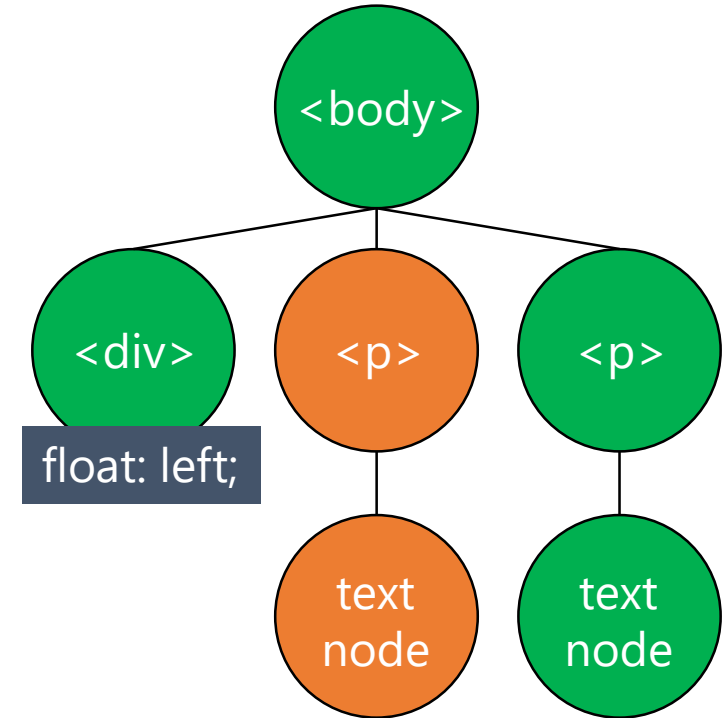


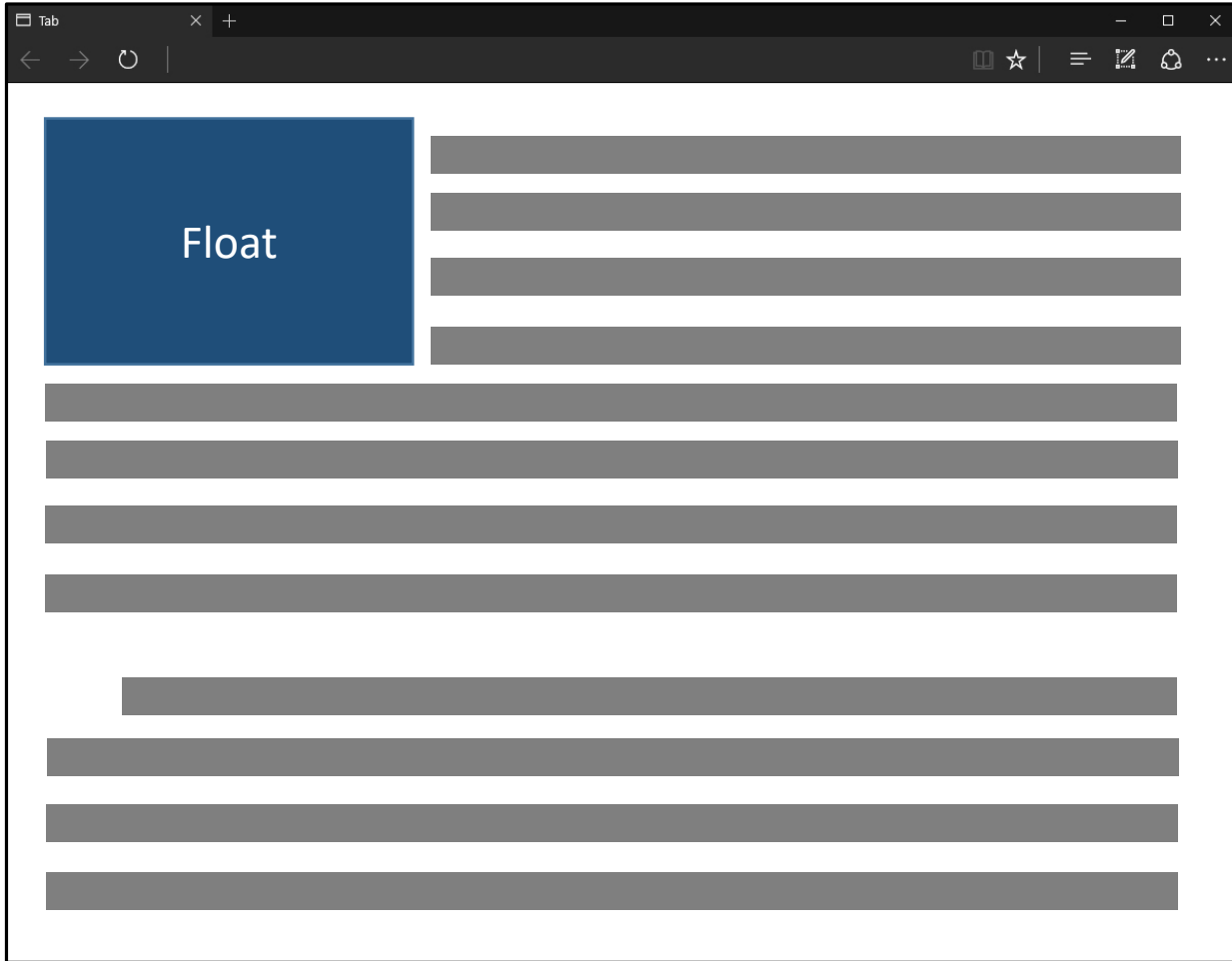
DOM Tree



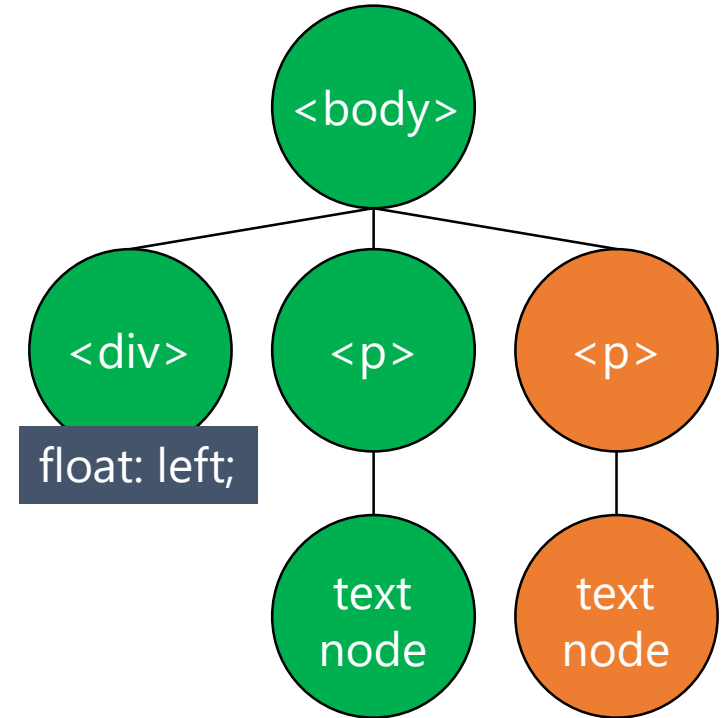


DOM Tree





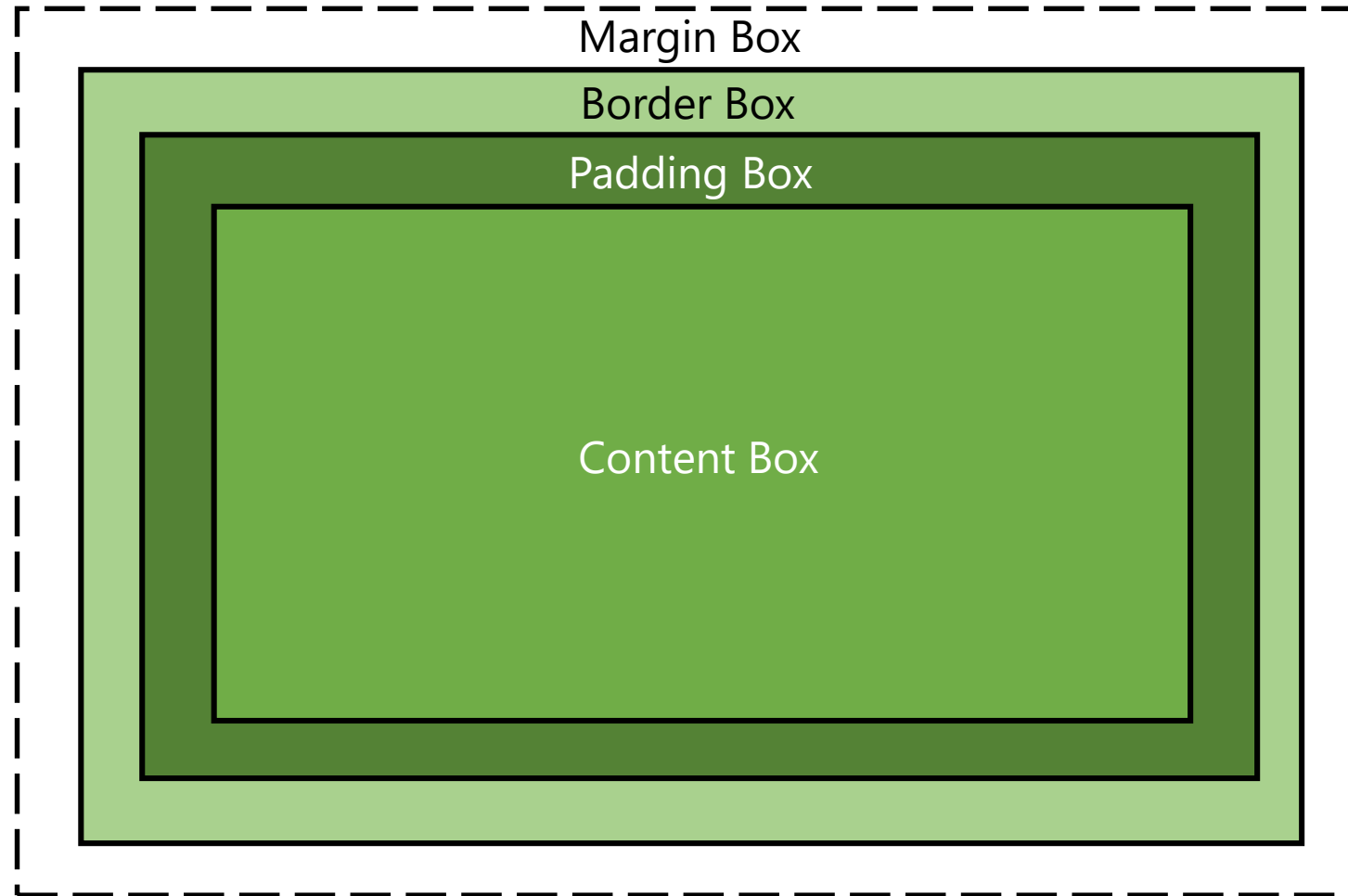
DOM Tree



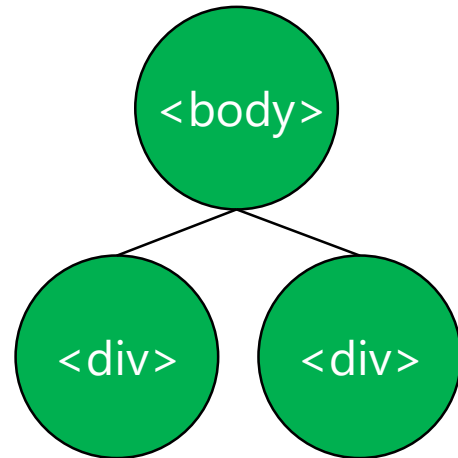
{layout:box generation}

For each element in the DOM
there will be 0 or more boxes
in the box tree

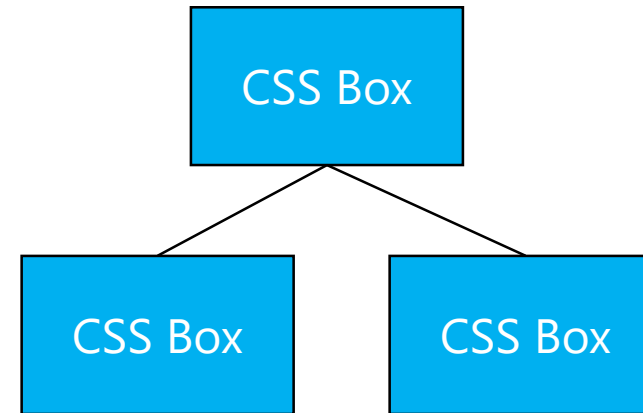
Box Model



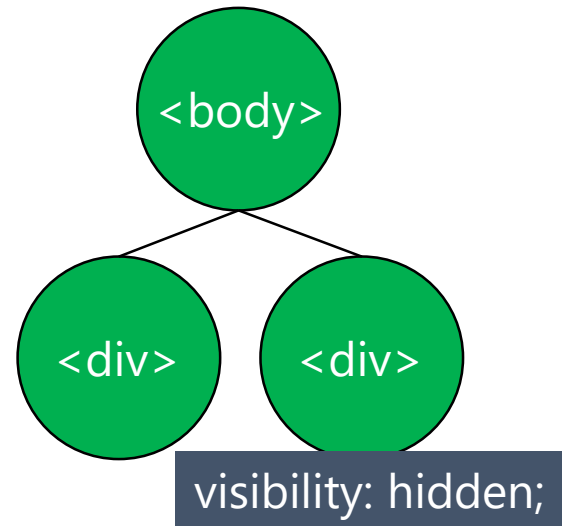
DOM Tree



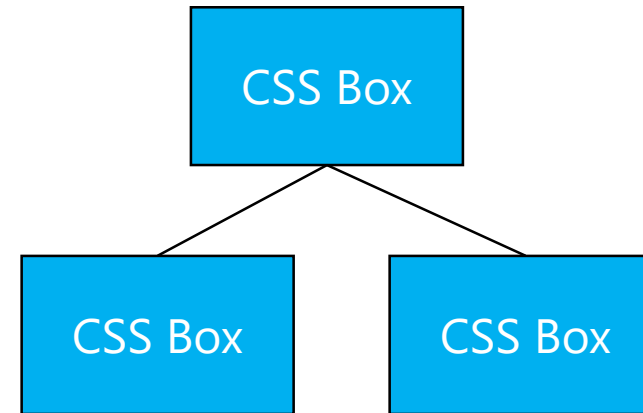
Box Tree



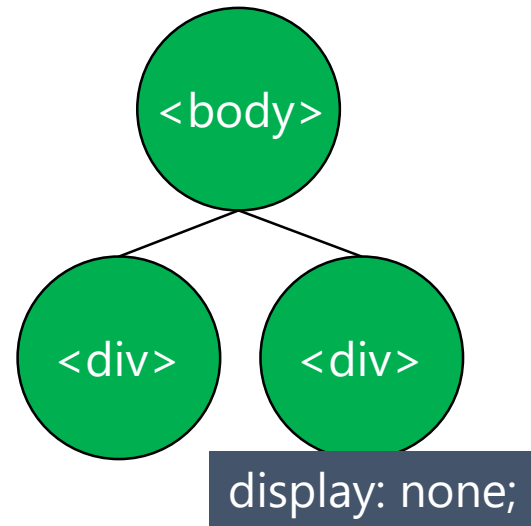
DOM Tree



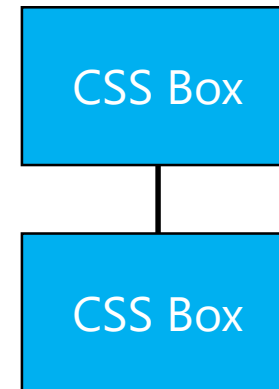
Box Tree



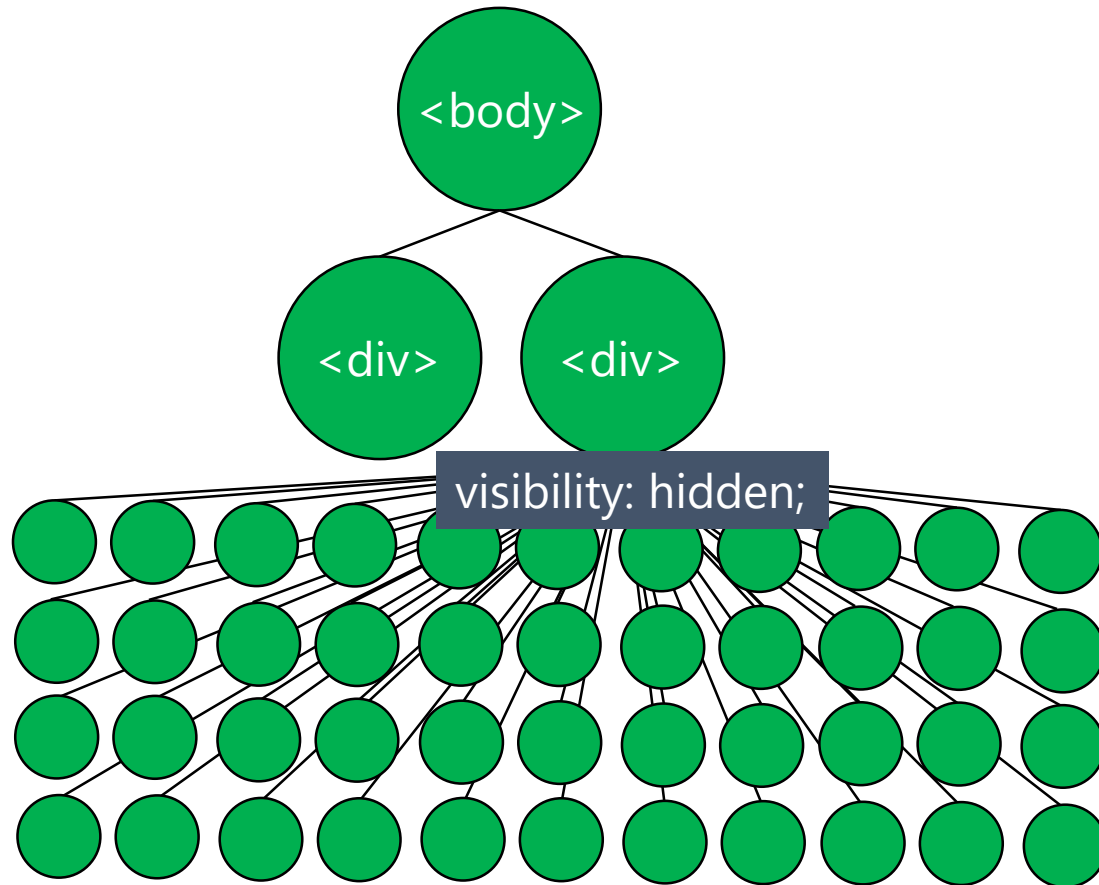
DOM Tree



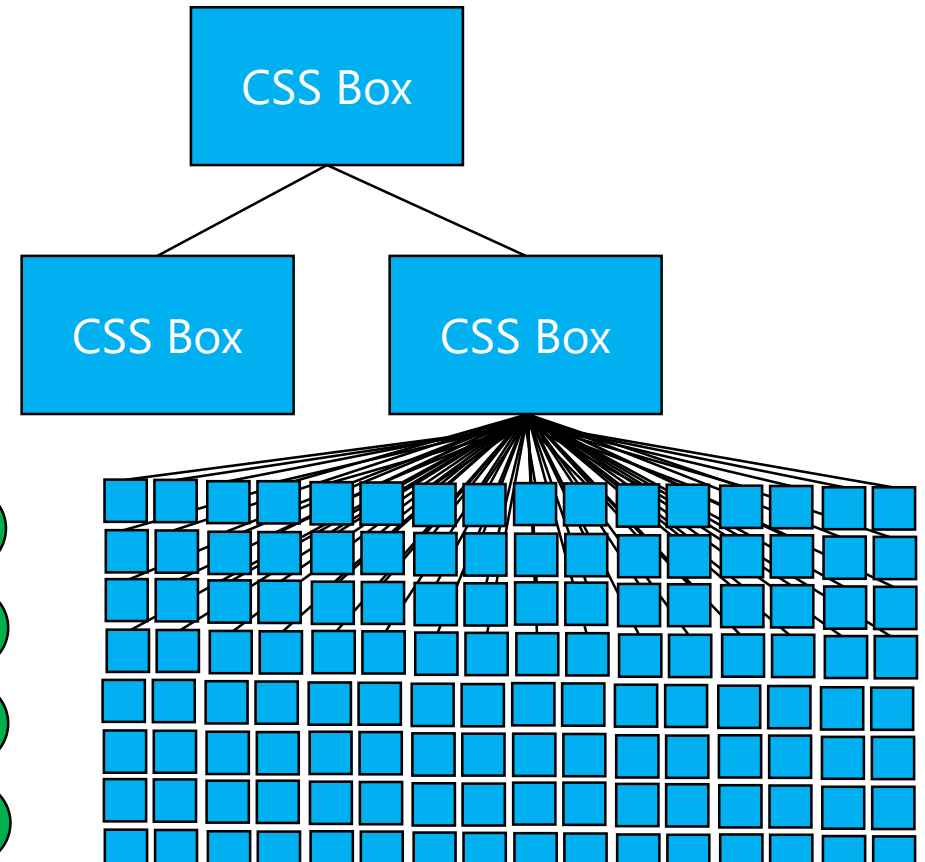
Box Tree



DOM Tree

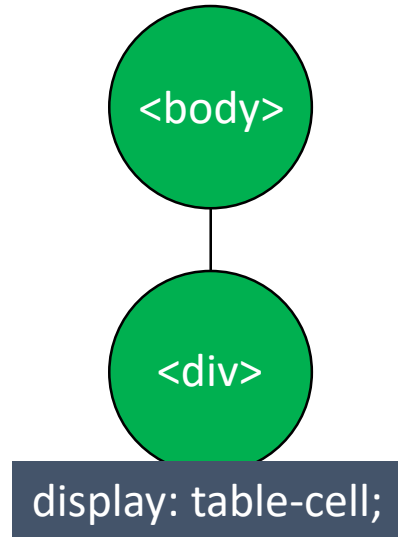


Box Tree

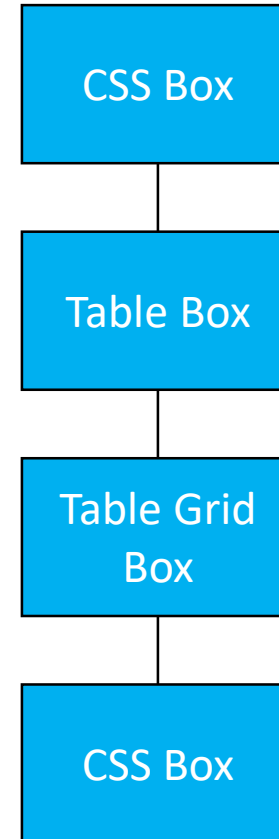


When necessary, boxes will be generated that are not in the DOM, these are known as **anonymous boxes**

DOM Tree



Box Tree



{layout:basic}

Basic Example

HTML

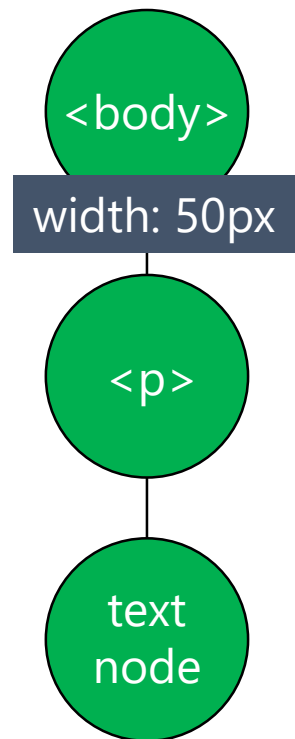
```
<body>  
  <p>  
    Hello World  
  </p>  
</body>
```

CSS

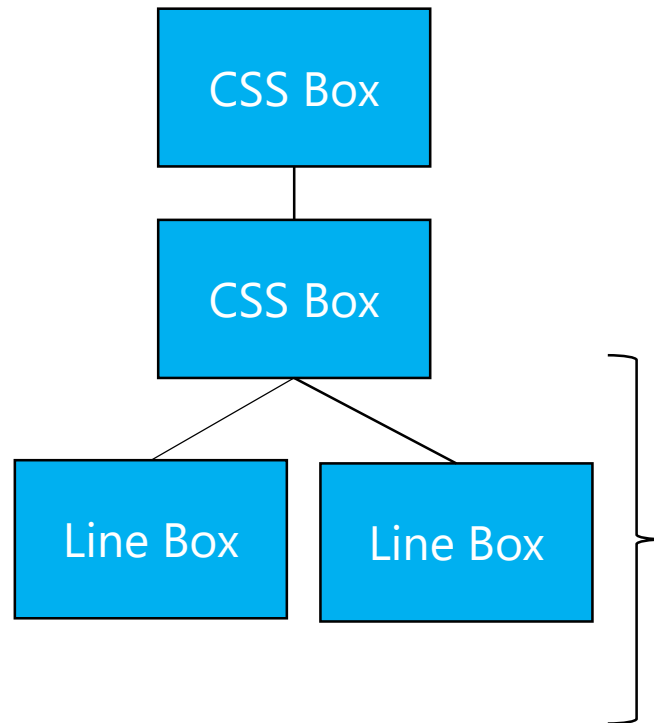
```
body {  
  width: 50px;  
}
```

Basic Example

DOM Tree



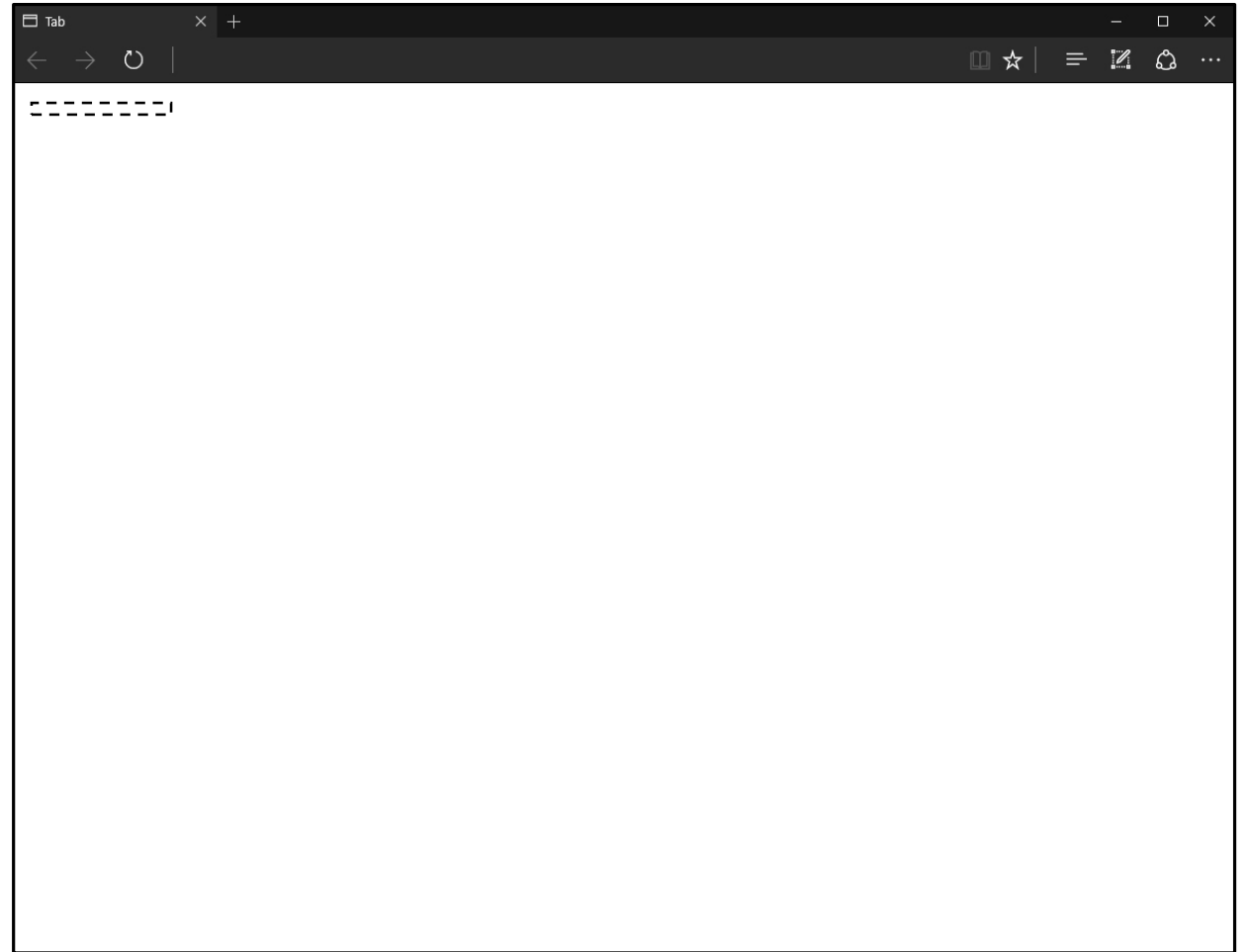
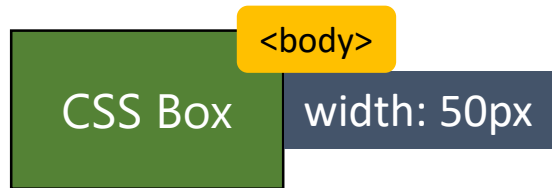
Box Tree



Why are there 2 line boxes for the 1 text node?

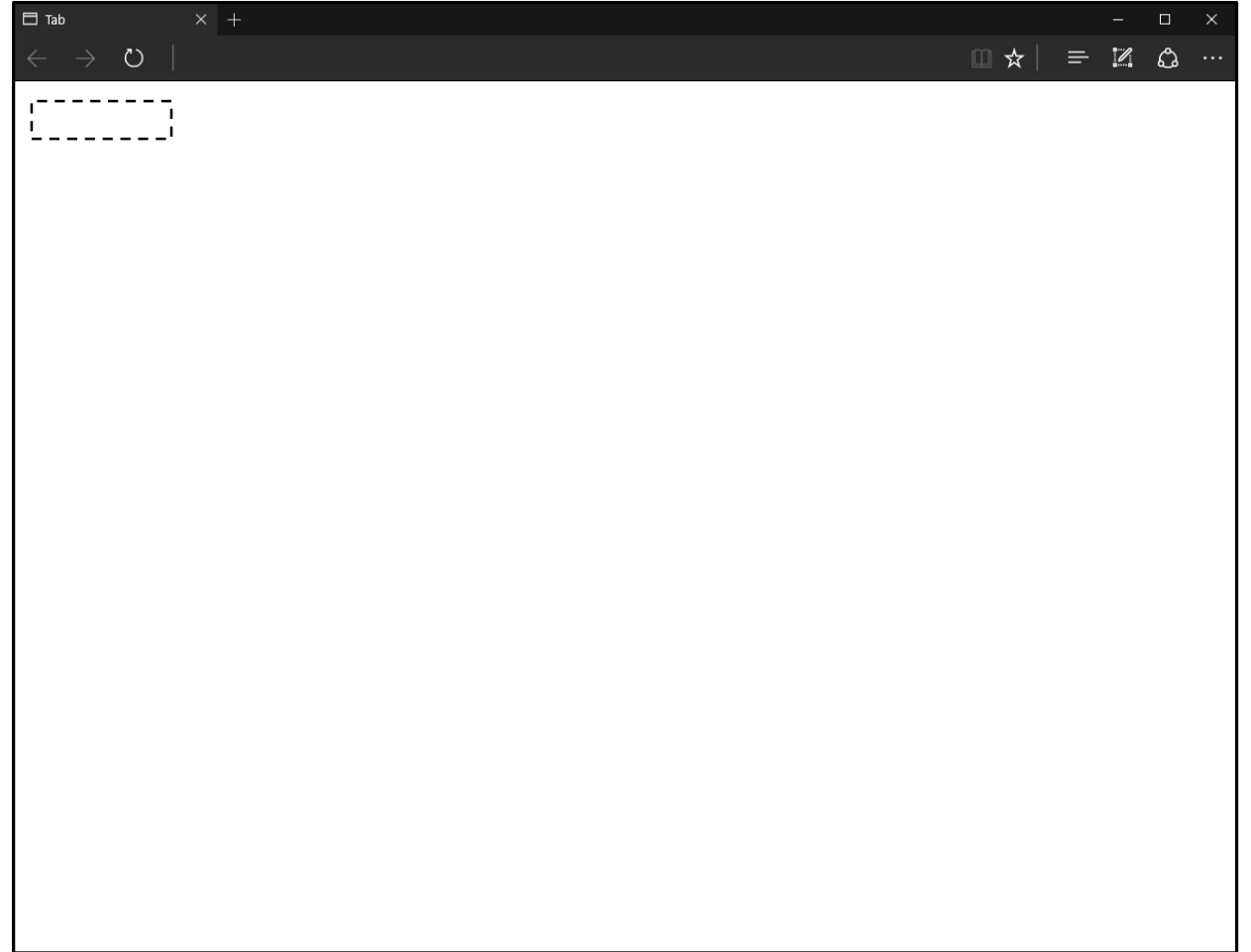
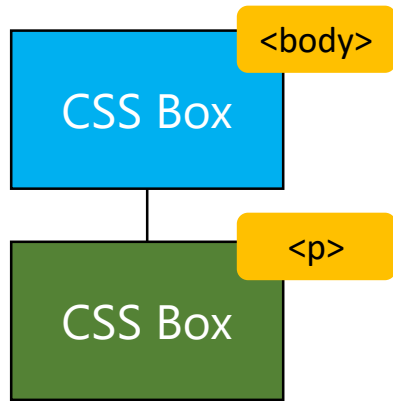
Basic Example

Box Tree



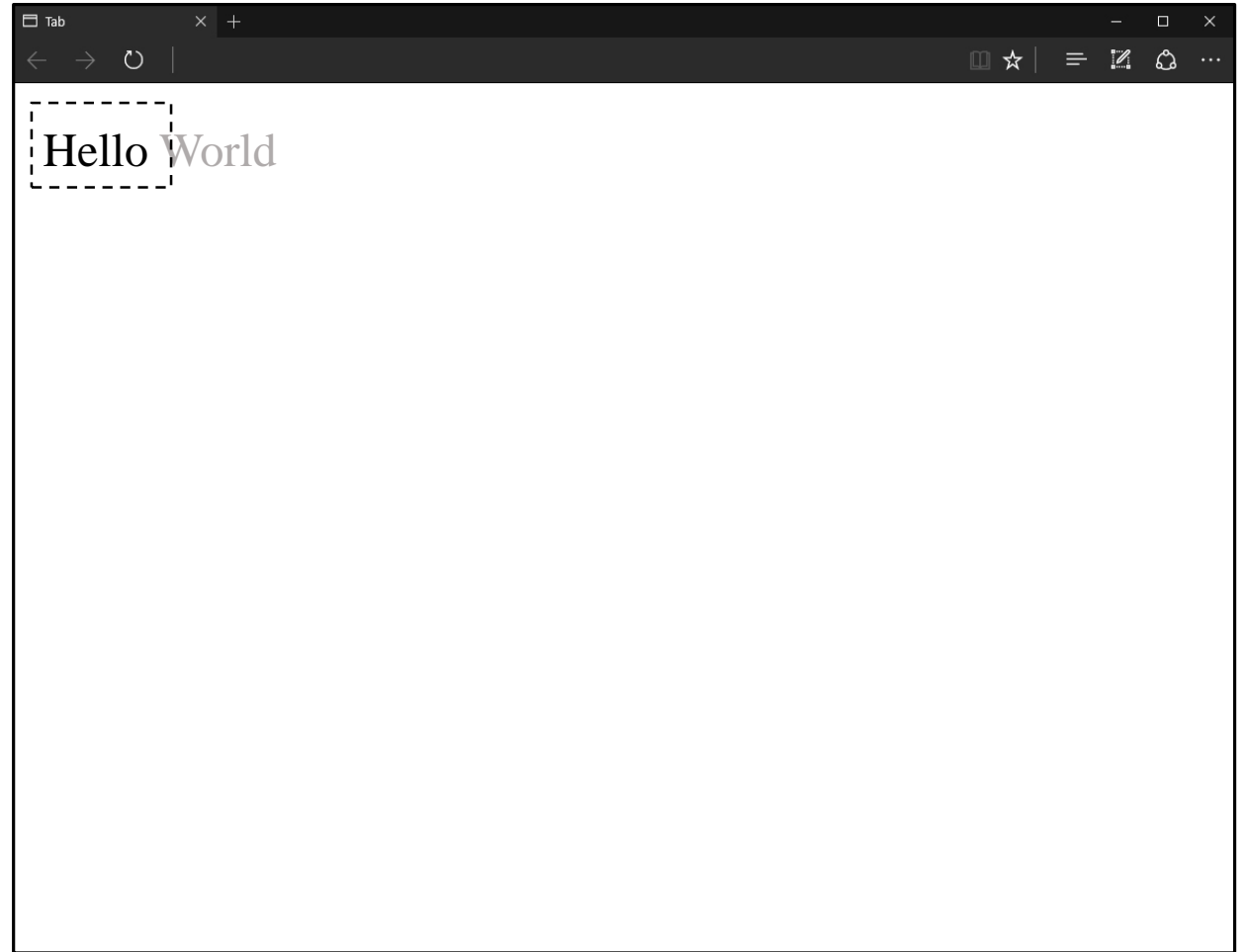
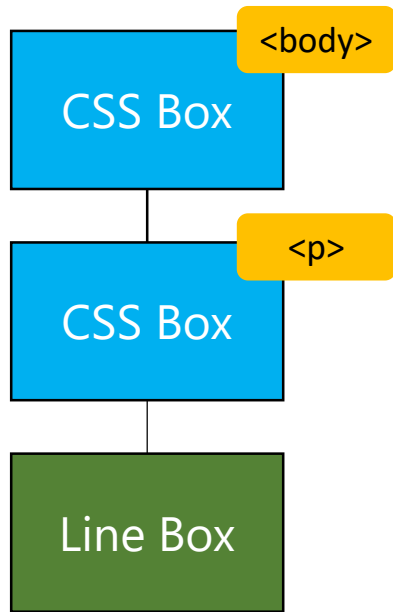
Basic Example

Box Tree



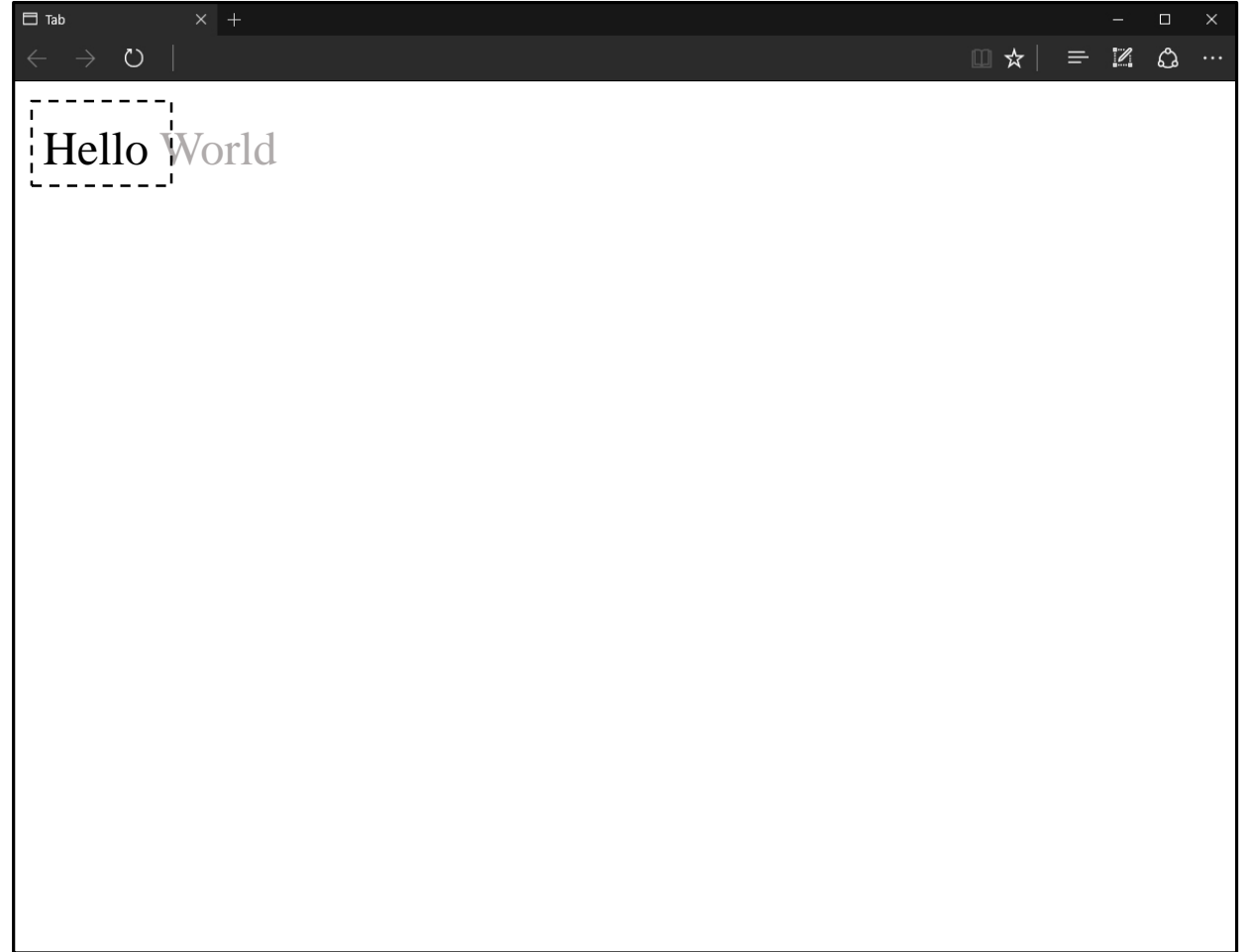
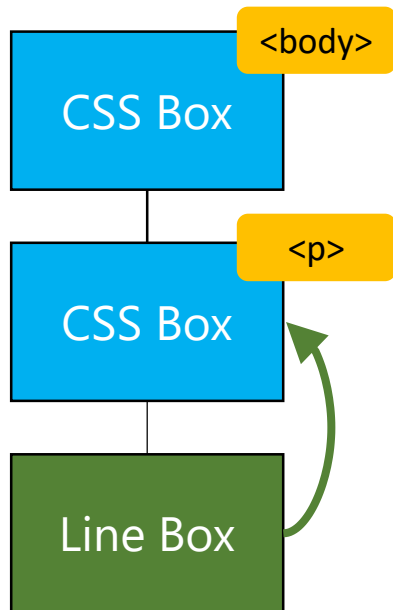
Basic Example

Box Tree



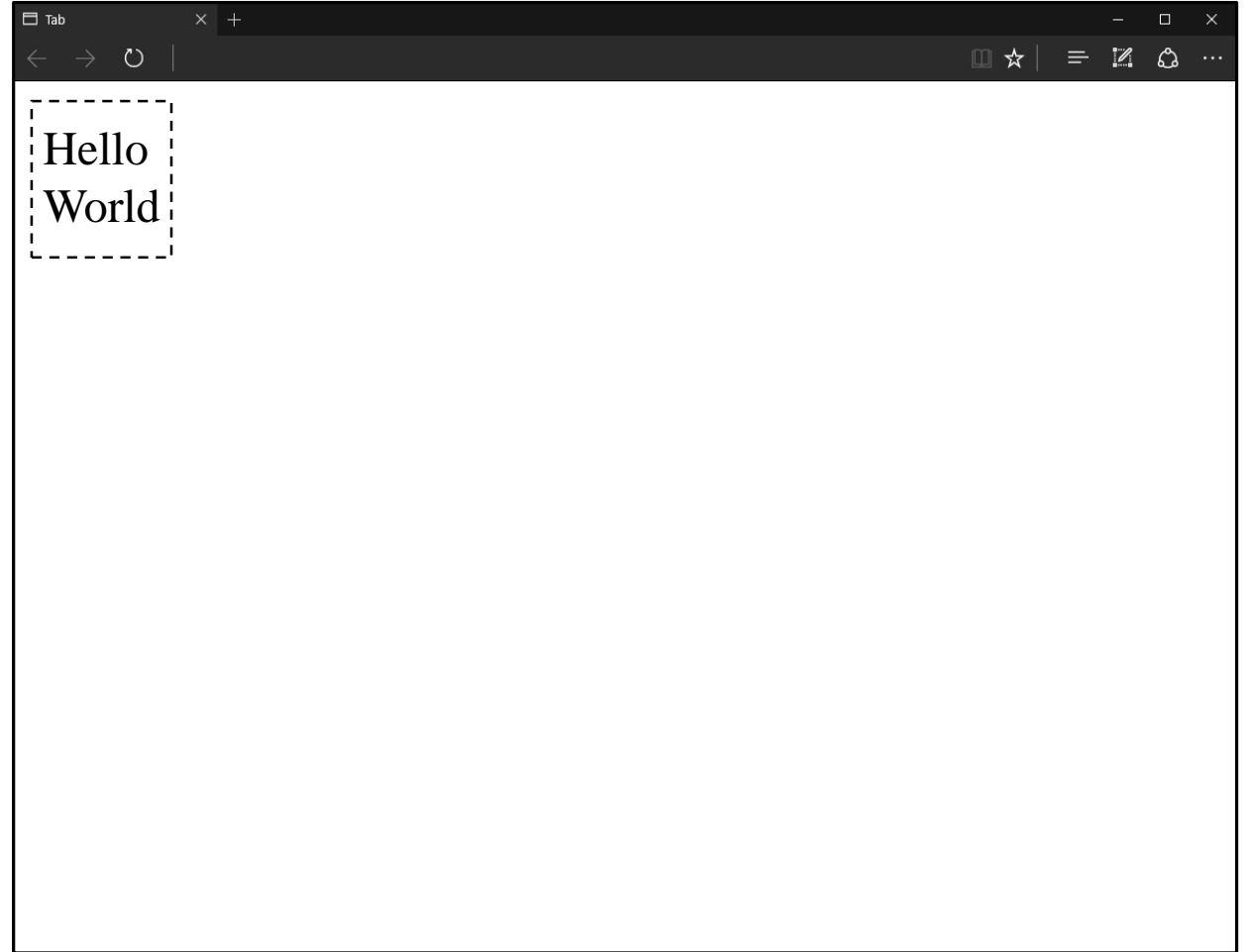
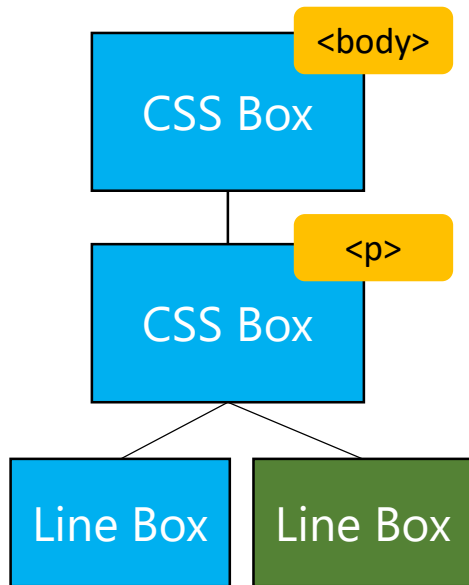
Basic Example

Box Tree



Basic Example

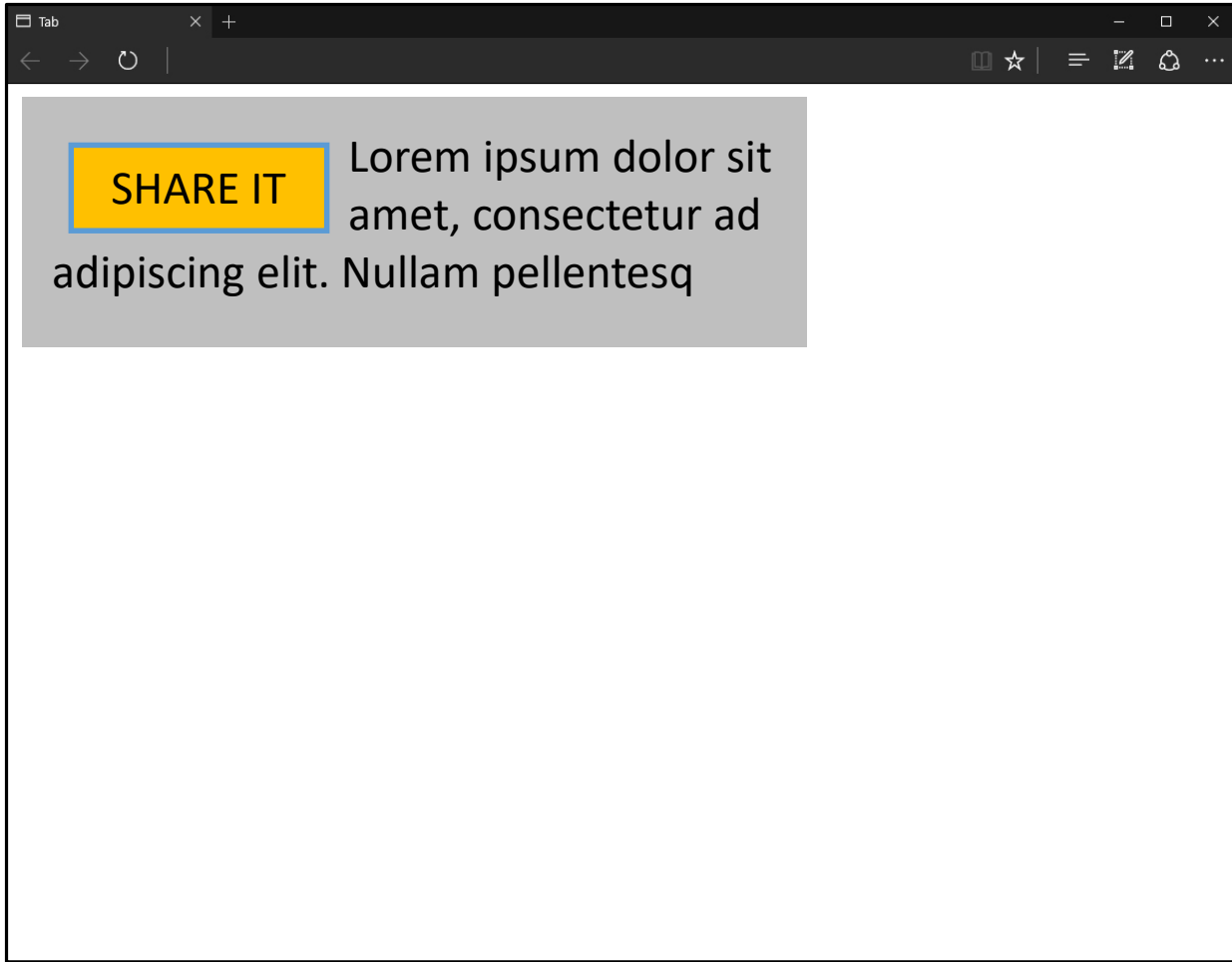
Box Tree



{layout:content measure}

Shrink-to-fit Context

When your your inline dimensions depends on its contents.



HTML

```
<article>
  <button>
    SHARE IT
  </button>
  <p>
    Lorem ipsum dolor sit amet,
    consectetur adipiscing elit. Nullam
    pellentesq
  </p>
</article>
```

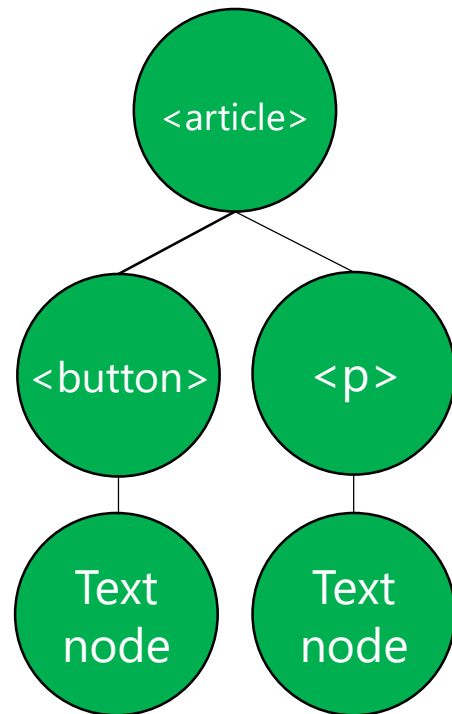
CSS

```
article {
  min-width: 400px;
  max-width: 800px;
  background: rgb(191,191,191);
  padding: 5px;
}

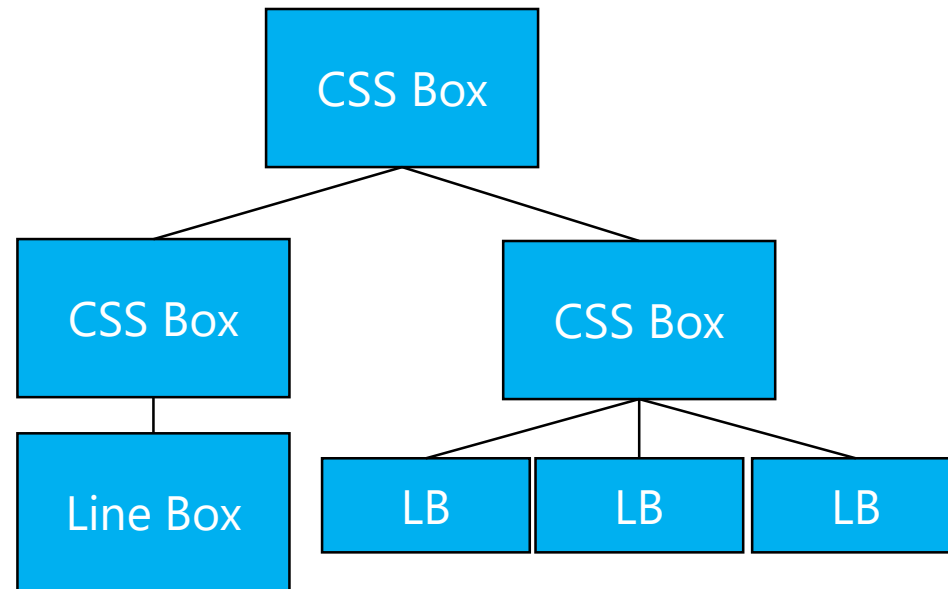
button {
  float: left;
  background: orange;
  padding: 3px 10px;
  border: 2px solid blue;
  margin: 5px;
}

p {margin: 0;}
```

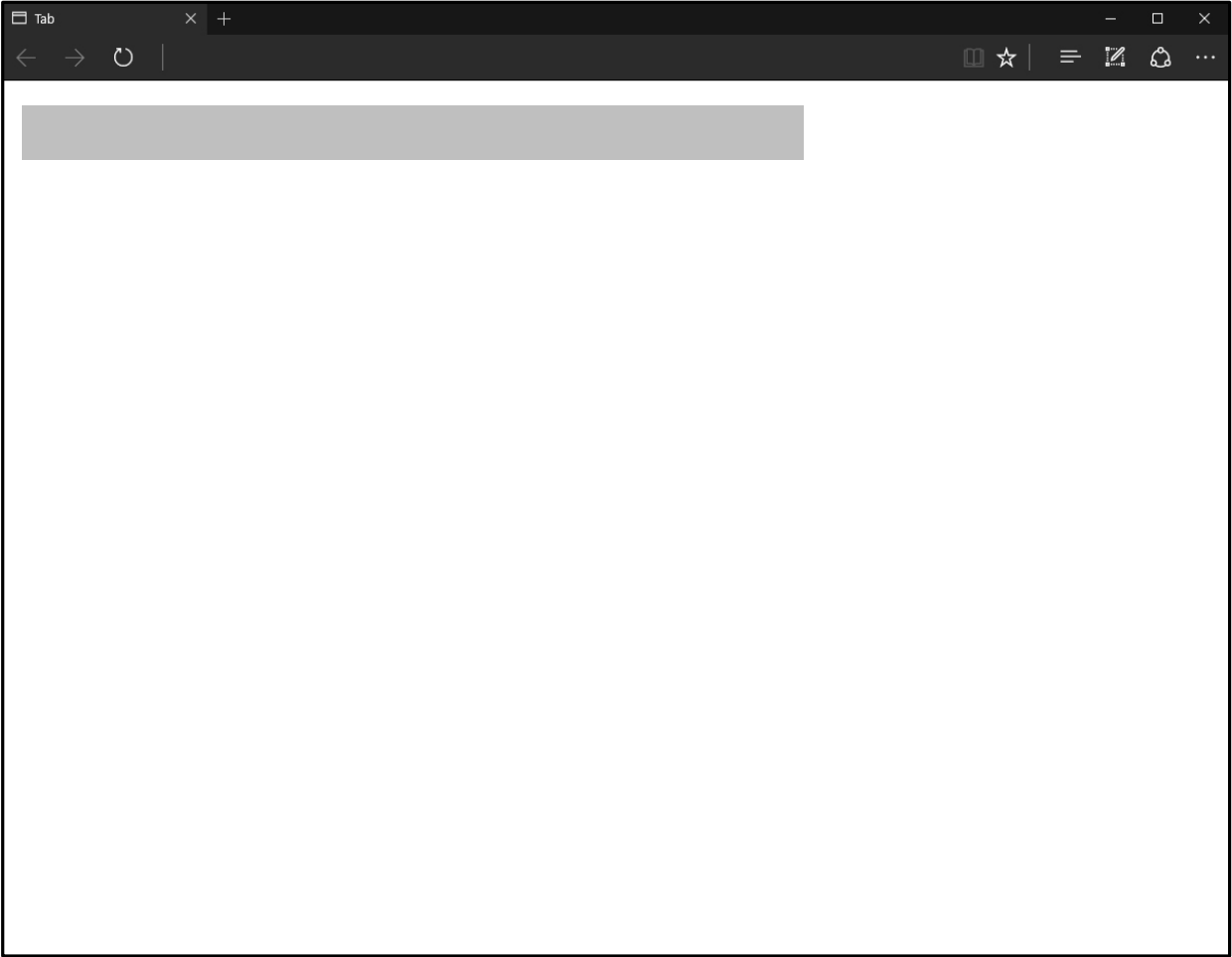

DOM Tree



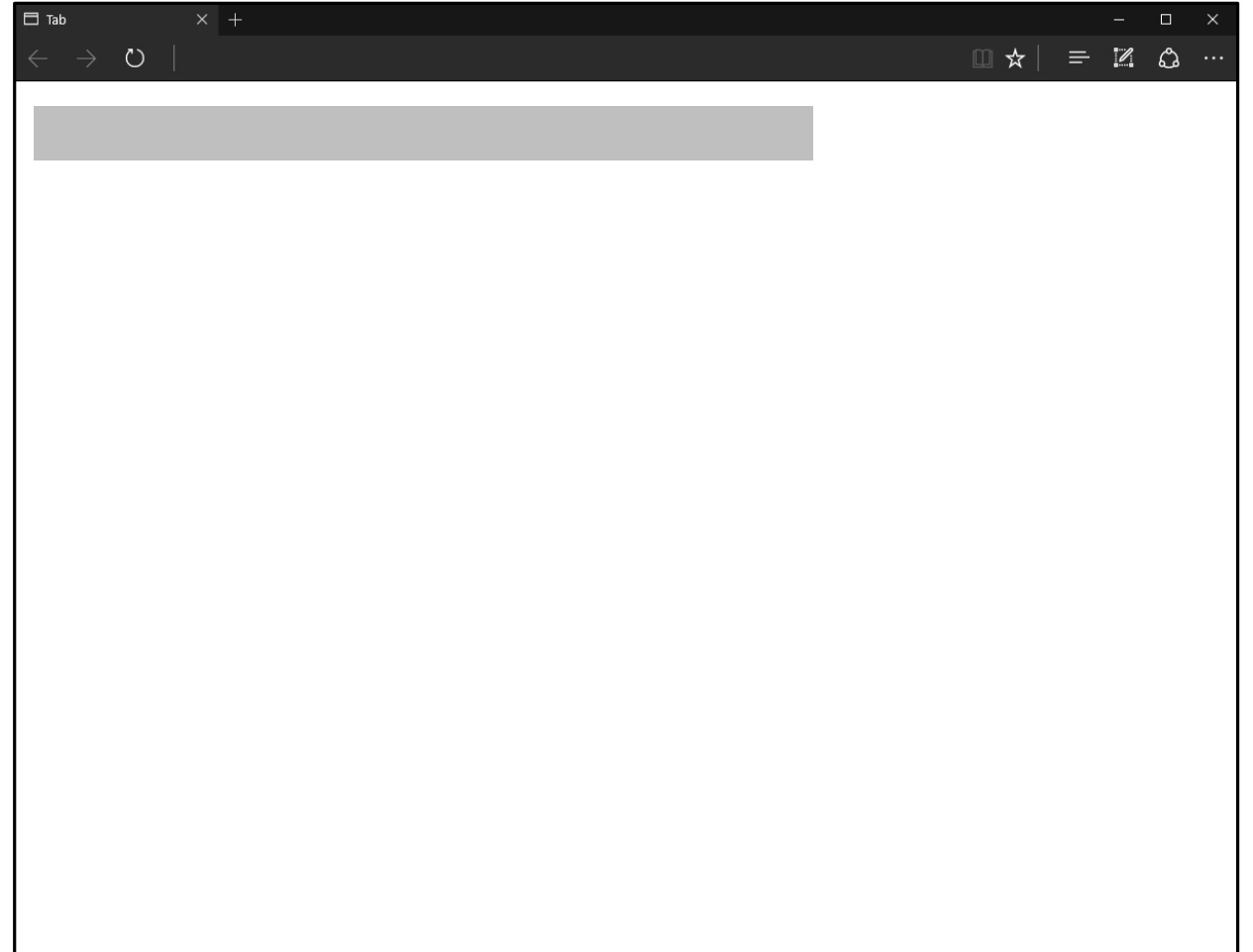
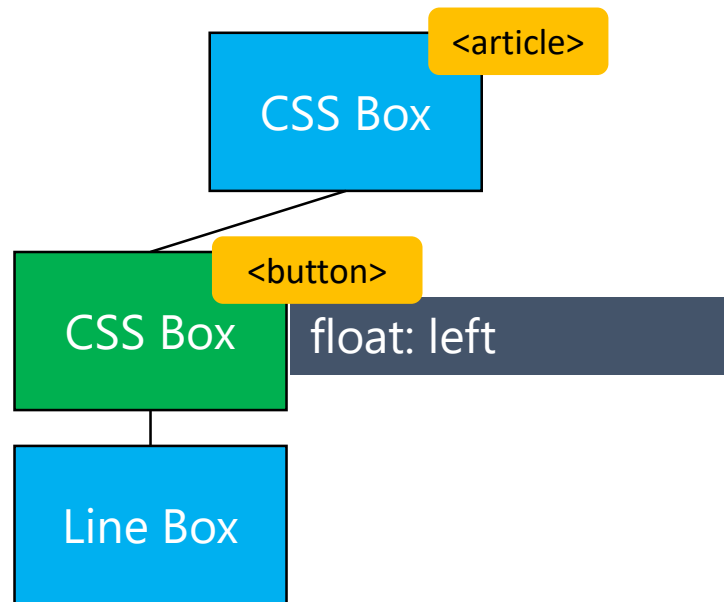
Box Tree



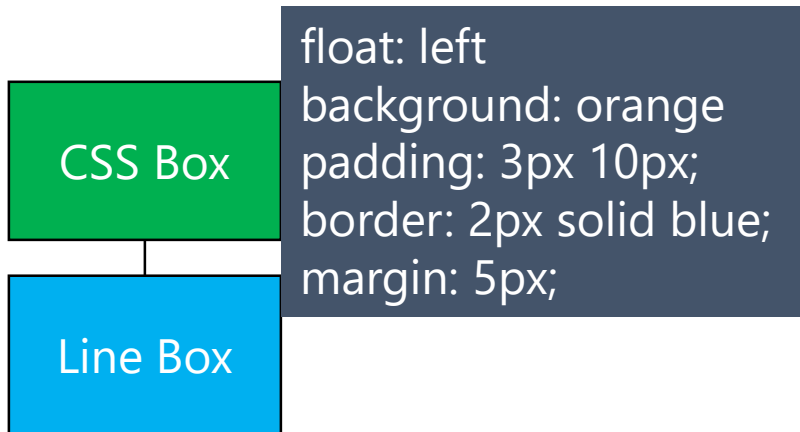
Box Tree



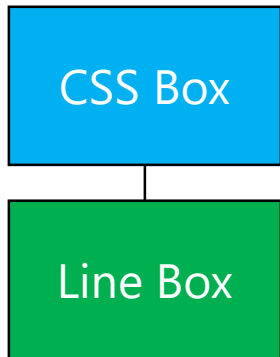
Box Tree



Box Tree



Box Tree



115px



Max Width

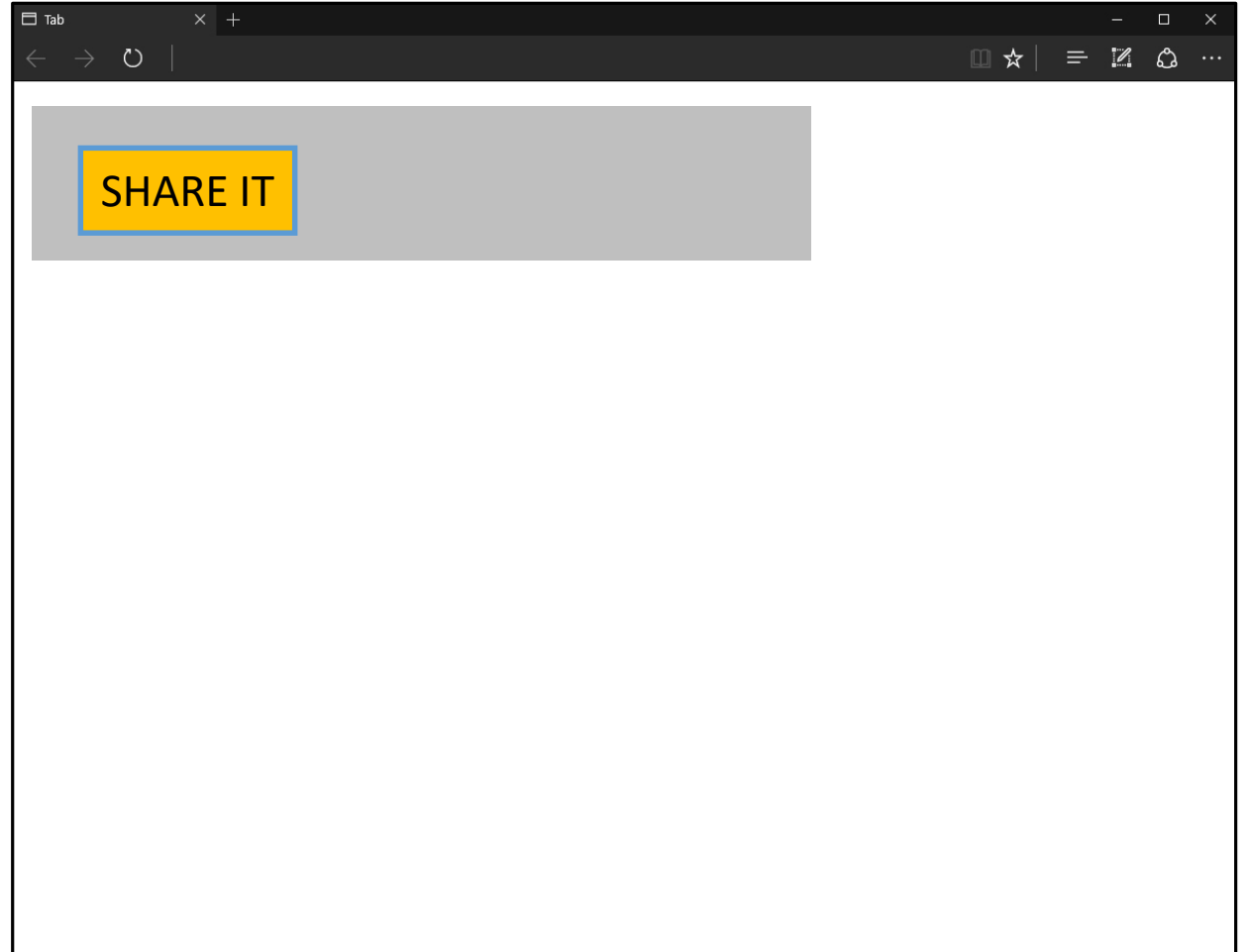
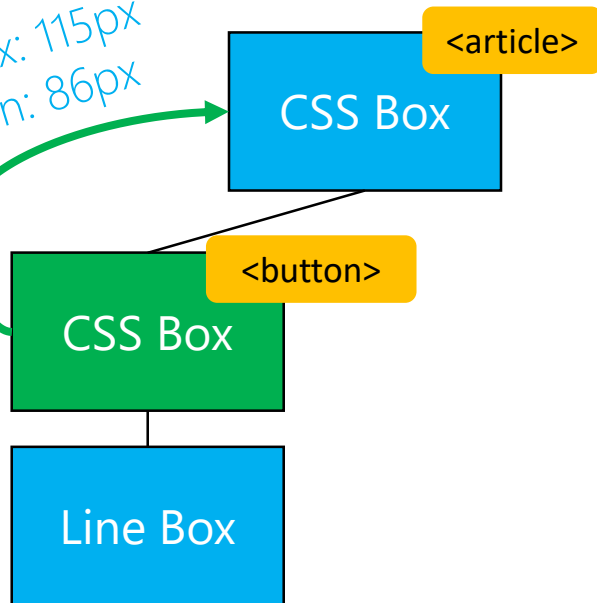
86px



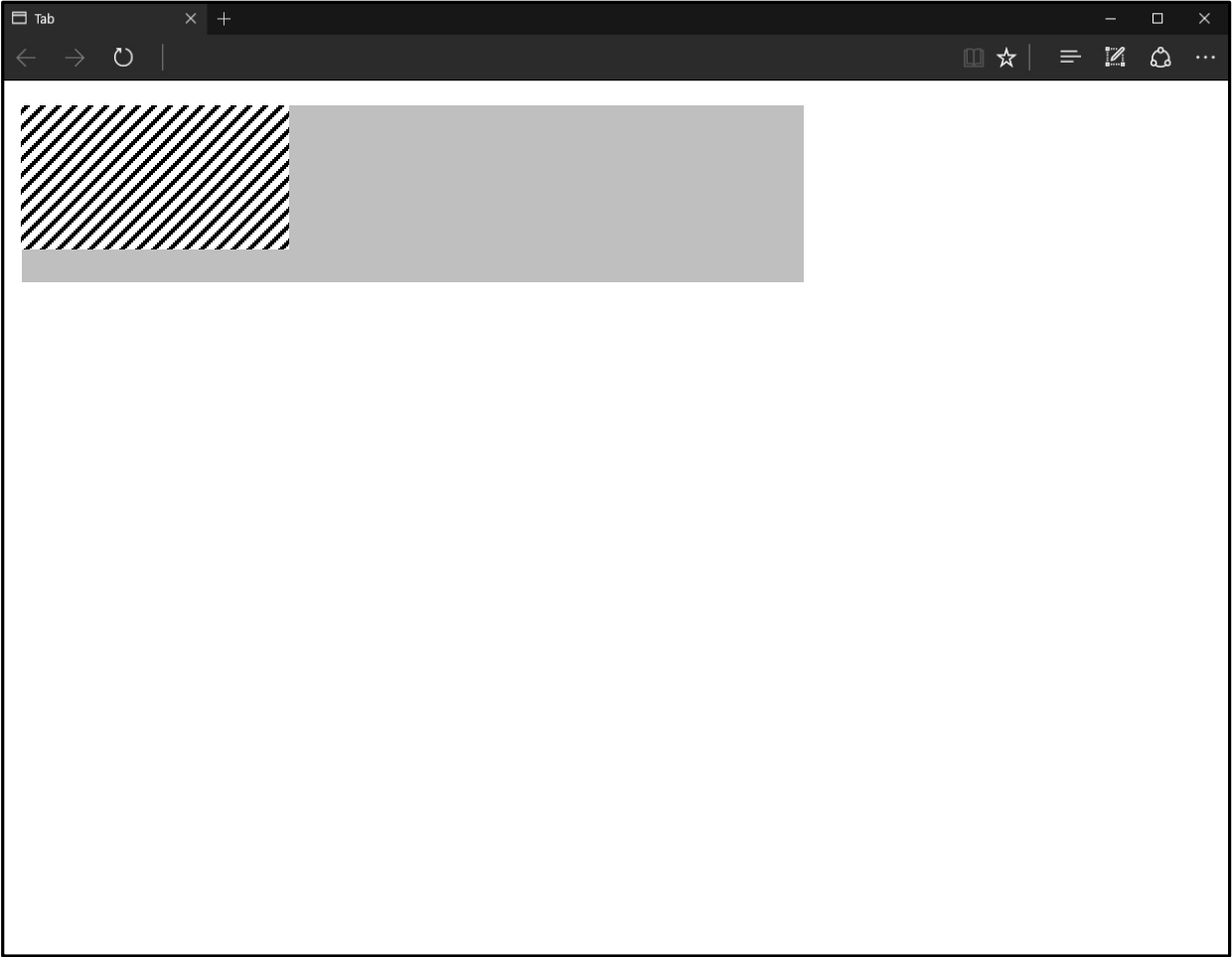
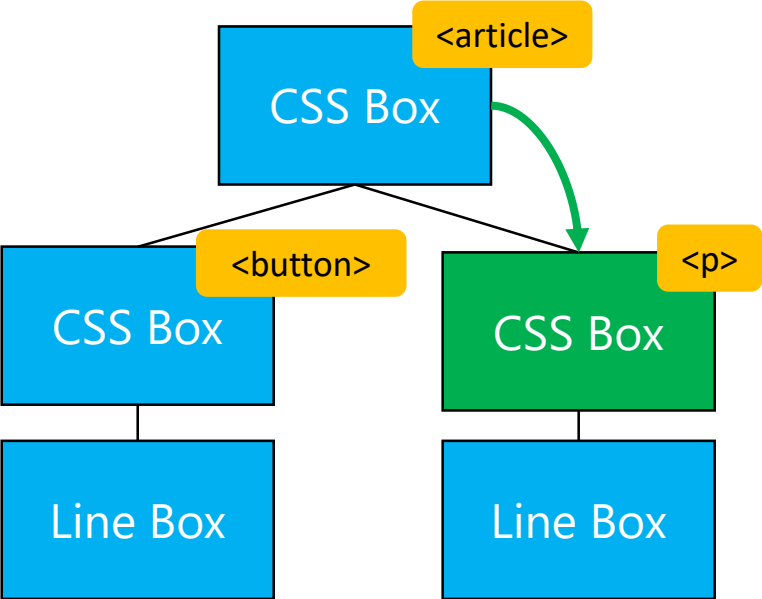
Min Width

Box Tree

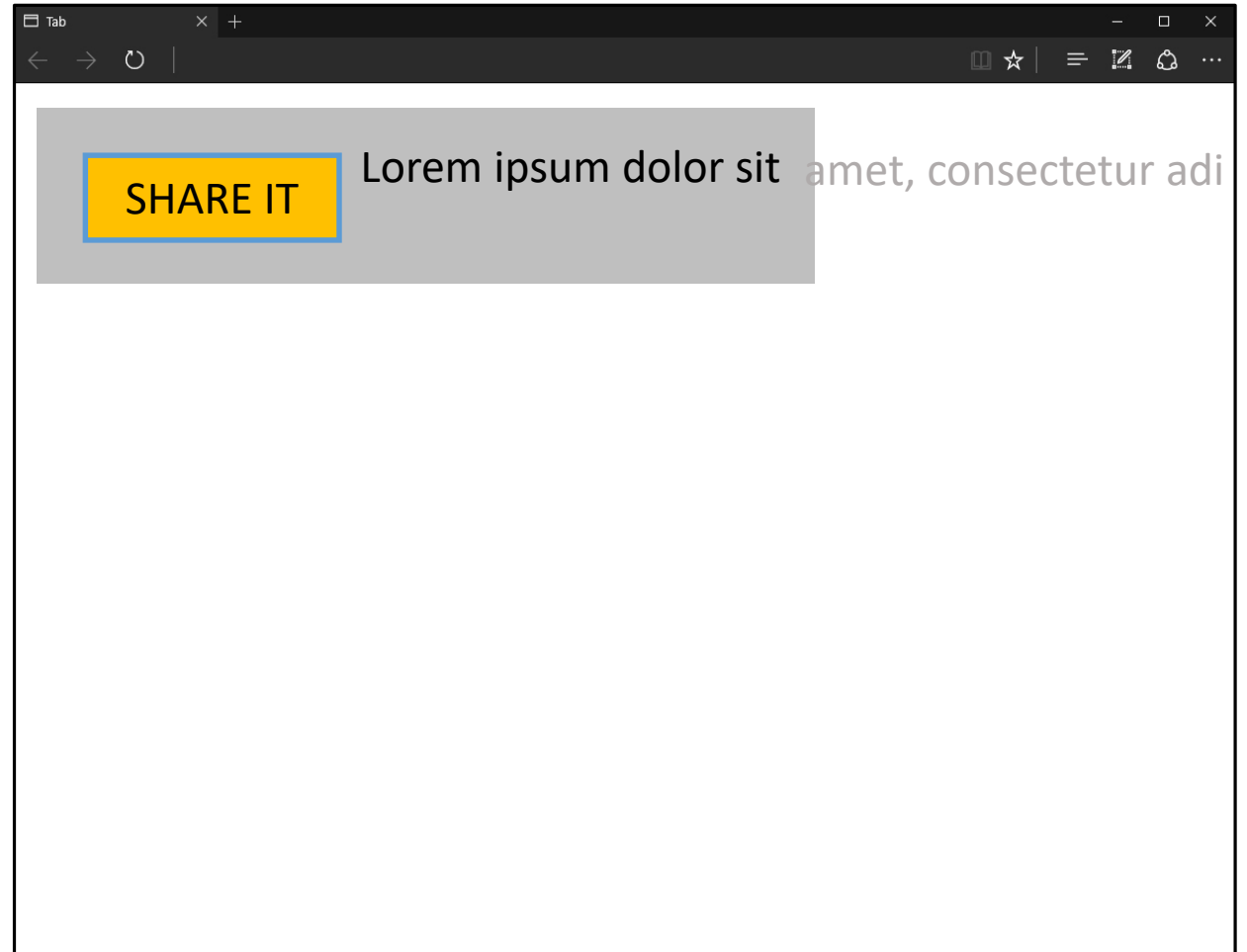
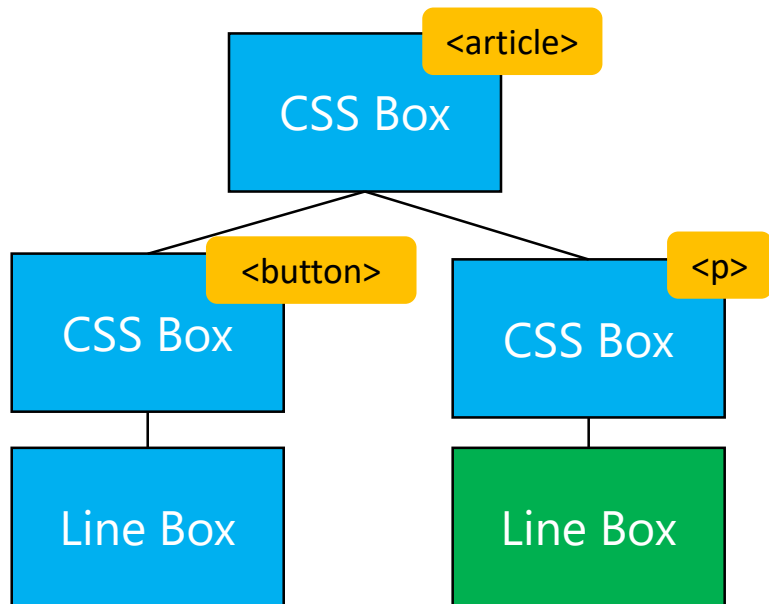
max: 115px
min: 86px



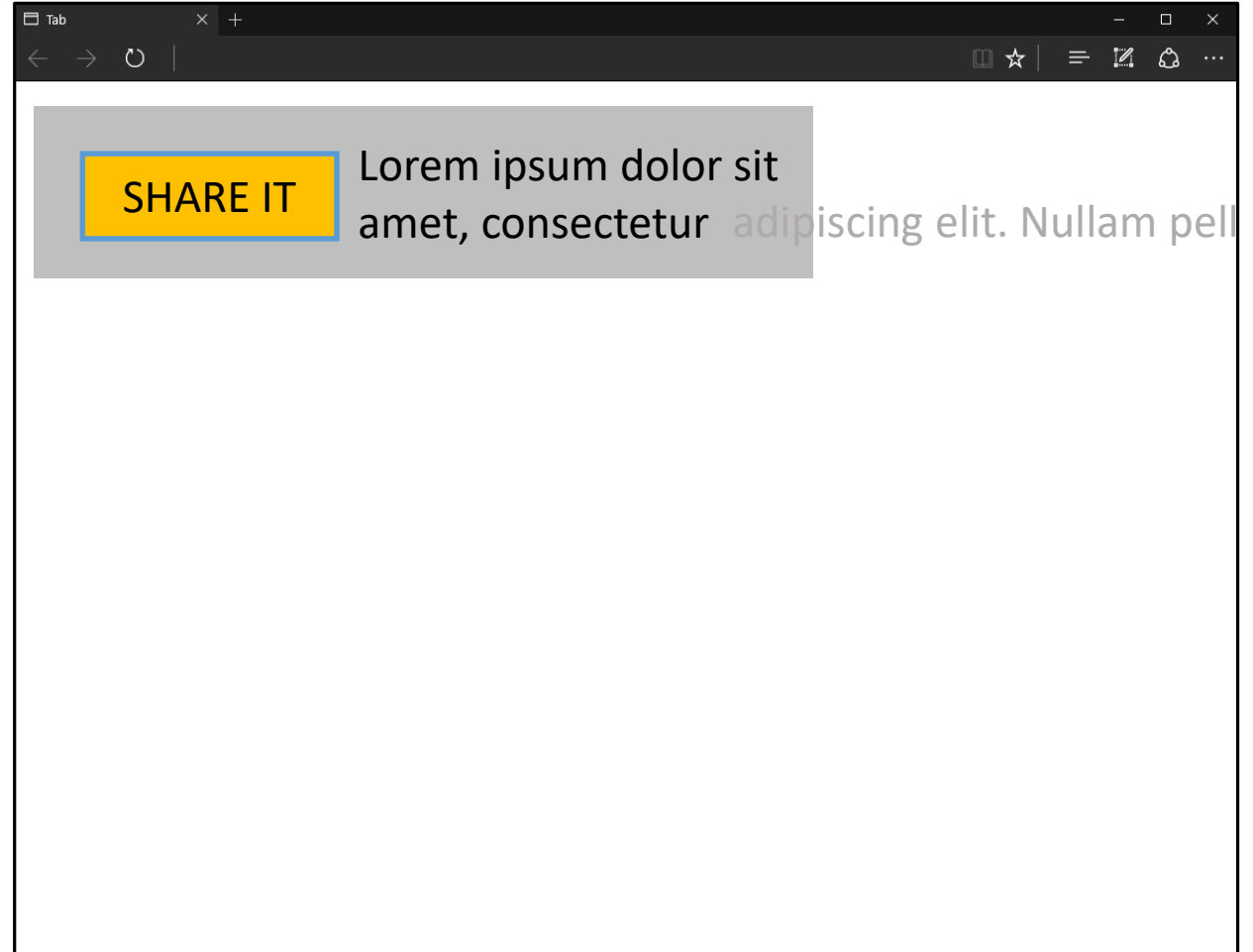
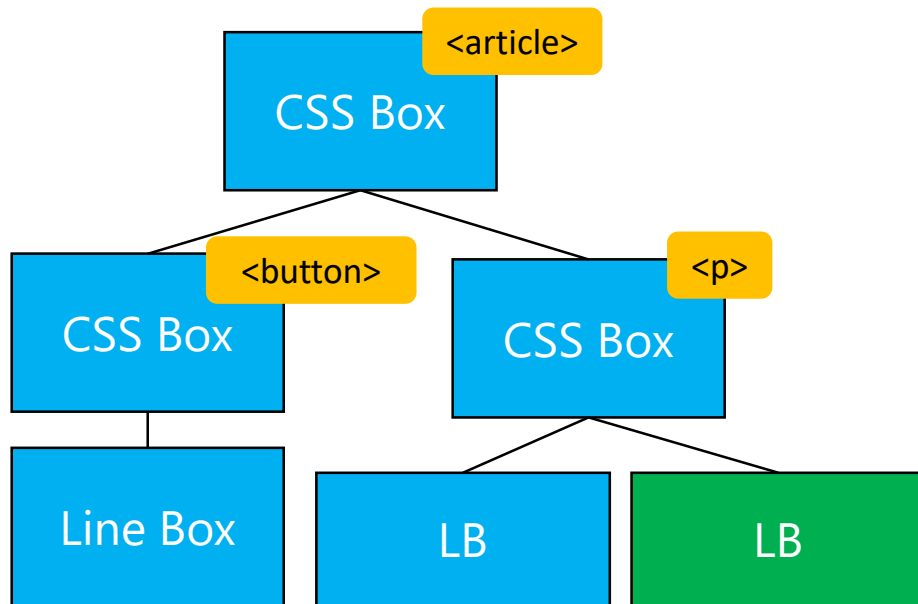
Box Tree



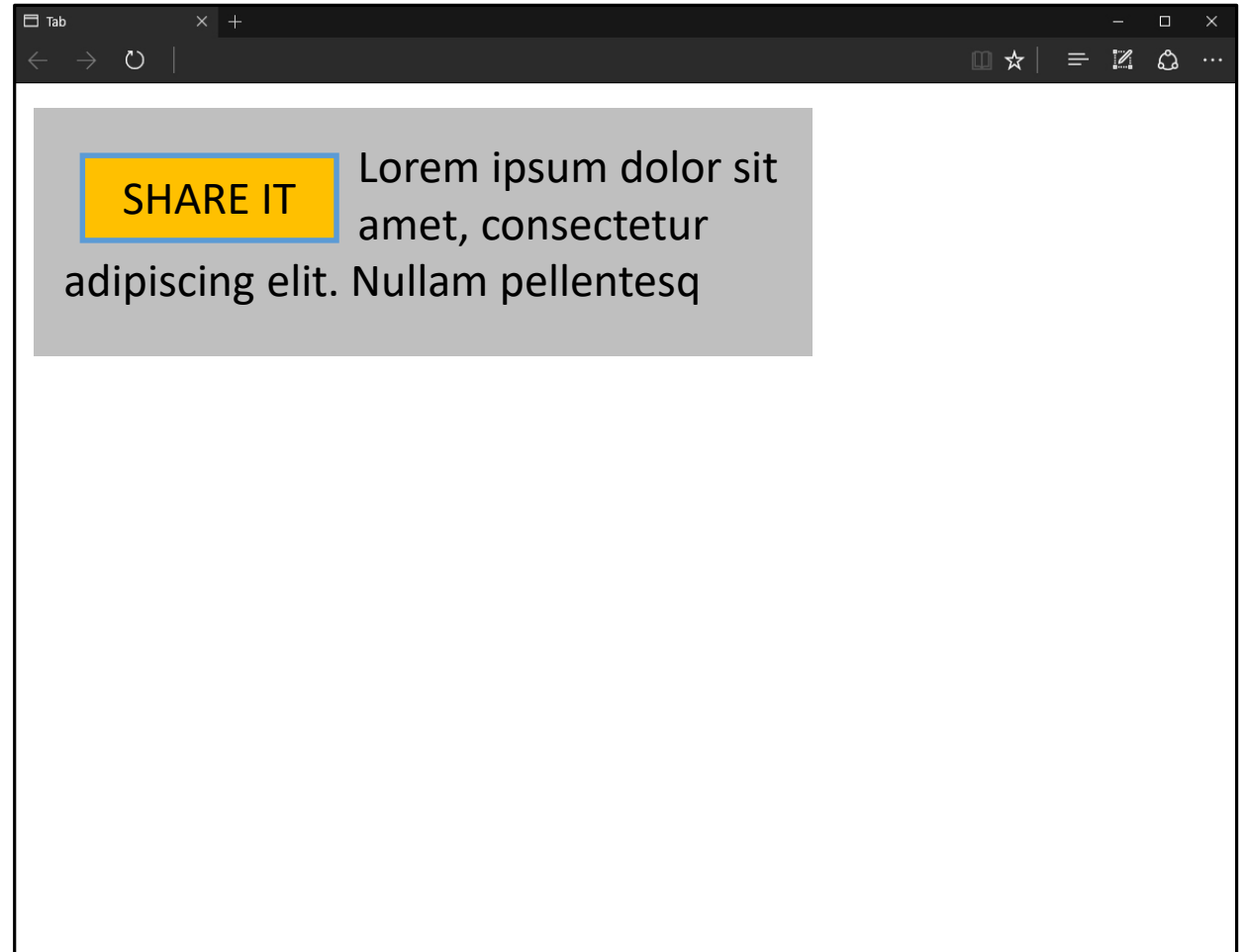
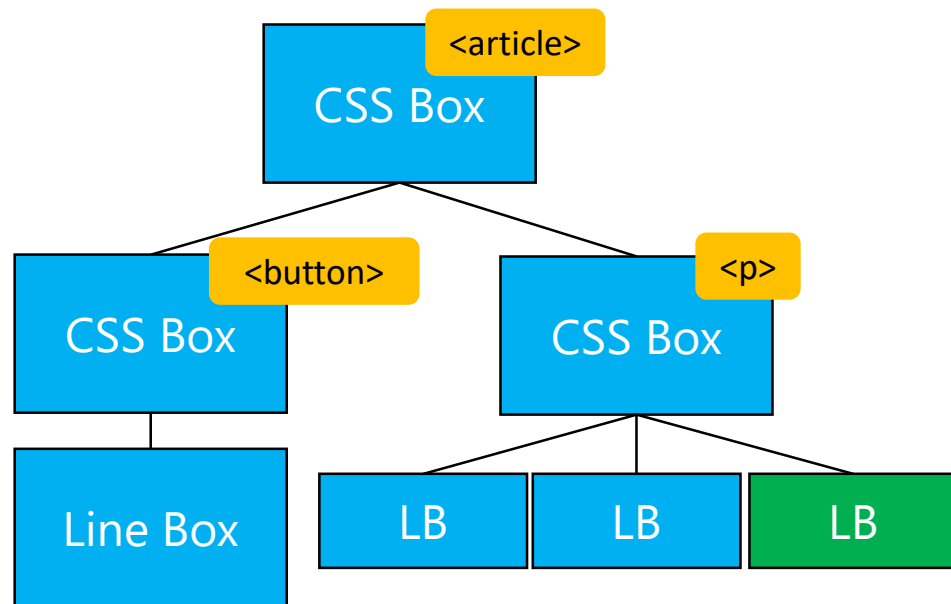
Box Tree



Box Tree



Box Tree



{layout:fragmentation}

Fragmentainer

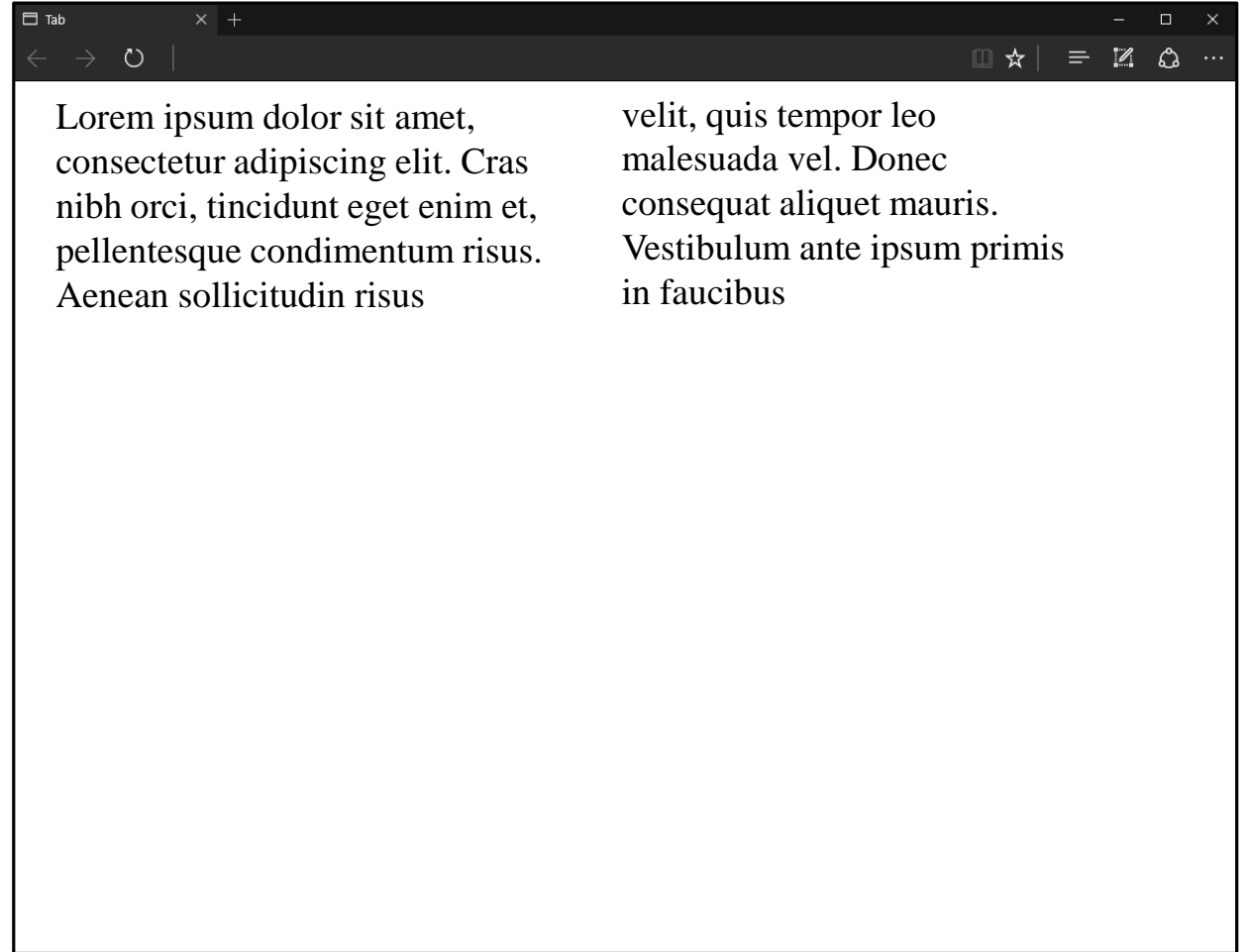
A box—such as a page box, column box, or region—that contains a portion (or all) of a fragmented flow.

HTML

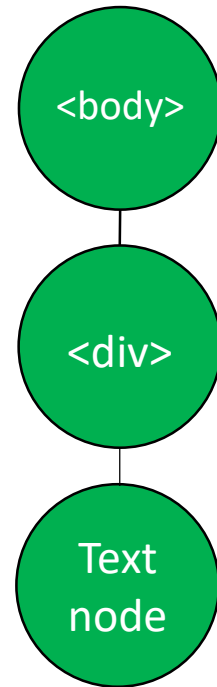
```
<body>
  <div>
    Lorem ipsum dolor sit amet,
    consectetur adipiscing elit. Cras nibh
    orci, tincidunt eget enim et,
    pellentesque condimentum risus. Aenean
    sollicitudin risus velit, quis tempor
    leo malesuada vel. Donec consequat
    aliquet mauris. Vestibulum ante ipsum
    primis in faucibus
  </div>
</body>
```

CSS

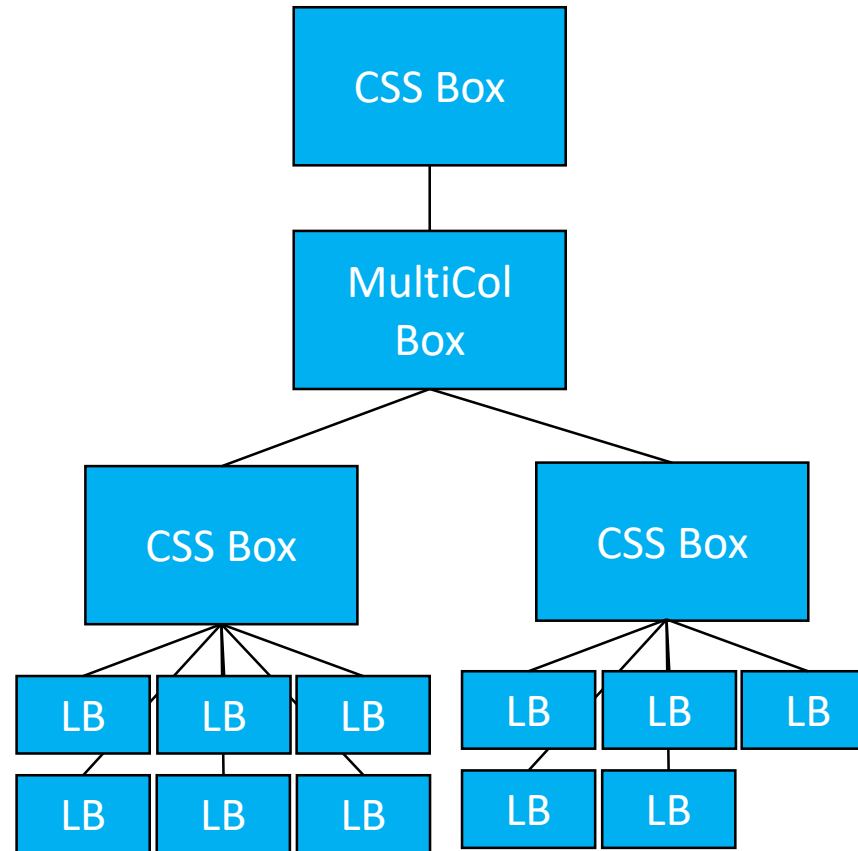
```
div {
  columns: 2;
  column-fill: auto;
  height: 300px;
}
```



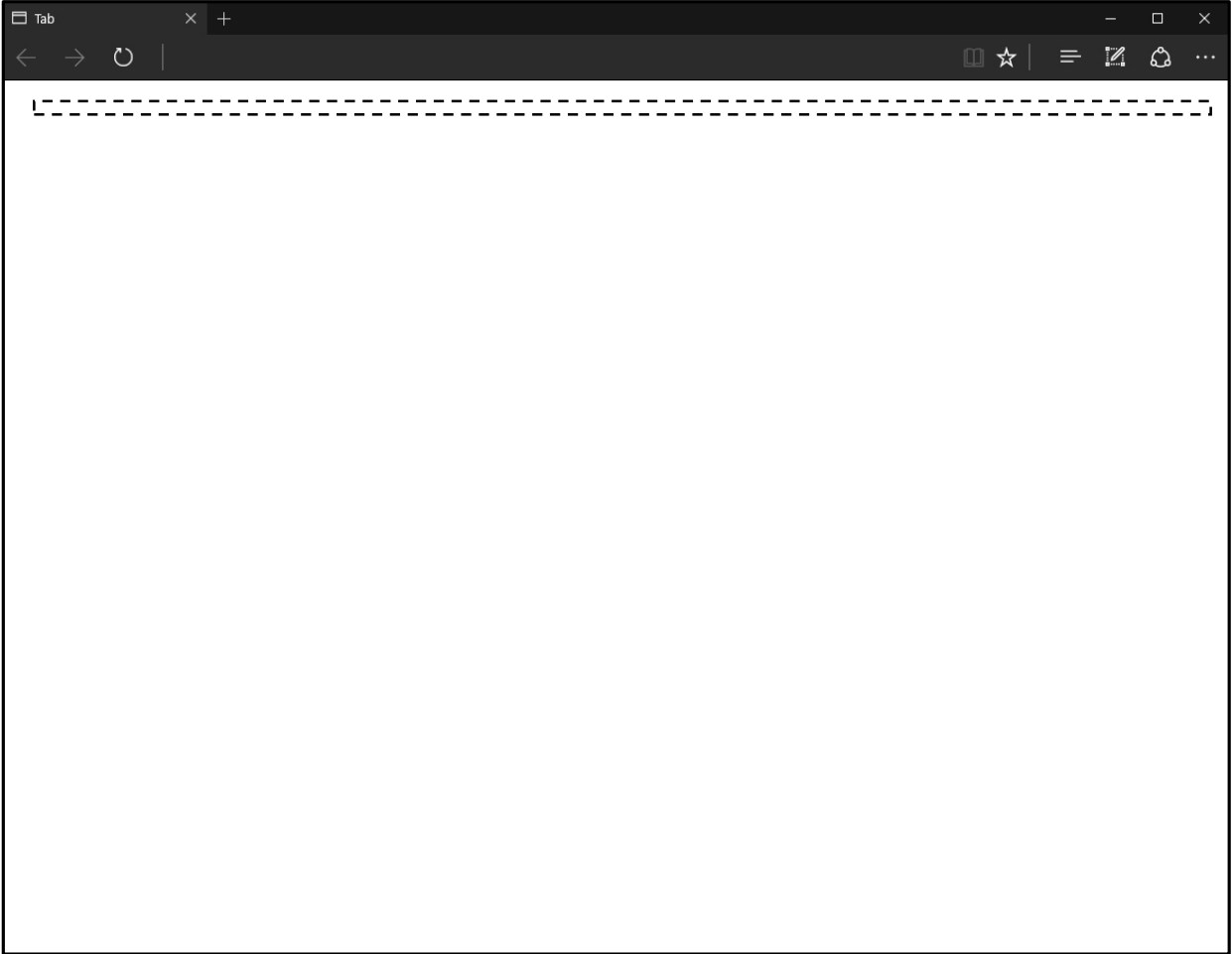
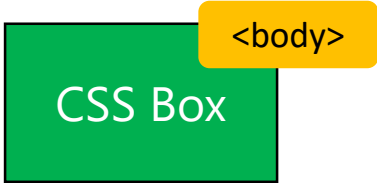
DOM Tree



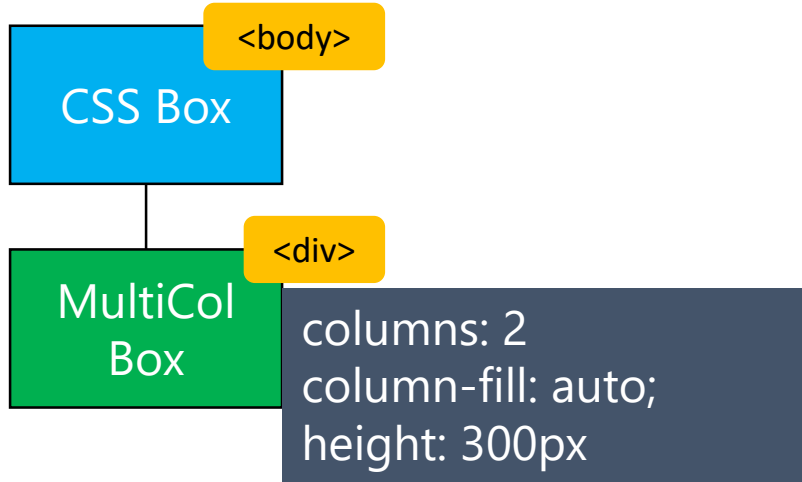
Box Tree



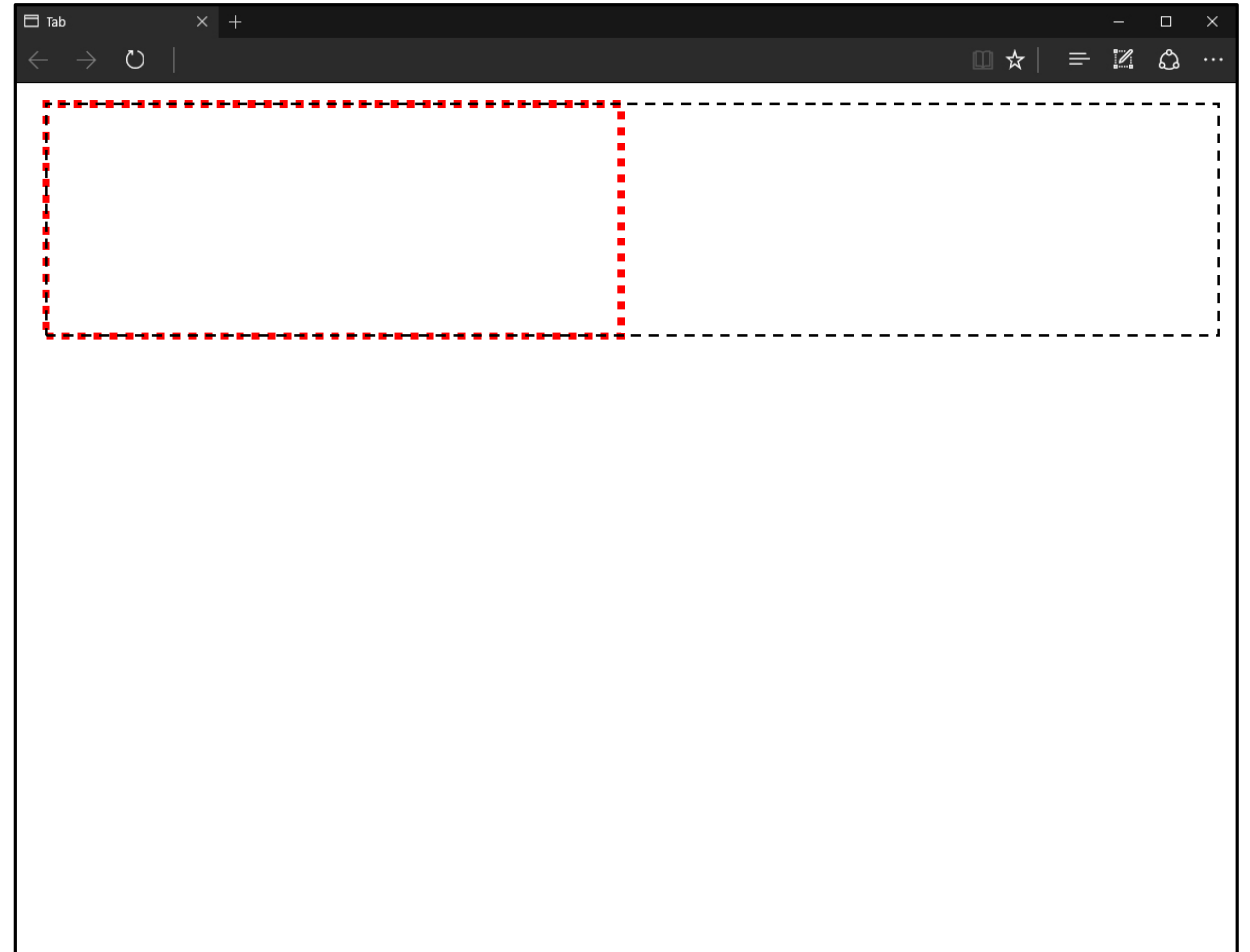
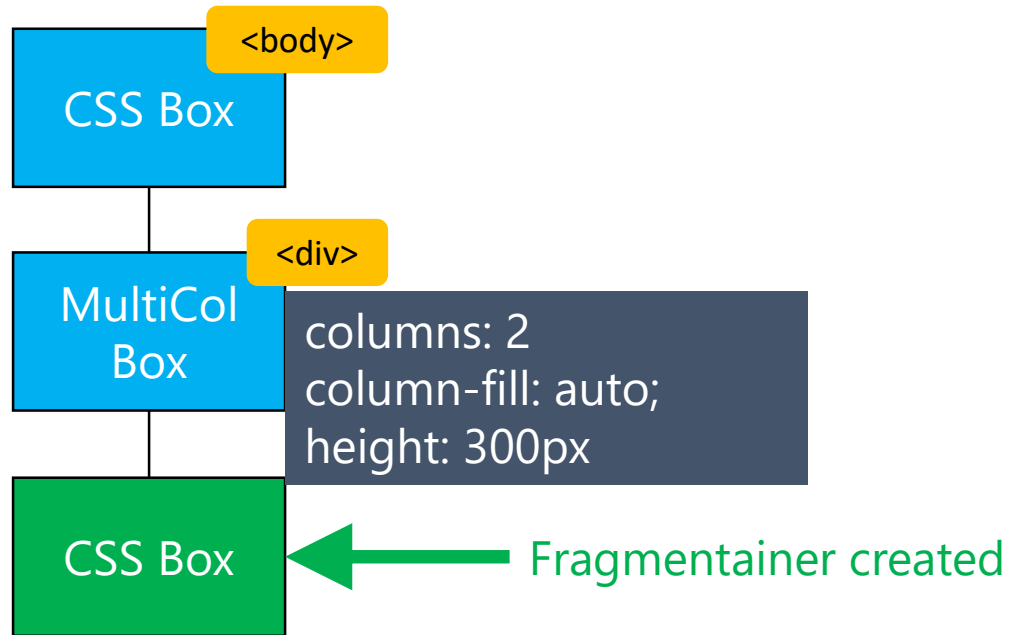
Box Tree



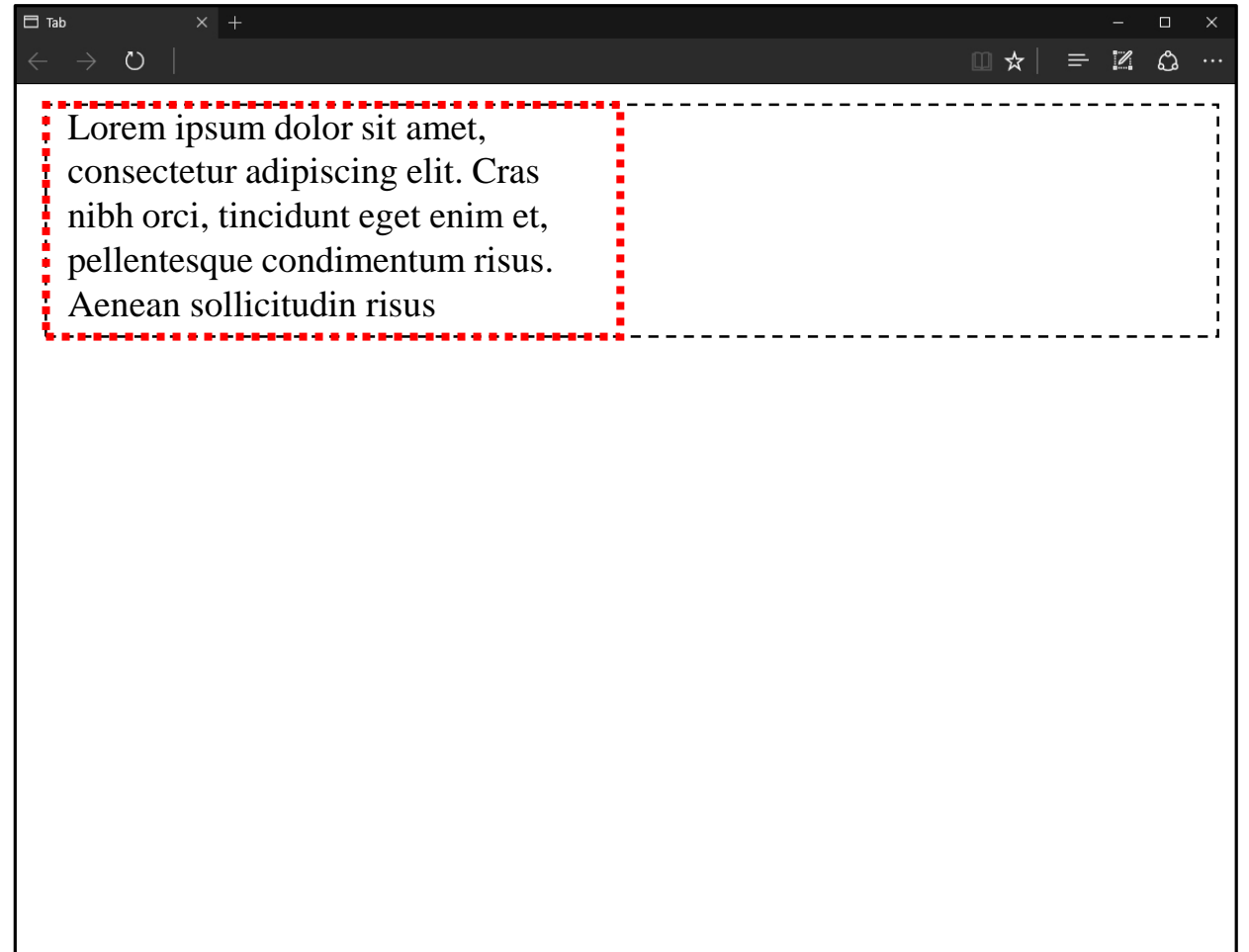
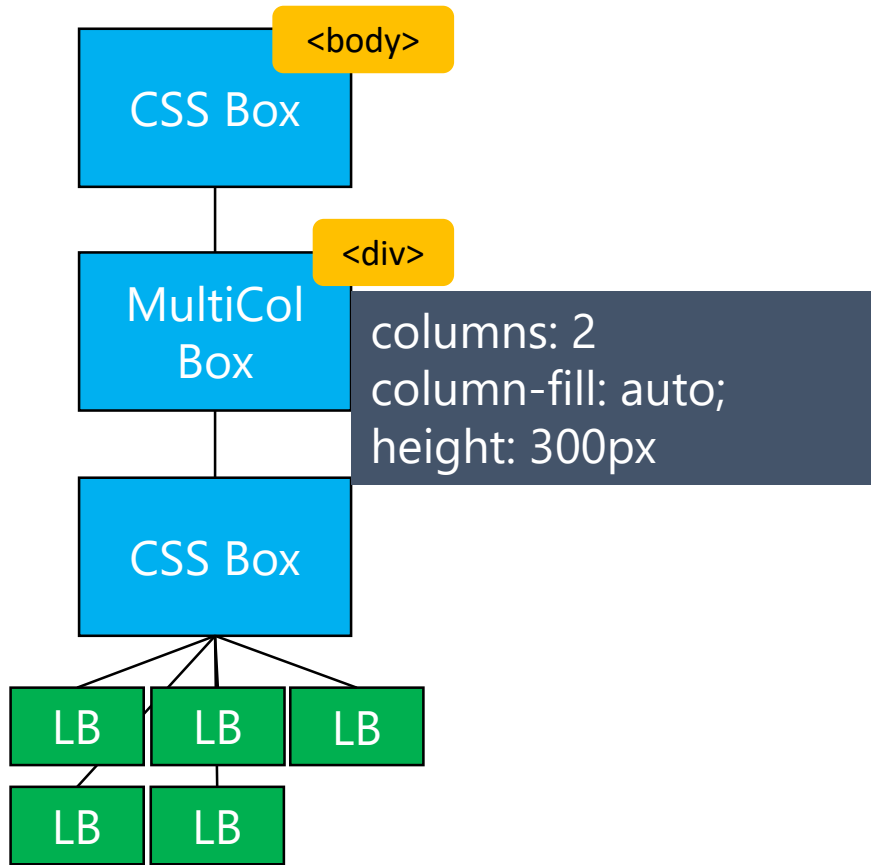
Box Tree



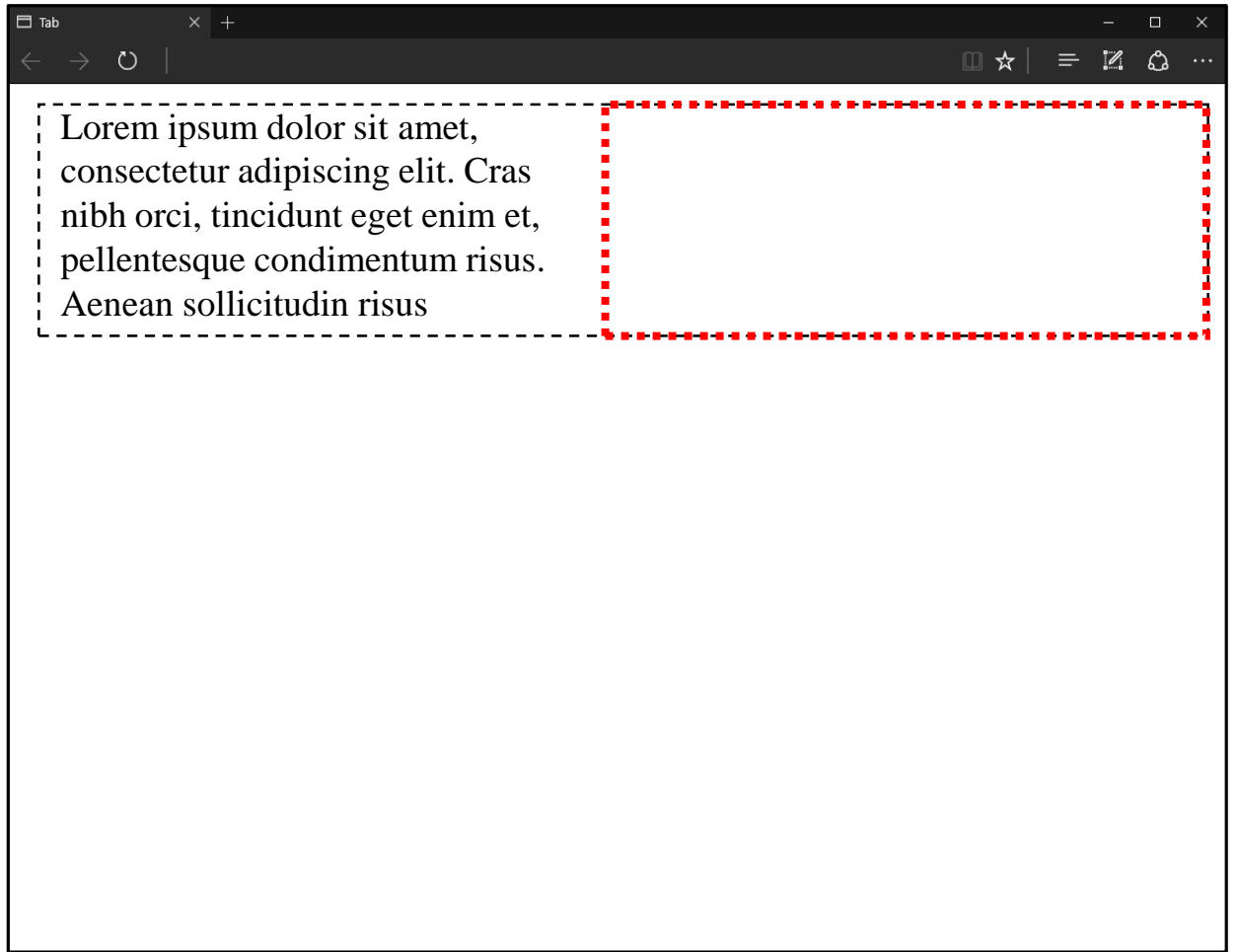
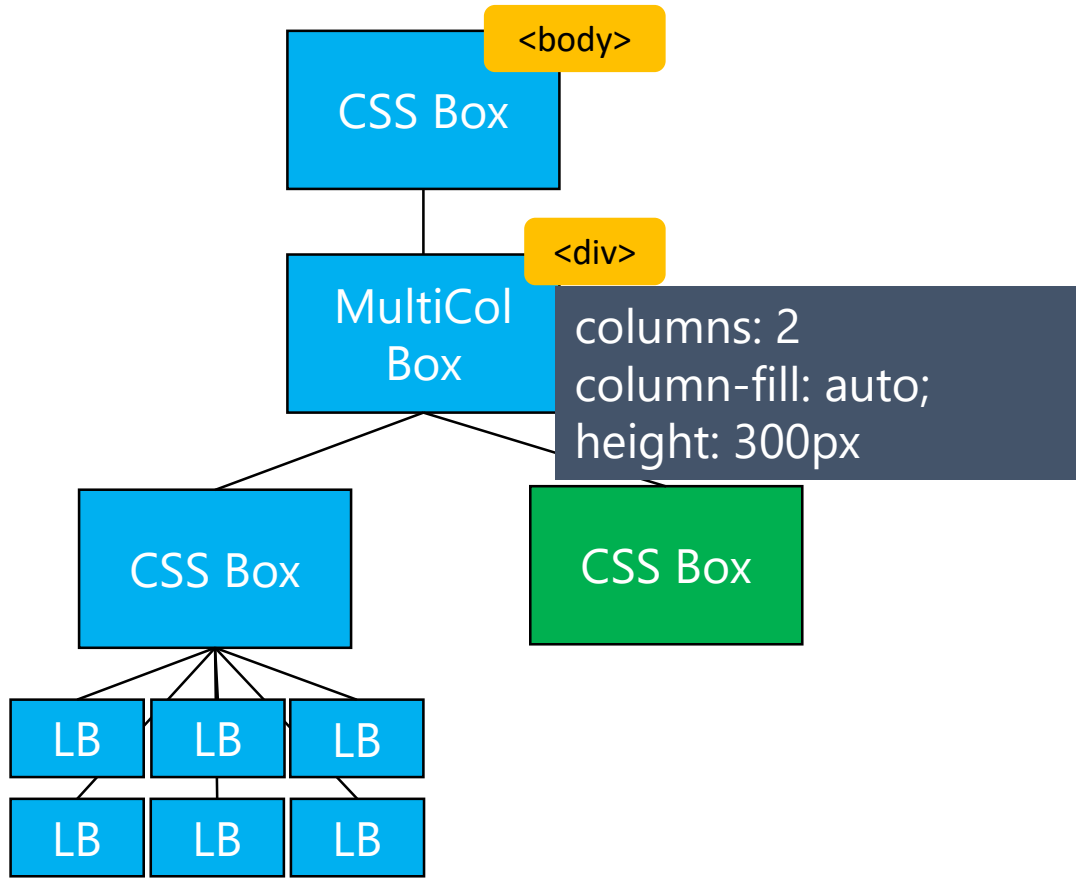
Box Tree



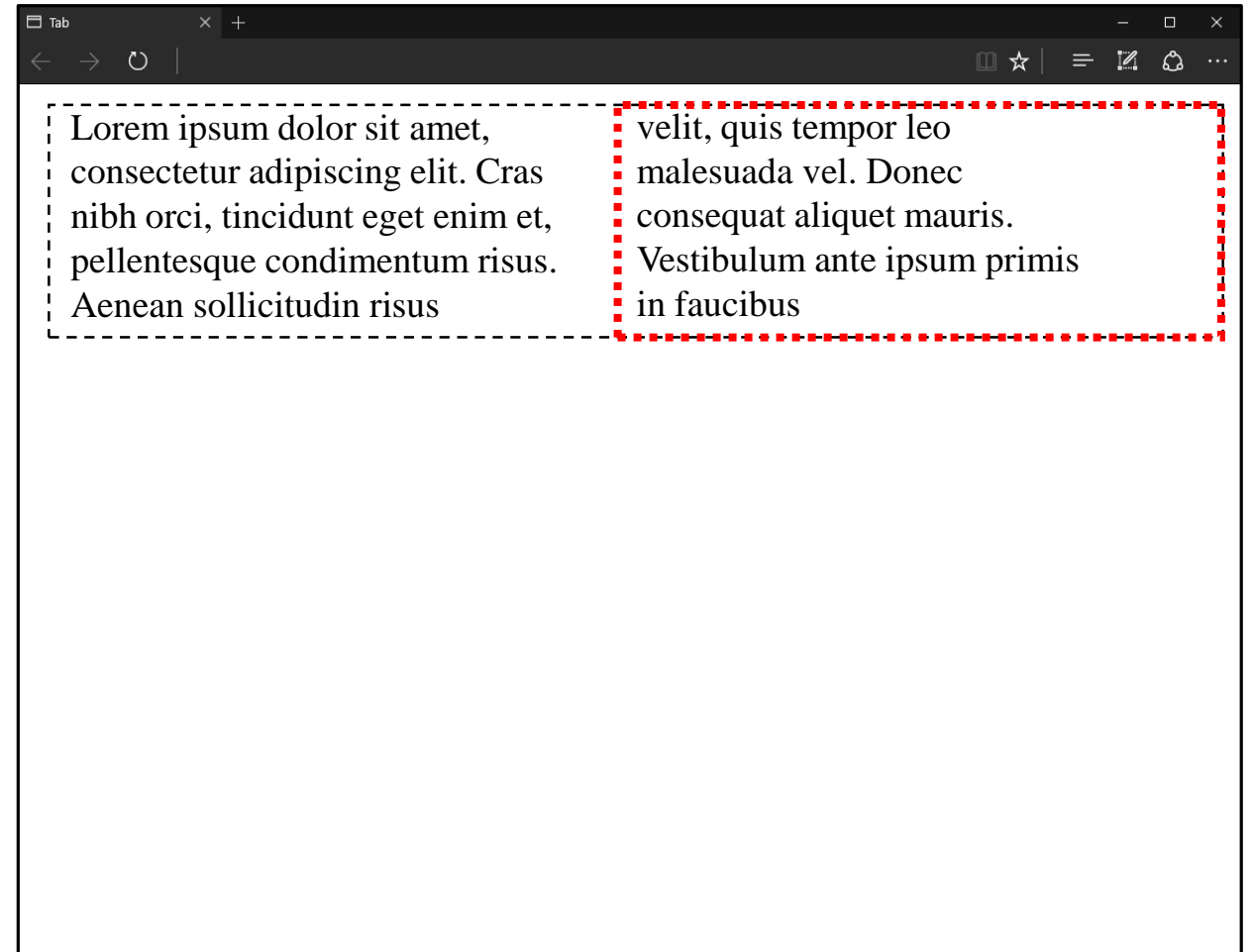
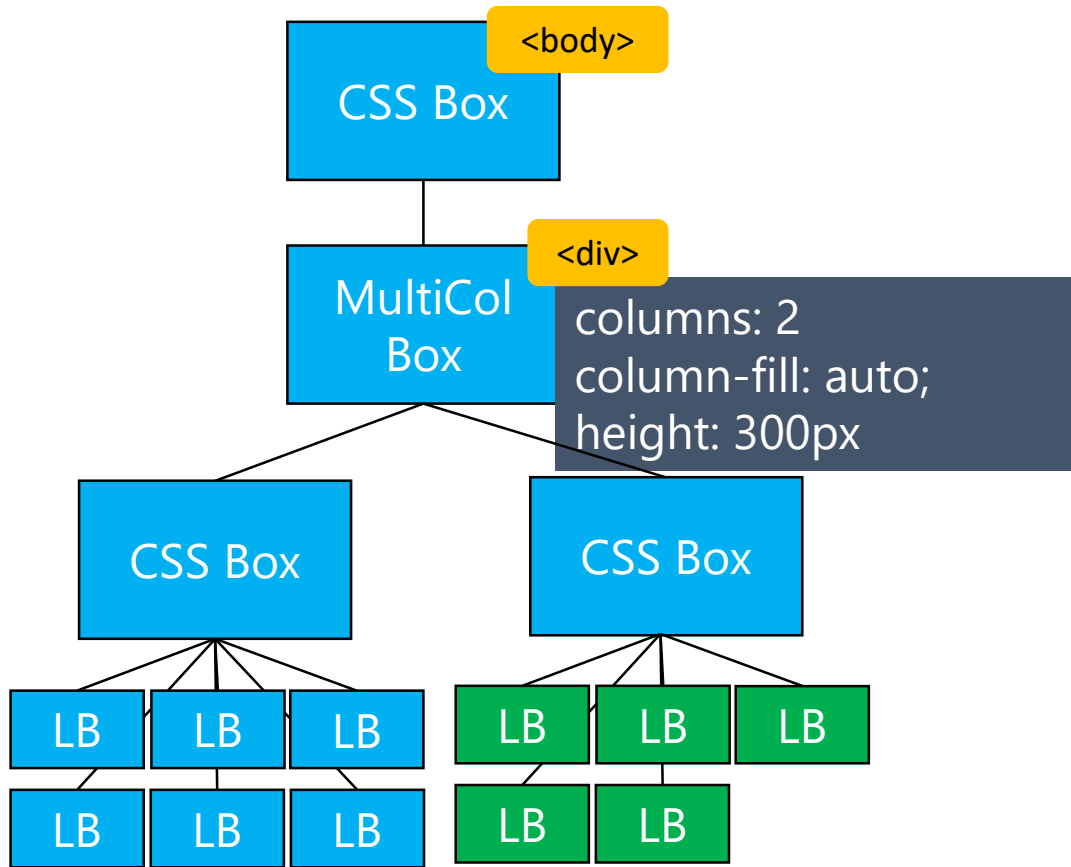
Box Tree



Box Tree



Box Tree



{paint}

Network

Content
Parsing

DOM

Style

Layout

Paint

Compose

YOU
ARE
HERE

SHARE IT

Step 1: Paint the elements background



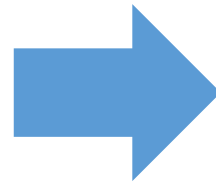
Step 2: Paint the element's border



Step 3: Paint the element's content



SHARE IT



Stacking Context (SC)

The order in which the rendering tree is painted onto the canvas is described by stacking contexts

Stacking Context (SC)

Actually it
sort of does
though?



Stacking context \neq layers

Stacking Context (SC)



HTML

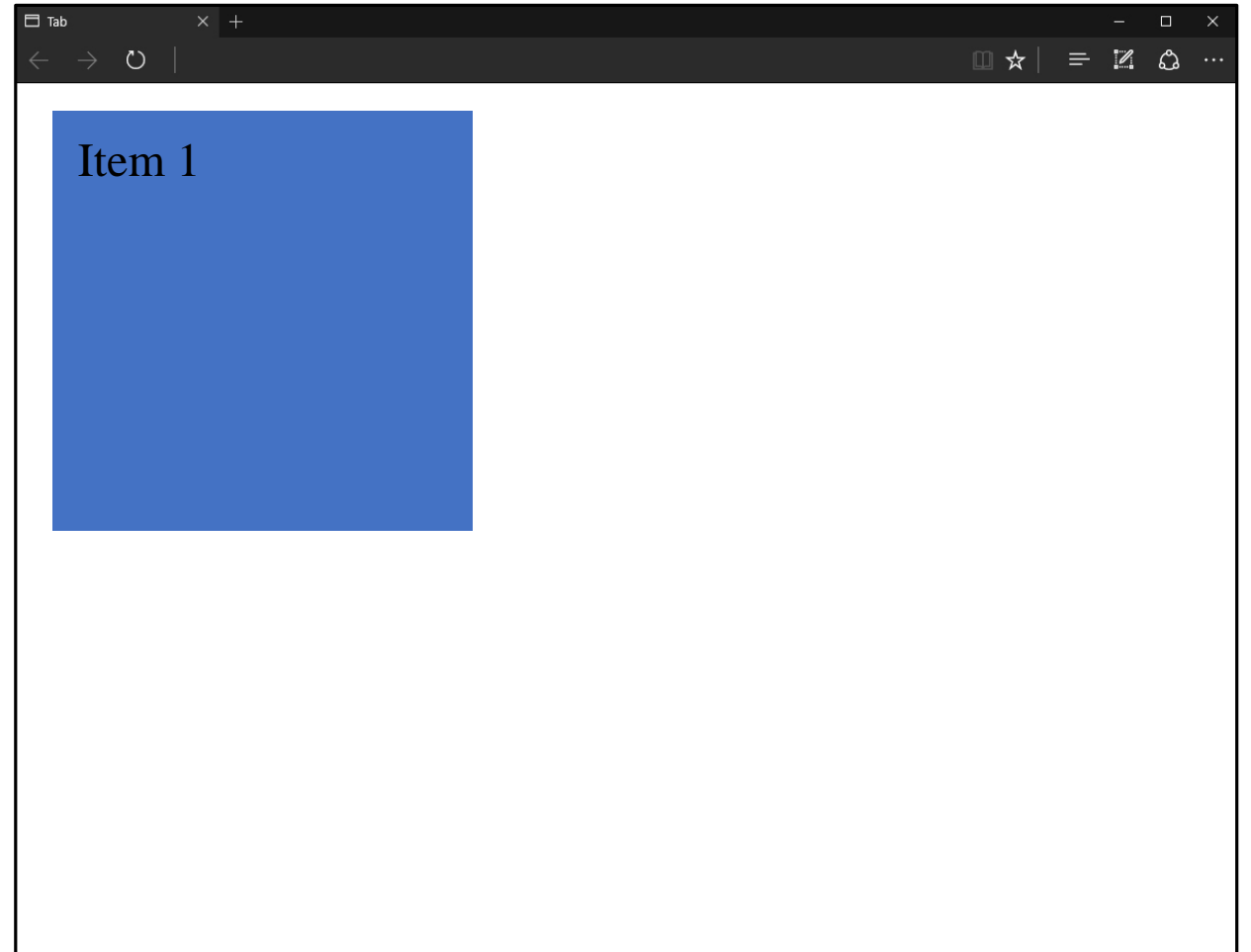
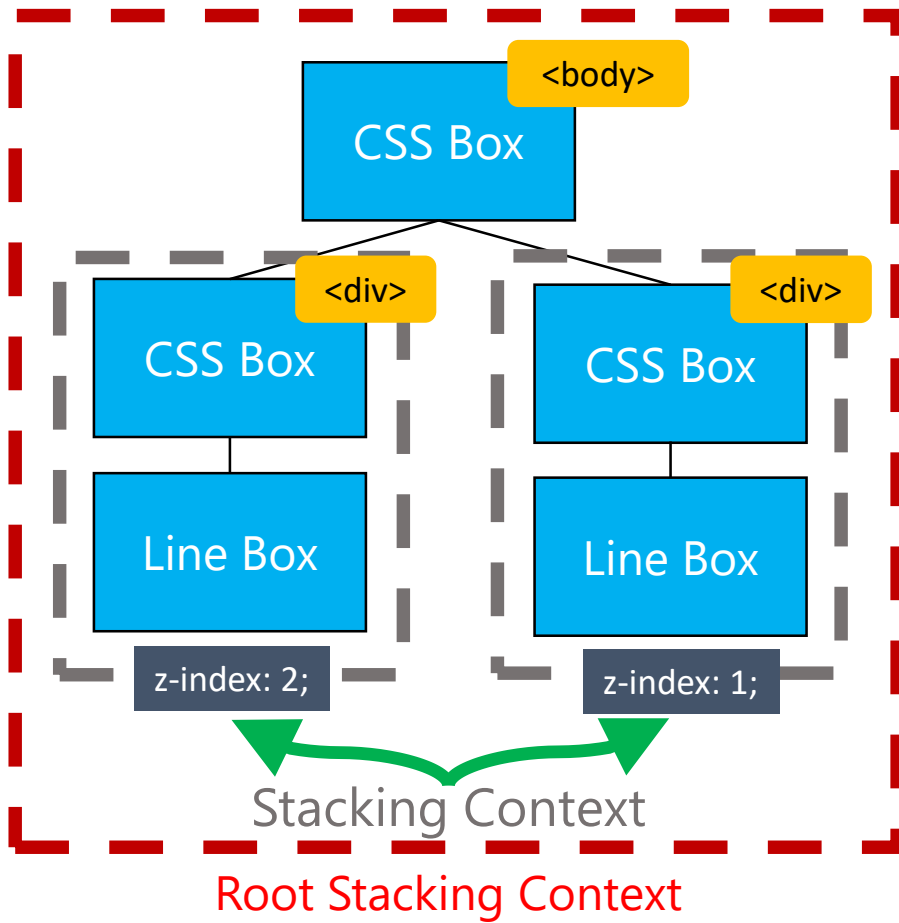
```
<body>
  <div id="one">
    Item 1
  </div>
  <div id="two">
    Item 2
  </div>
</body>
```

CSS

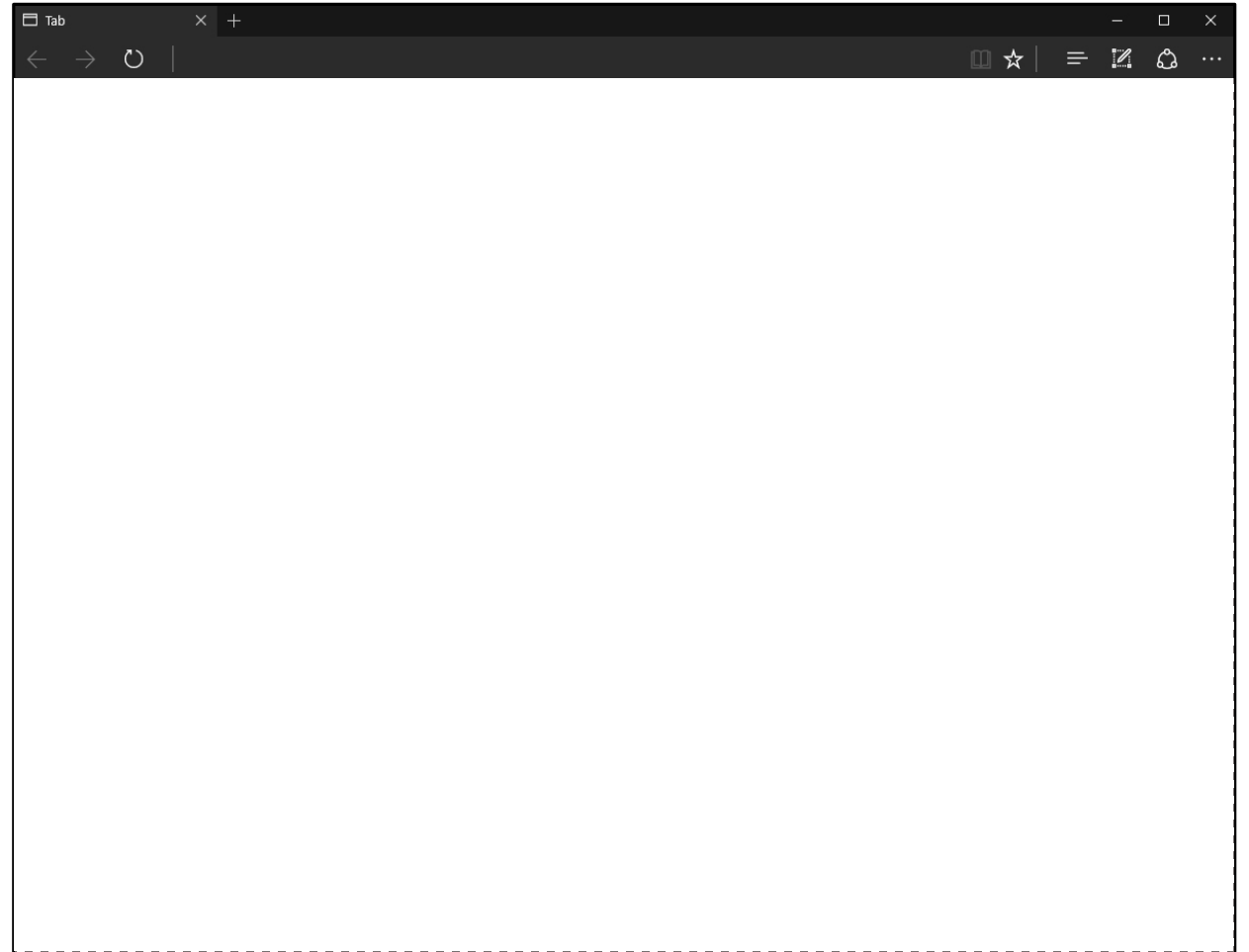
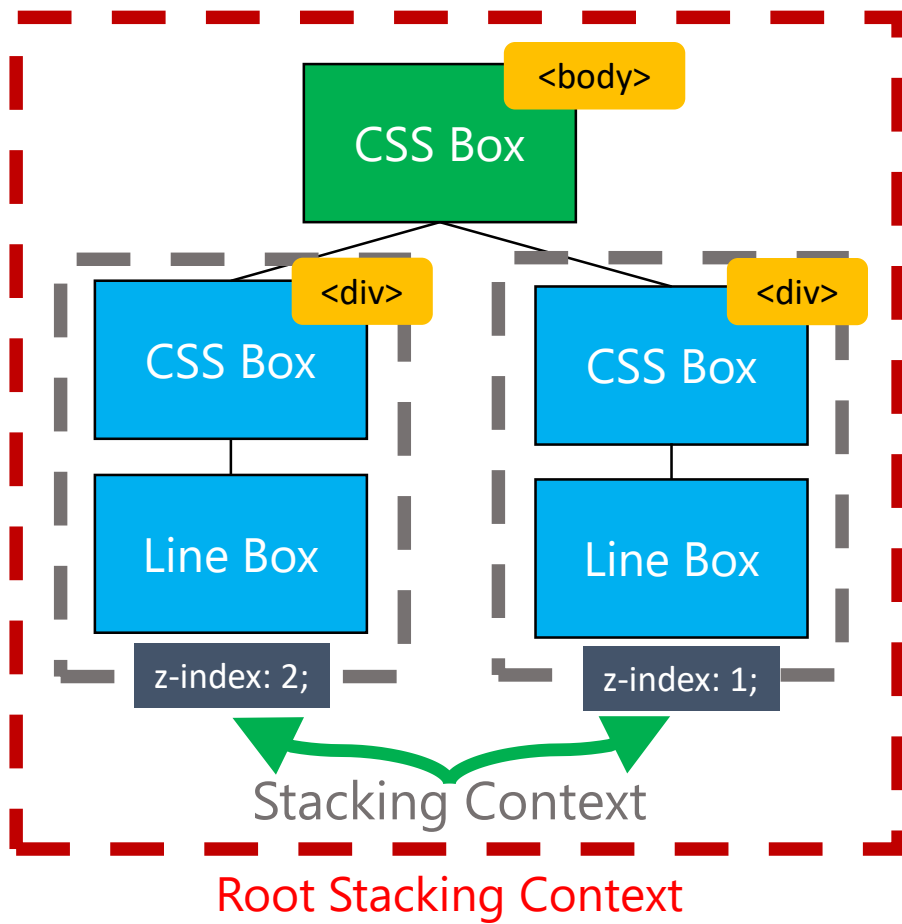
```
div {
  width: 300px;
  height: 300px;
  position: absolute;
  background: blue;
  z-index: 2;
}

#two {
  background: green;
  z-index: 1;
}
```

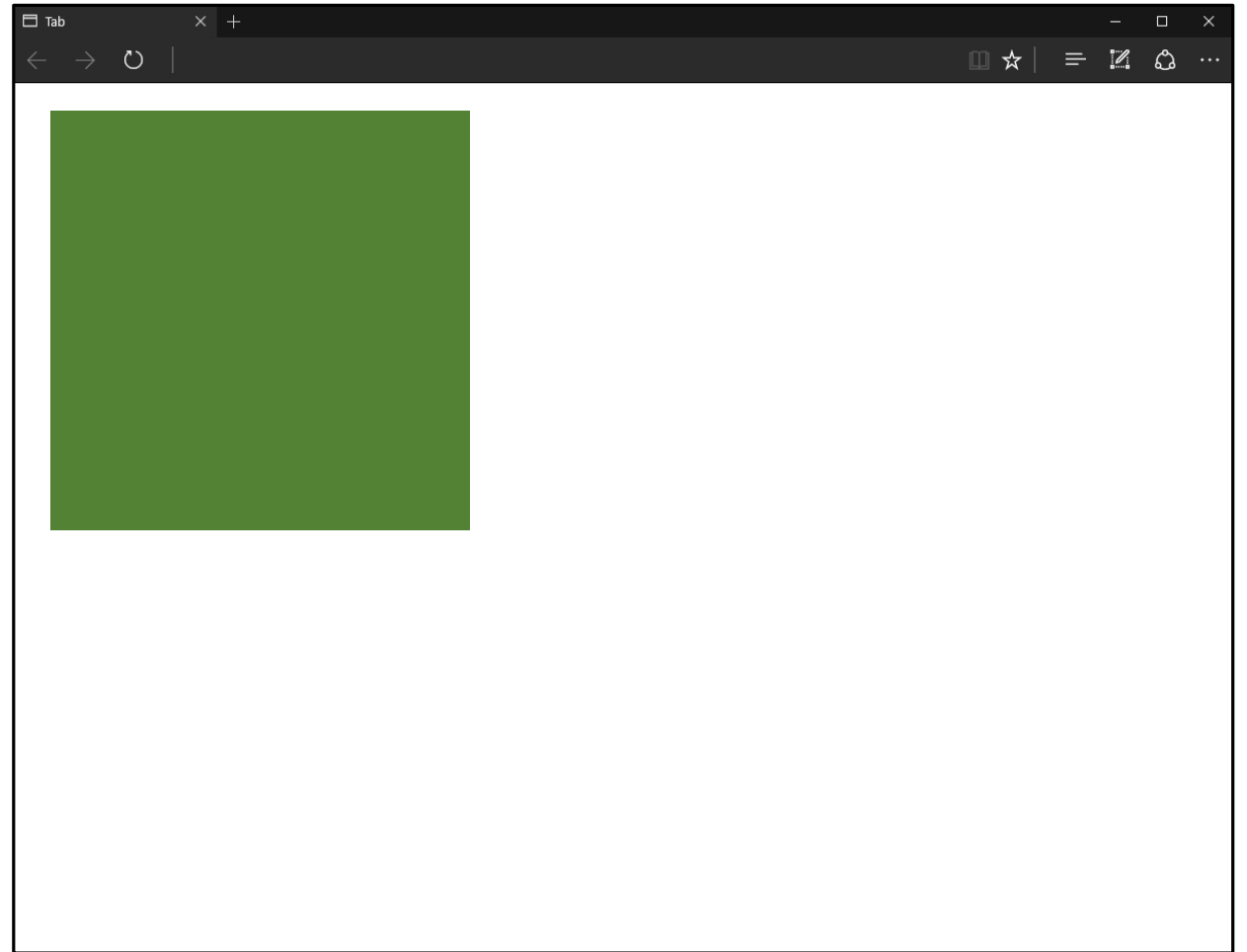
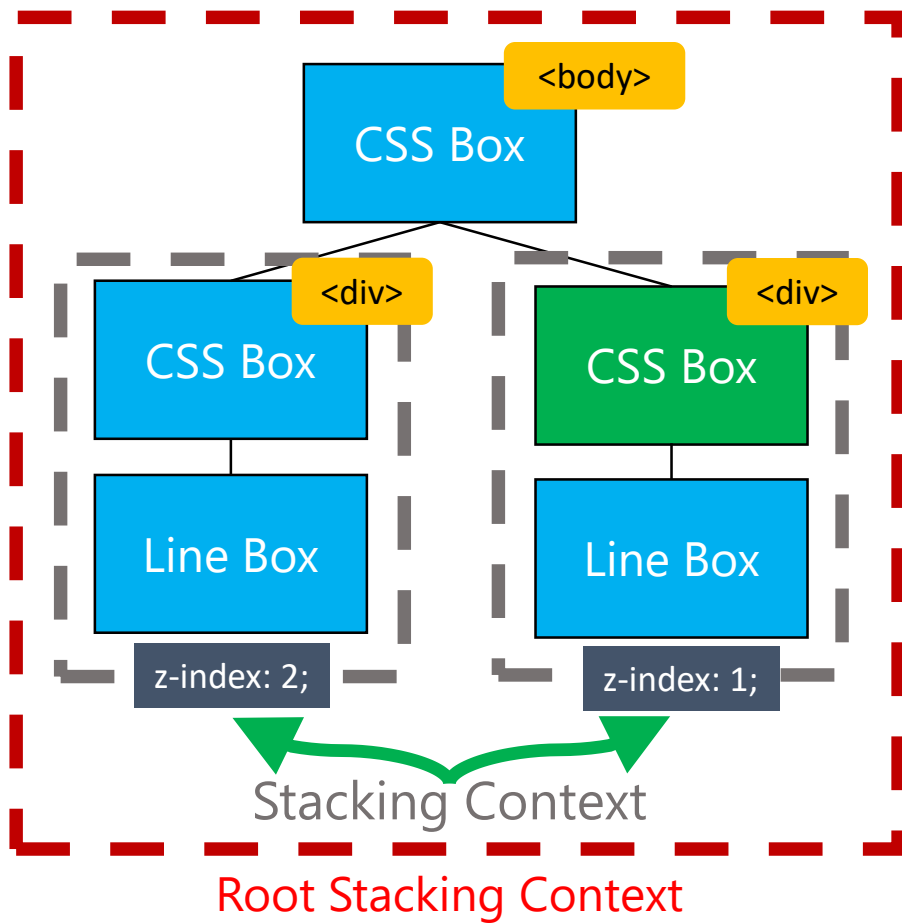
Box Tree



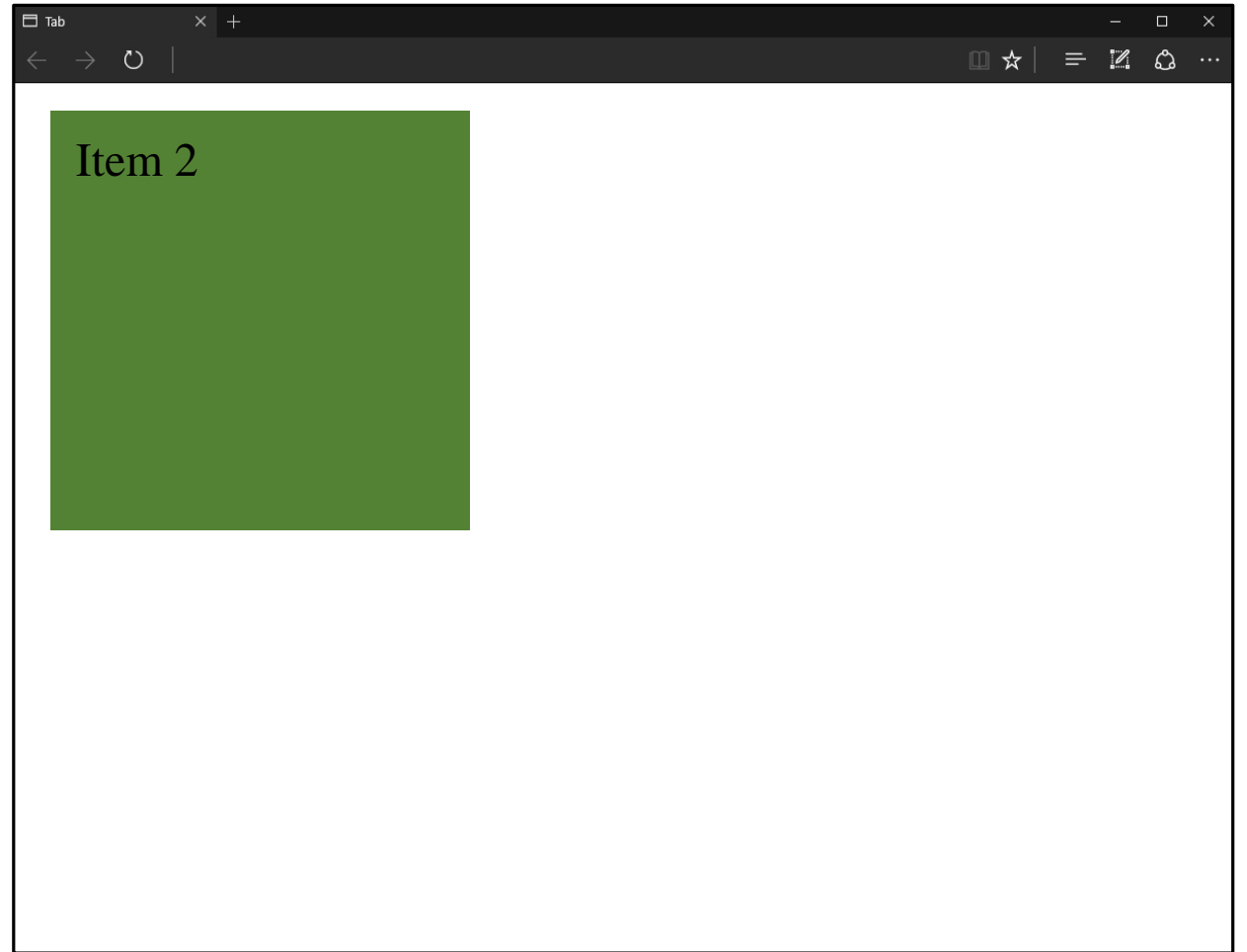
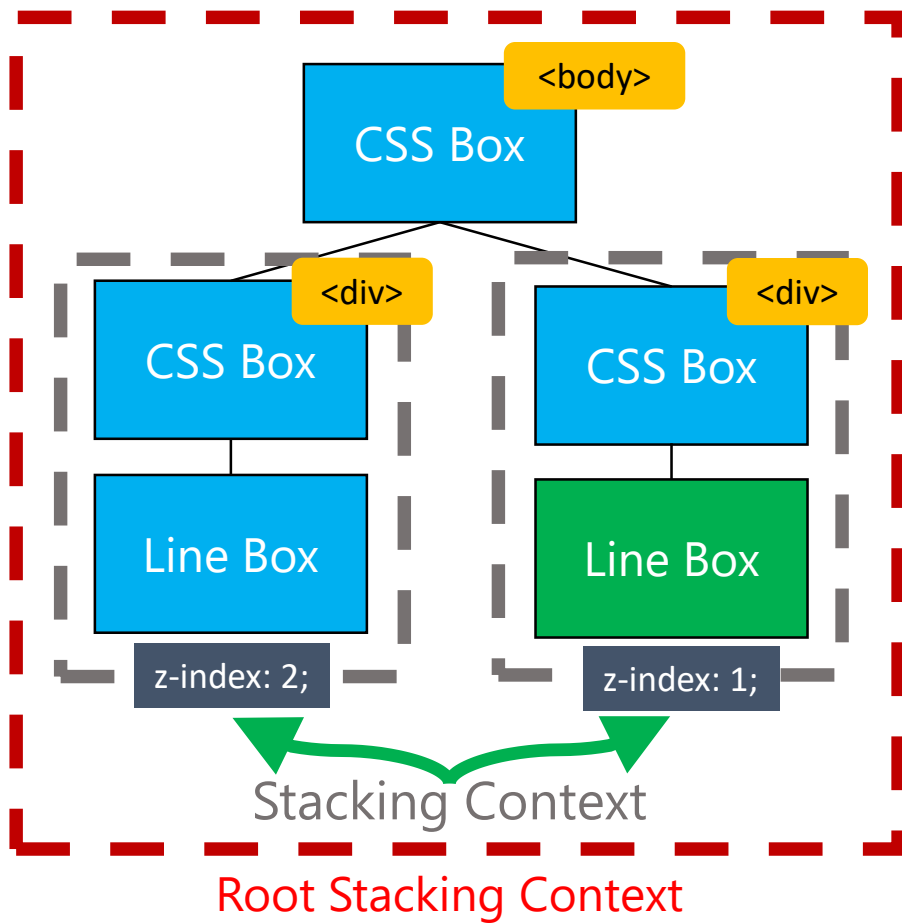
Box Tree



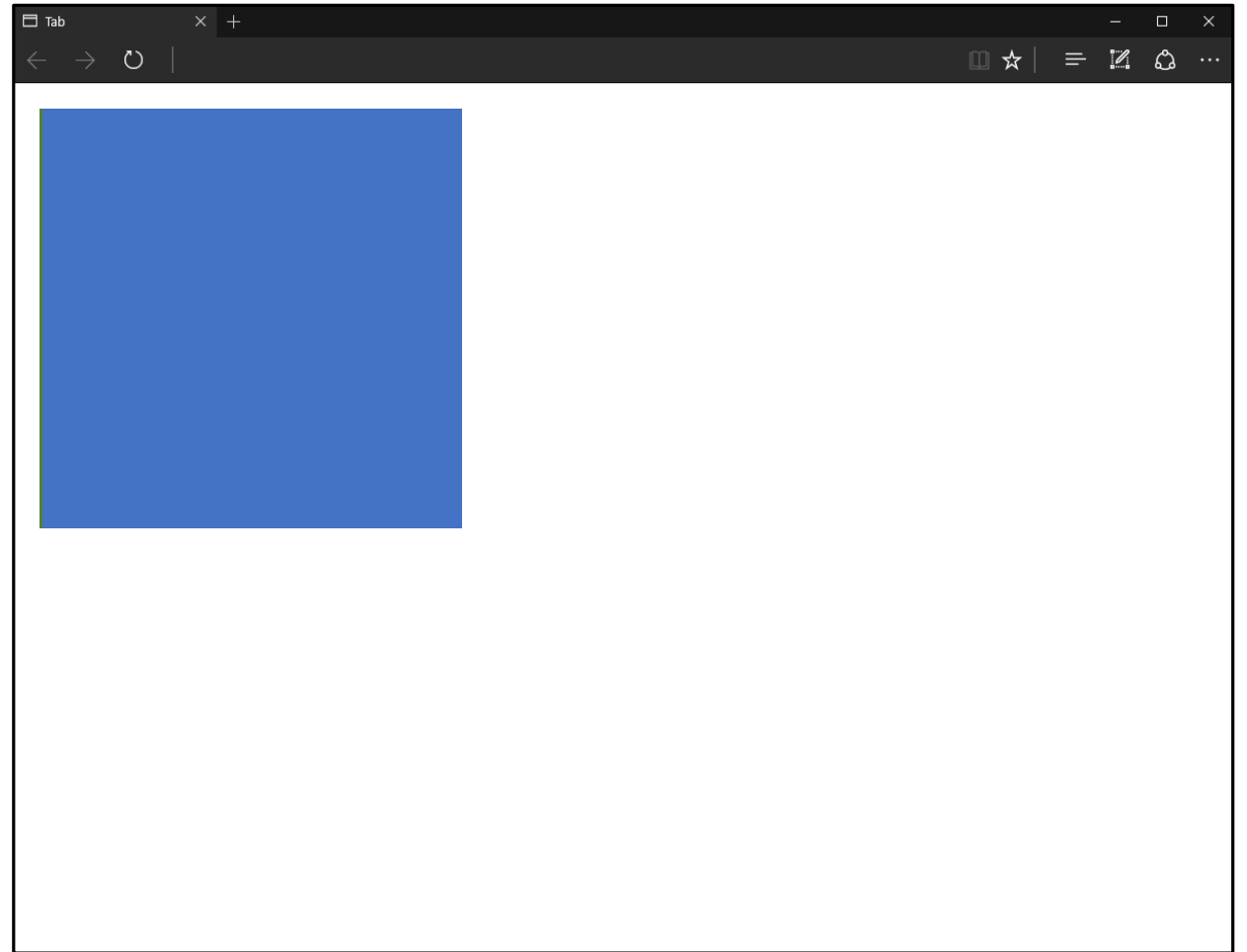
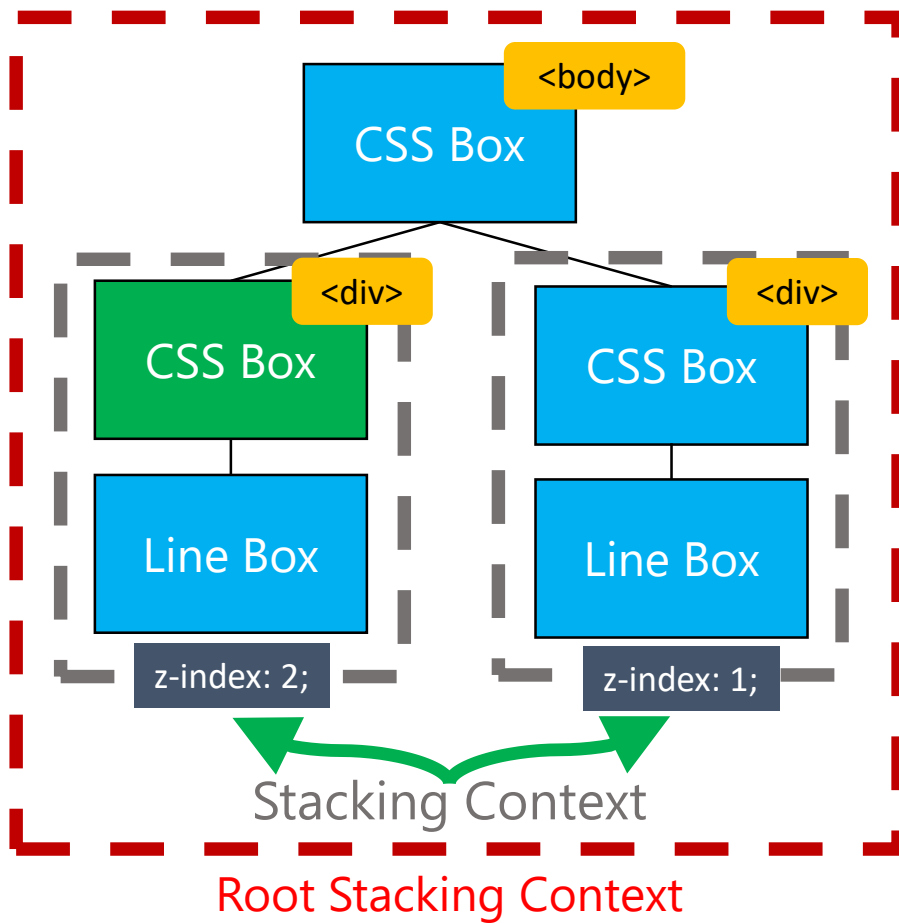
Box Tree



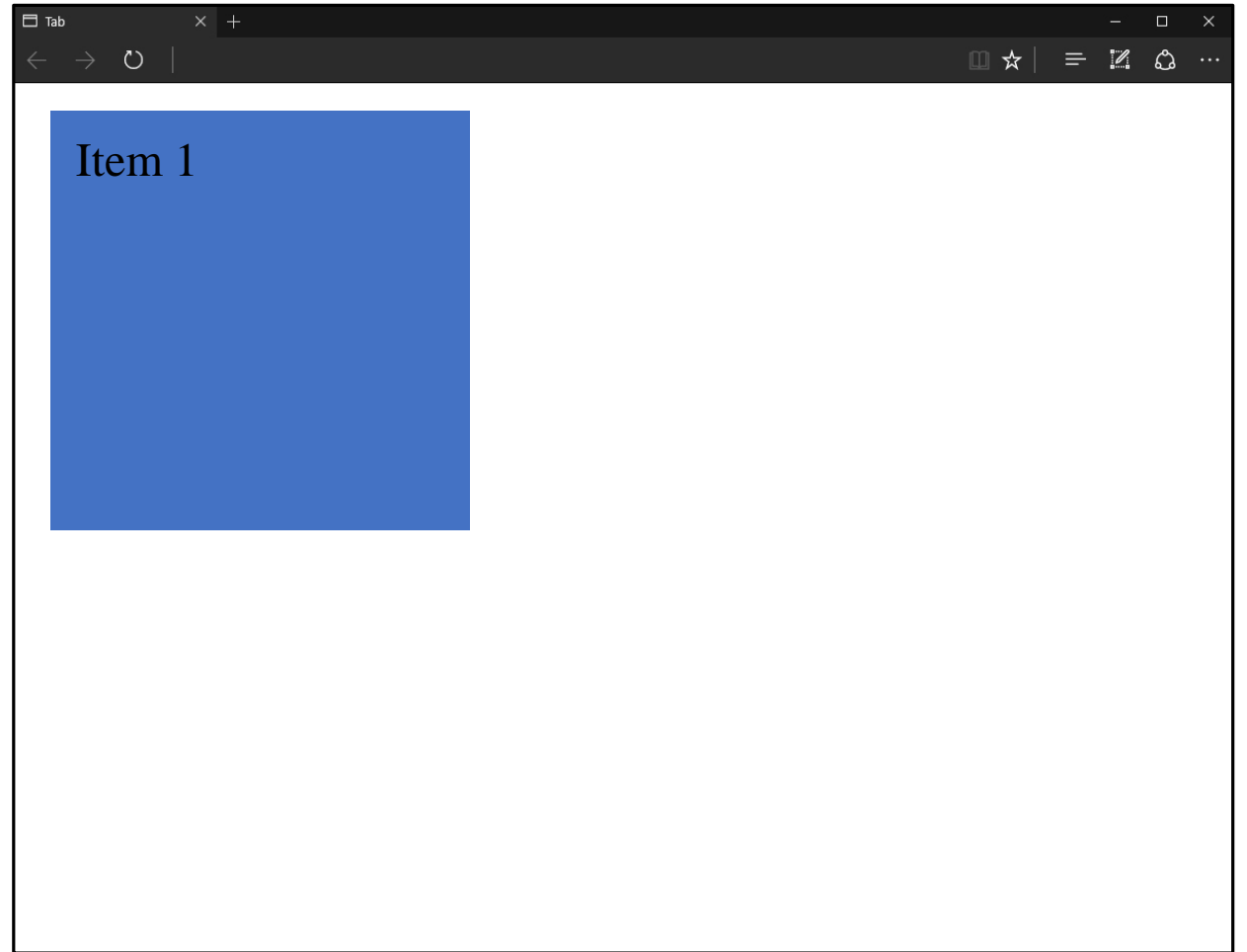
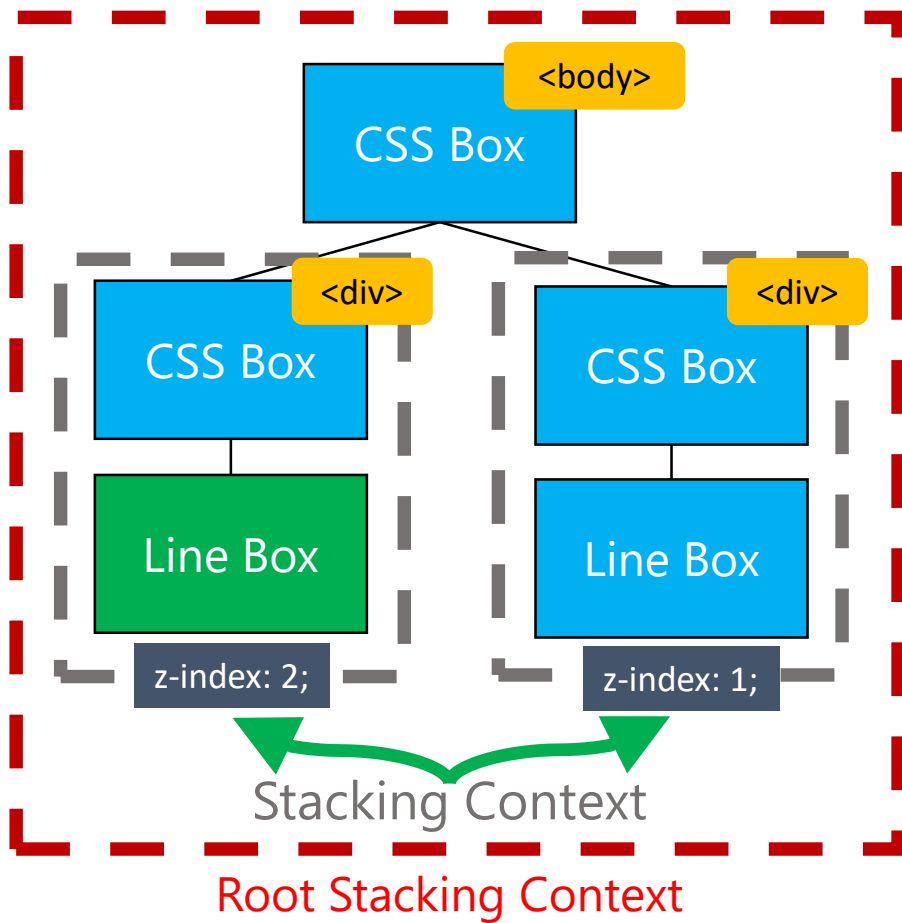
Box Tree



Box Tree



Box Tree



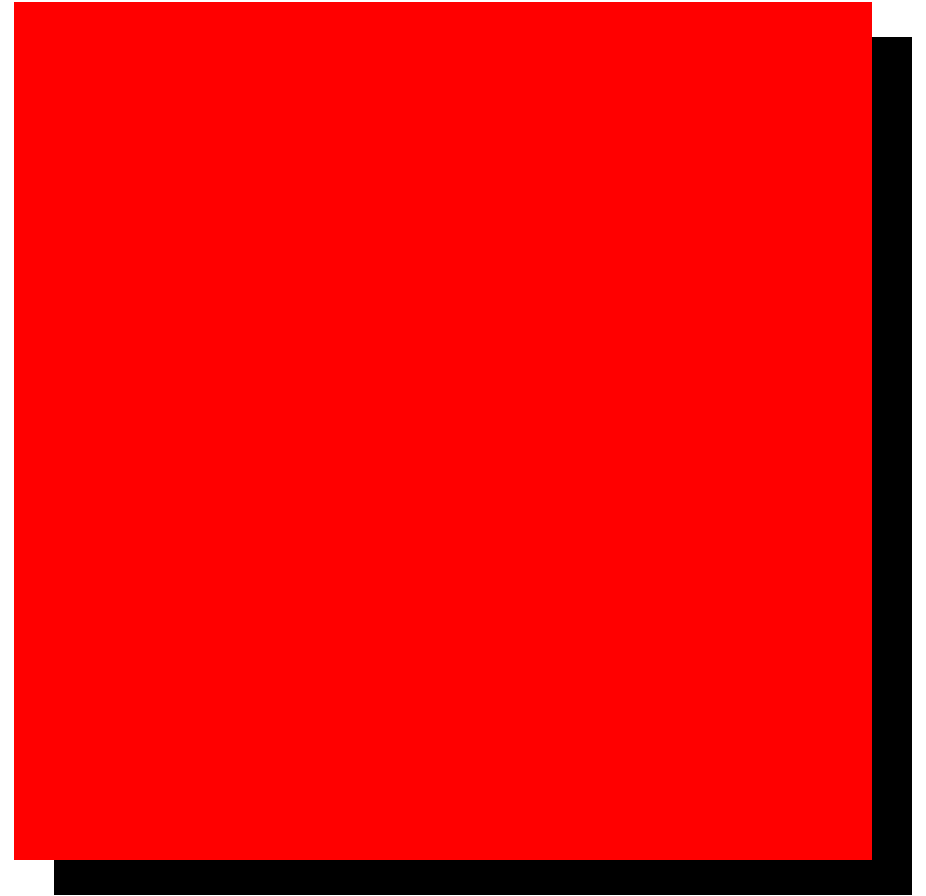
{paint:bounds}

HTML

```
<div></div>
```

CSS

```
div {  
  width: 400px;  
  height: 400px;  
  background: red;  
  box-shadow: black 20px 20px;  
}
```

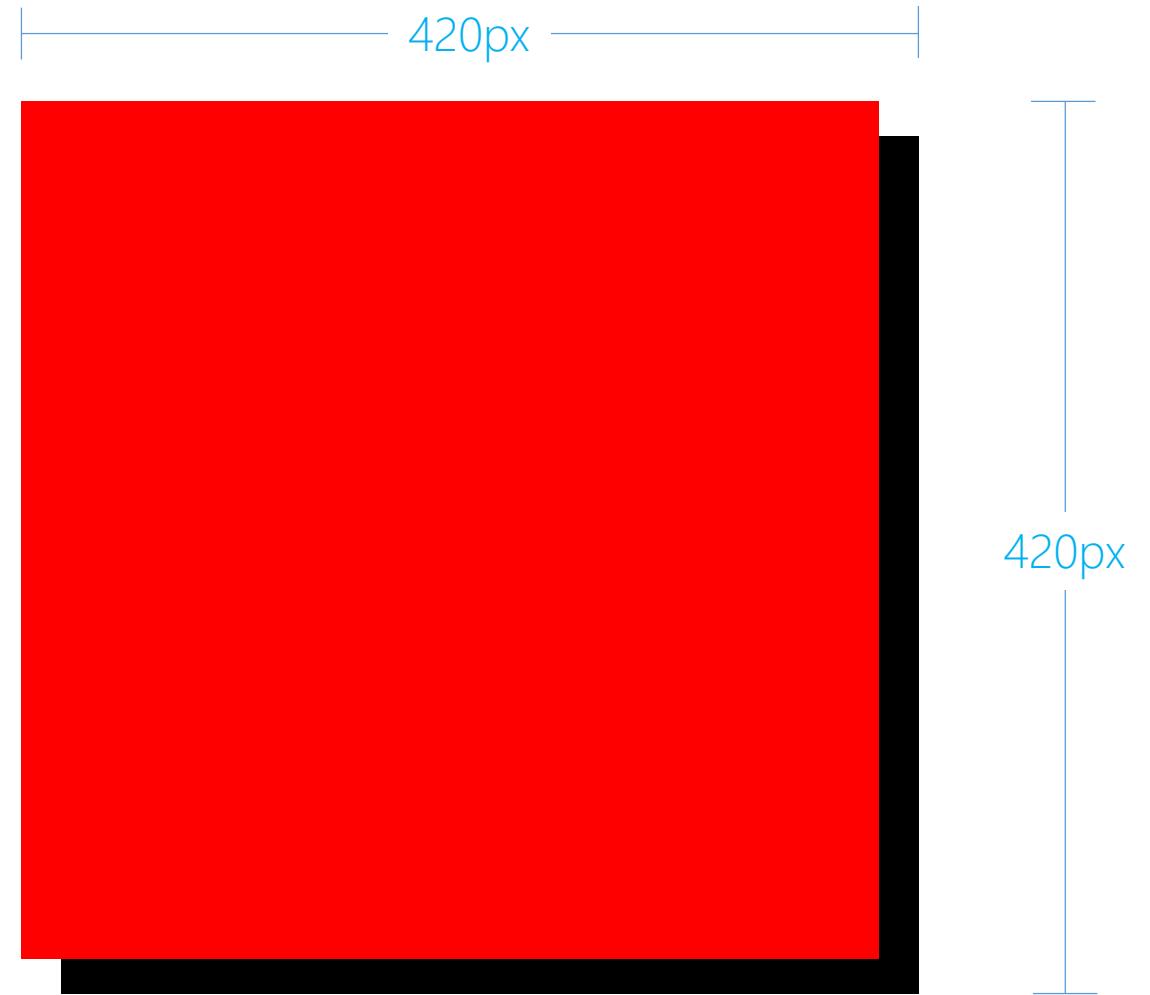


Box Tree

CSS Box	width: 400px height: 400px box-shadow: black 20px 20px
---------	--

Display Node

Disp Node	width: 420px height: 420px
-----------	-------------------------------



{compose}

Network

Content
Parsing

DOM

Style

Layout

Paint

Compose

YOU
ARE
HERE

Tab

Word Online OneDrive Documents Document 2 - Saved to OneDrive Share Greg Whitworth Sign out

FILE HOME INSERT PAGE LAYOUT REVIEW VIEW Tell me what you want to do EDIT IN WORD

Large Heading

[Redacted text block]

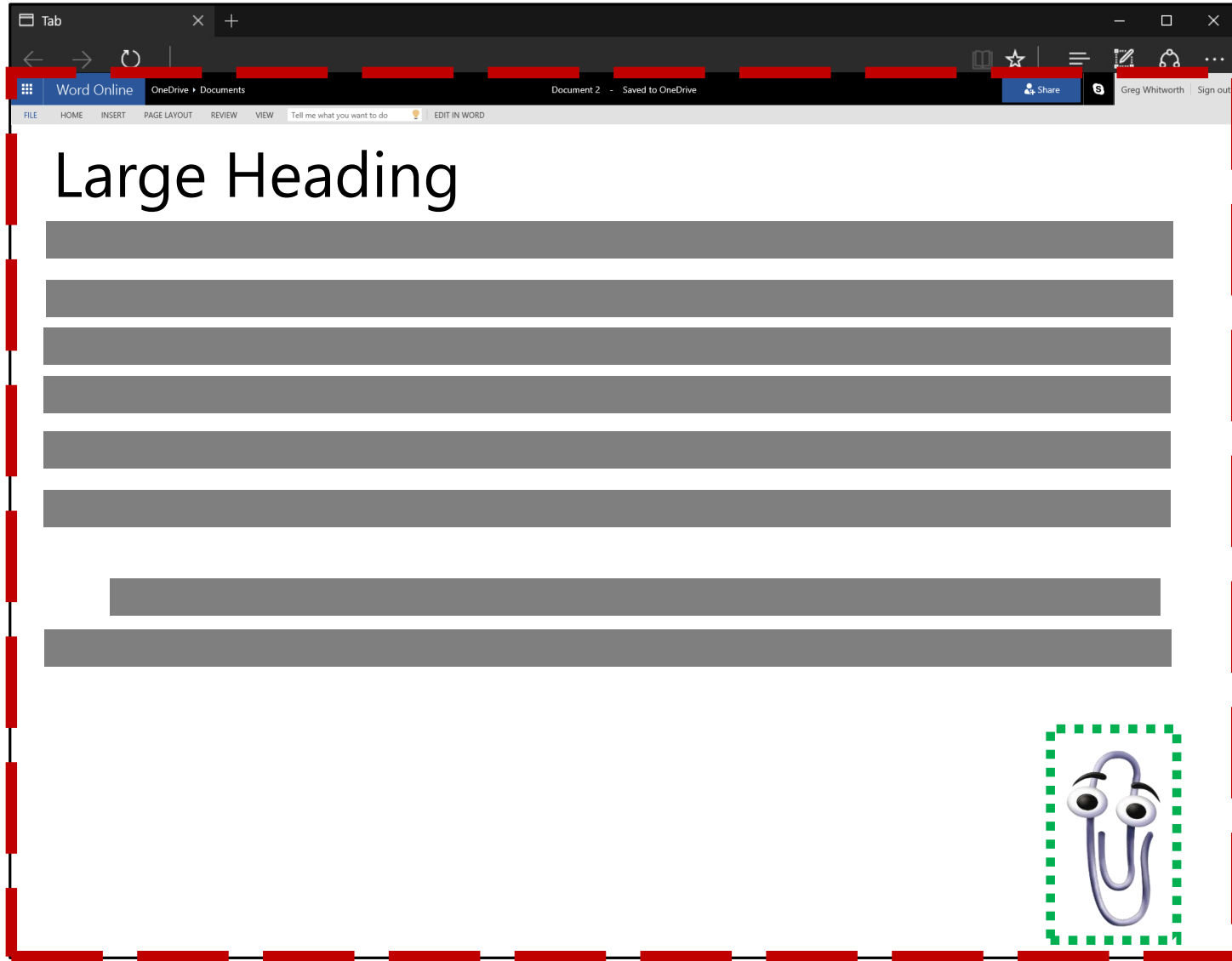


HTML

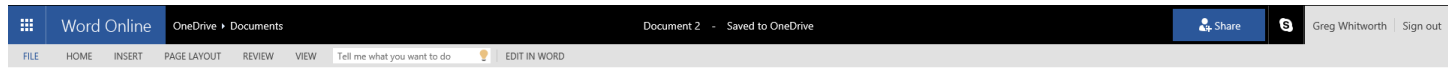
```
<div class="clippy">  
</div>
```

CSS

```
.clippy {  
  width: 100px;  
  height: 100px;  
  animation: pulse 1s infinite;  
  background: green;  
}  
  
@keyframes pulse {  
  from {  
    transform: scale(1, 1);  
  }  
  to {  
    transform: scale(2, 2);  
  }  
}
```



Root Composite Layer



Large Heading

[Redacted content]





Large Heading

[Redacted content consisting of multiple horizontal grey bars]



{invalidation}

Input



JS



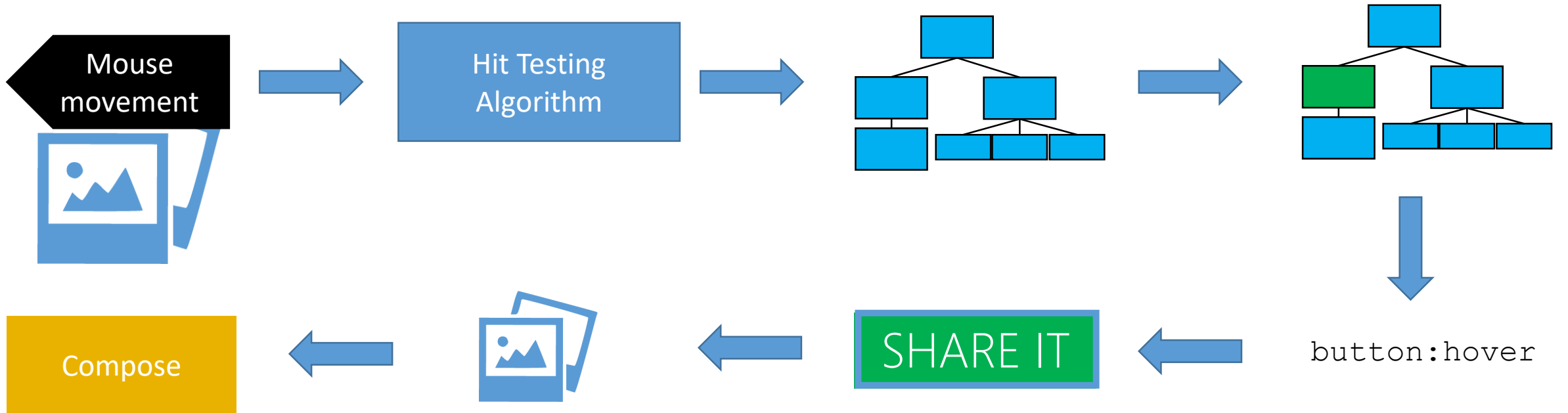
SHARE IT

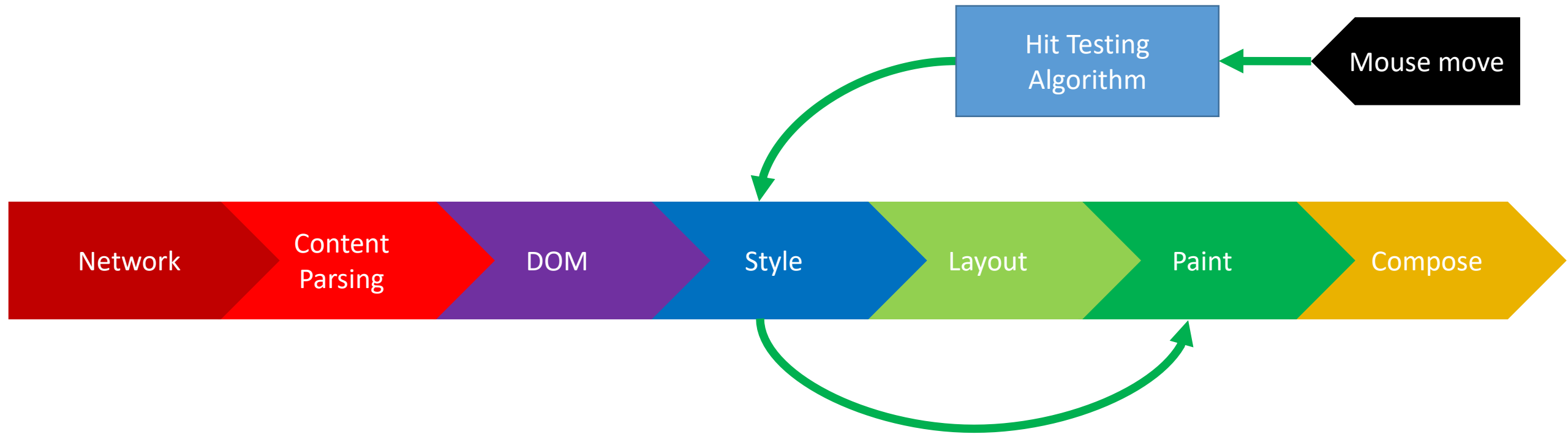
CSS

```
button {  
  float: left;  
  background: orange;  
  padding: 3px 10px;  
  border: 2px solid blue;  
}
```

```
button:hover {  
  background: green;  
  color: white;  
}
```

Box Tree





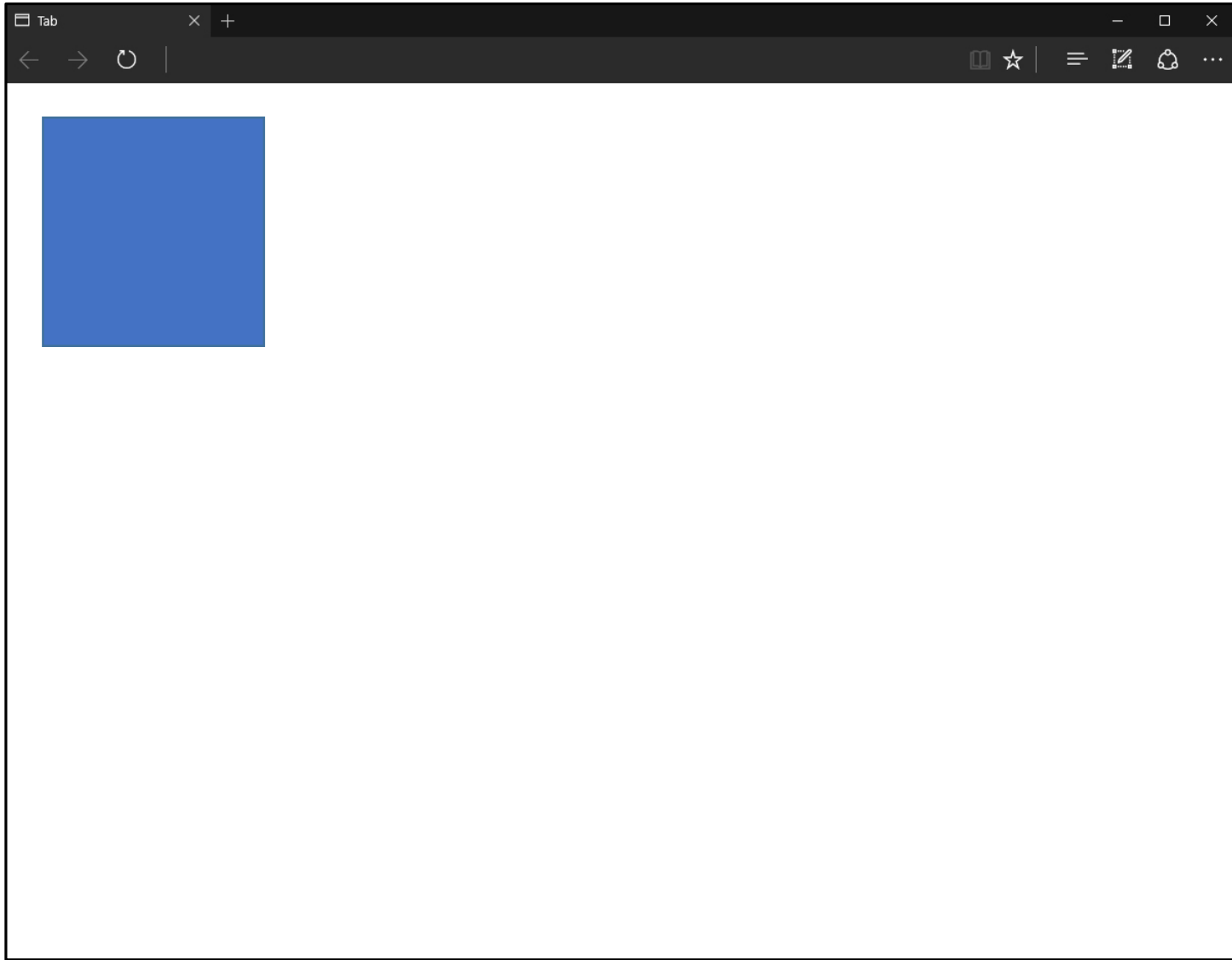
Modifying the box model on :hover

3971	3972	3973	3974	3975	3976	3977	3978	3979	3980
3981	3982	3983	3984	3985	3986	3987	3988	3989	3990
3991	3992	3993	3994	3995	3996	3997	3998	3999	4000
4001	4002	4003	4004	4005	4006	4007	4008	4009	4010
4011	4012	4013	4014	4015	4016	4017	4018	4019	4020
4021	4022	4023	4024	4025	4026	4027	4028	4029	4030
4031	4032	4033	4034	4035	4036	4037	4038	4039	4040
4041	4042	4043	4044	4045	4046	4047	4048	4049	4050
4051	4052	4053	4054	4055	4056	4057	4058	4059	4060
4061	4062	4063	4064	4065	4066	4067	4068	4069	4070
4071	4072	4073	4074	4075	4076	4077	4078	4079	4080
4081	4082	4083	4084	4085	4086	4087	4088	4089	4090
4091	4092	4093	4094	4095	4096	4097	4098	4099	4100
4101	4102	4103	4104	4105	4106	4107	4108	4109	4110
4111	4112	4113	4114	4115	4116	4117	4118	4119	4120
4121	4122	4123	4124	4125	4126	4127	4128	4129	4130
4131	4132	4133	4134	4135	4136	4137	4138	4139	4140
4141	4142	4143	4144	4145	4146	4147	4148	4149	4150

Not modifying the box model on :hover

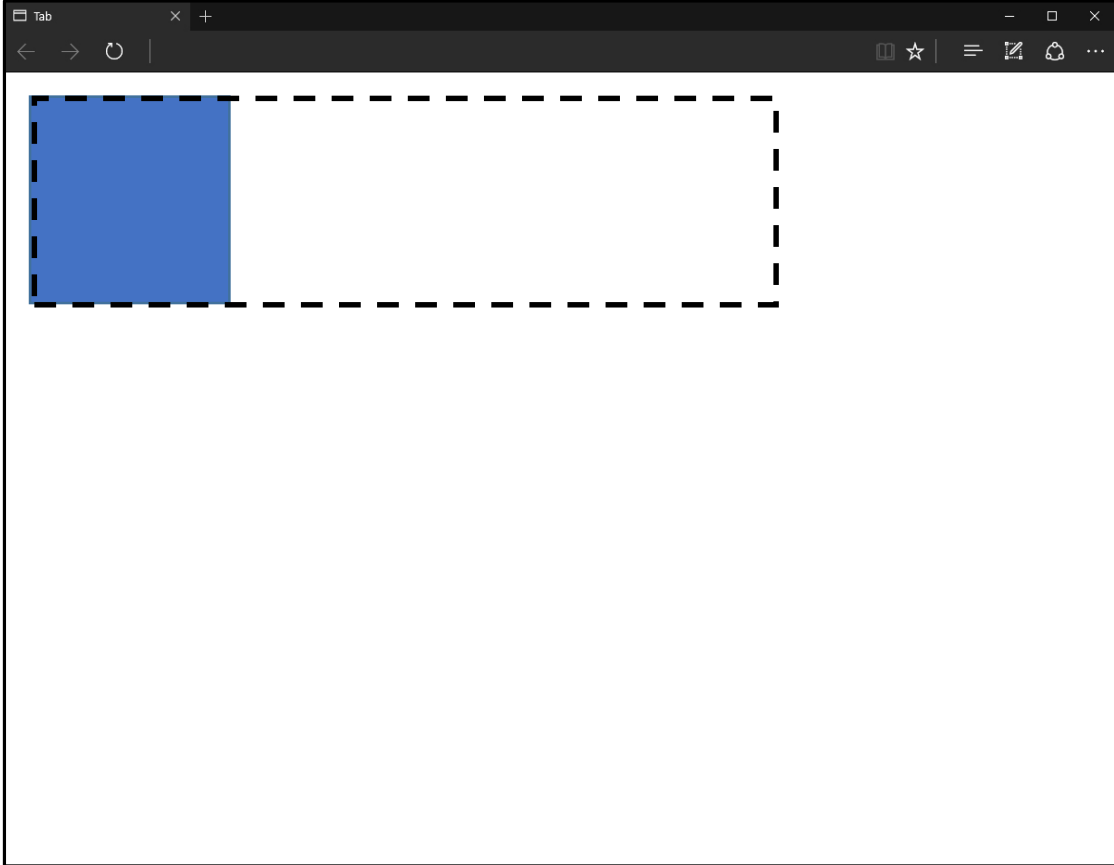
1591	1592	1593	1594	1595	1596	1597	1598	1599	1600
1601	1602	1603	1604	1605	1606	1607	1608	1609	1610
1611	1612	1613	1614	1615	1616	1617	1618	1619	1620
1621	1622	1623	1624	1625	1626	1627	1628	1629	1630
1631	1632	1633	1634	1635	1636	1637	1638	1639	1640
1641	1642	1643	1644	1645	1646	1647	1648	1649	1650
1651	1652	1653	1654	1655	1656	1657	1658	1659	1660
1661	1662	1663	1664	1665	1666	1667	1668	1669	1670
1671	1672	1673	1674	1675	1676	1677	1678	1679	1680
1681	1682	1683	1684	1685	1686	1687	1688	1689	1690

↳



JS

```
var moved = 0;  
var tooFar = 500;  
  
function move(el) {  
    if(moved <= tooFar)  
        moved++;  
    el.style.marginLeft = moved + "px";  
}
```

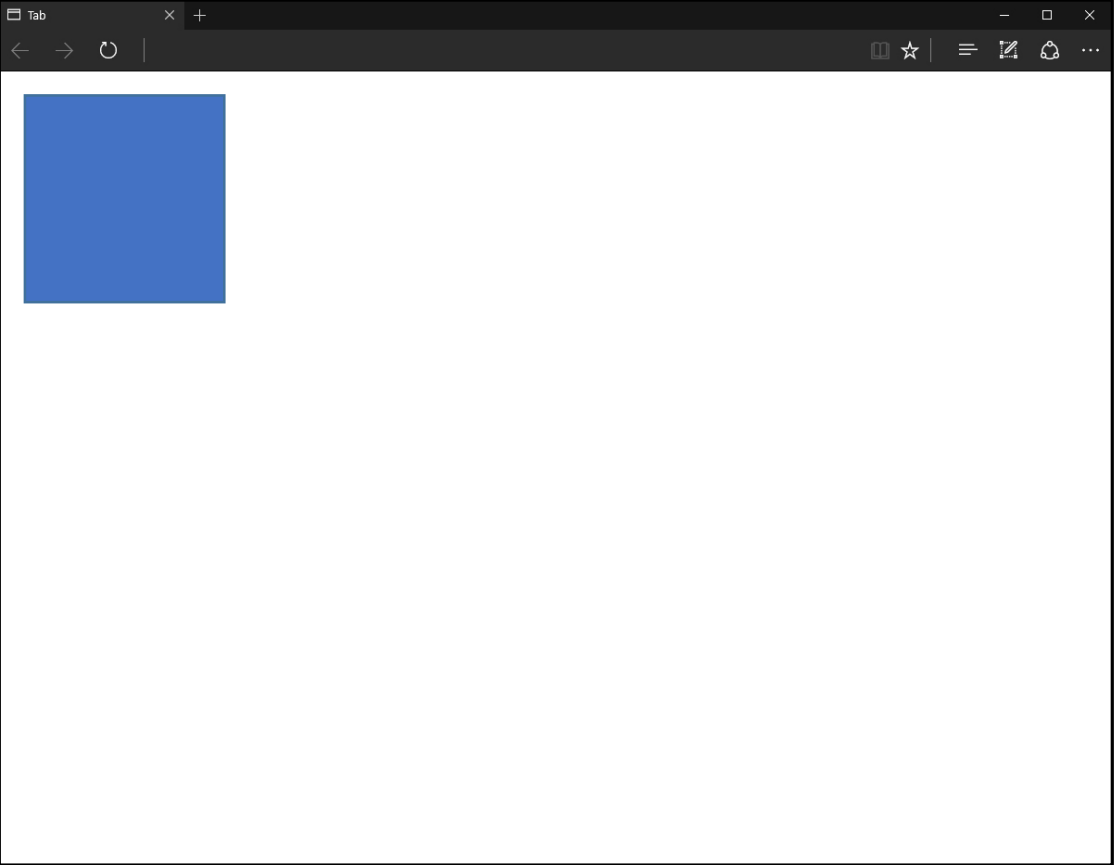


JS



CSS

```
#move {  
    width: 300px;  
    height: 300px;  
    background: blue;  
    animation: move 1s;  
}  
  
@keyframes {  
    to {  
        transform: translateX(500px);  
    }  
}
```



{take aways}

- Audit your CSS at least twice a year
- Remove unused CSS or redundant styles
- !important should be a sign that it may be time for a CSS refactor

- Try and keep DOM structure as flat as possible
- Try and use simple selectors (eg: ID, classes, tags)
- Limit Layout thrashing

- Make display only invalidations (layout is computationally heavy)
- Elicit feedback on the way you solved the solution



@gregwhitworth