CSS Positioning

CSS positioning determines the way elements get drawn and positioned within a webpage

Positioning is determined by the value of a property common to all HTML elements: *position*

It can takes 4 values:

CSS positioning determines the way elements get drawn and positioned within a webpage

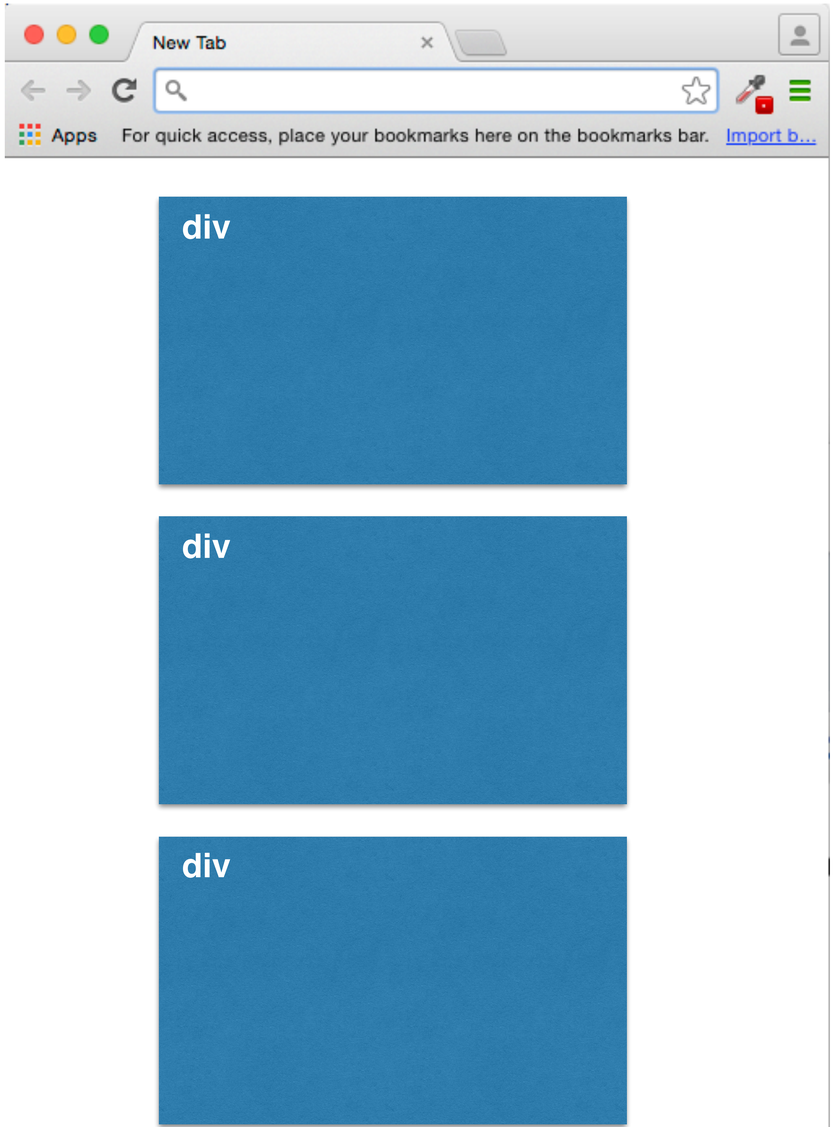
Positioning is determined by the value of a property common to all HTML elements: *position*

It can takes 4 values:

- static (the default)
- relative
- absolute
- fixed

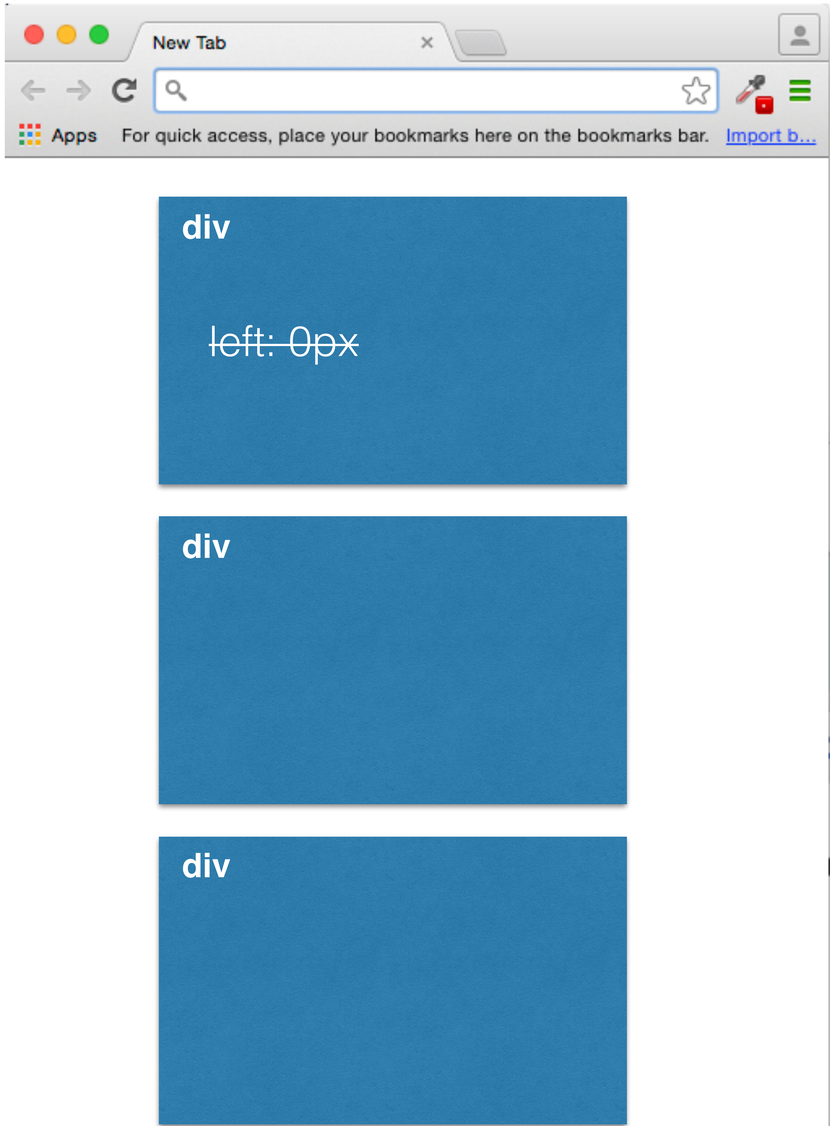
position: static

Static elements are not positioned in any particular way. they are laid down where they occur within the flow of the page.
They are unaffected by the css properties
top,left,bottom,right



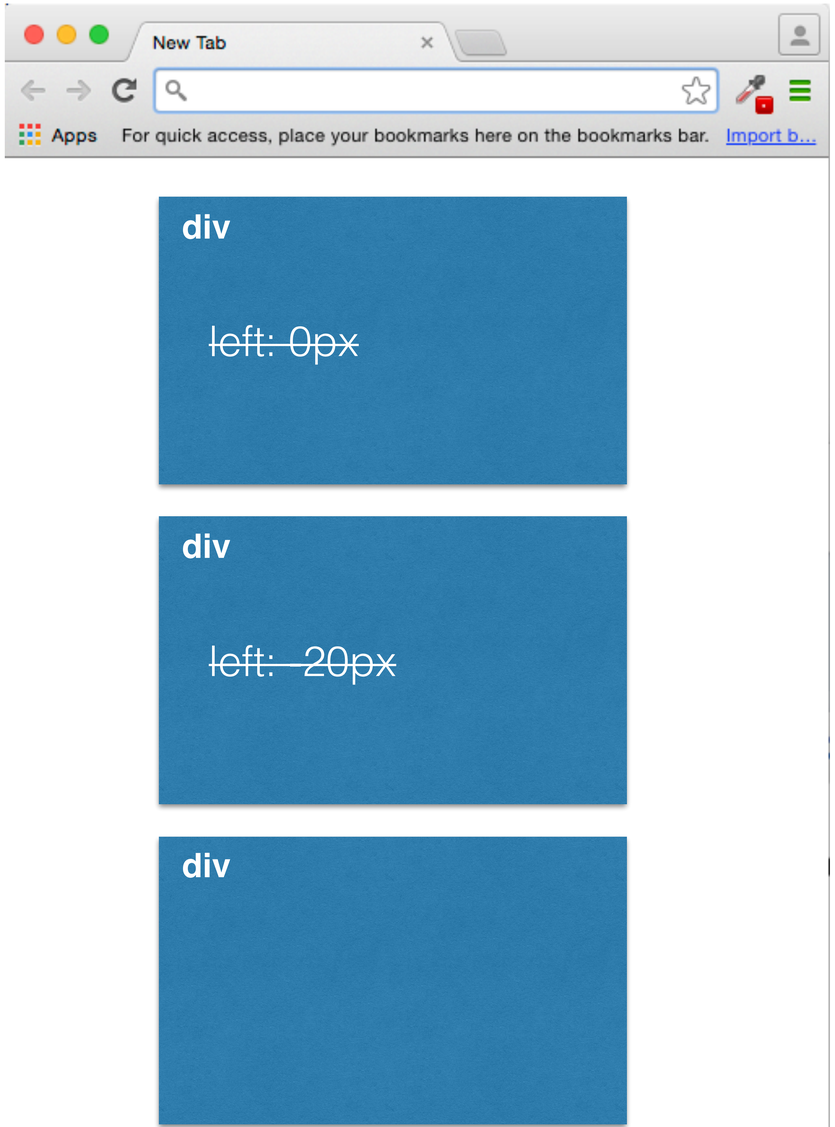
position: static

Static elements are not positioned in any particular way. they are laid down where they occur within the flow of the page.
They are unaffected by the css properties
top,left,bottom,right



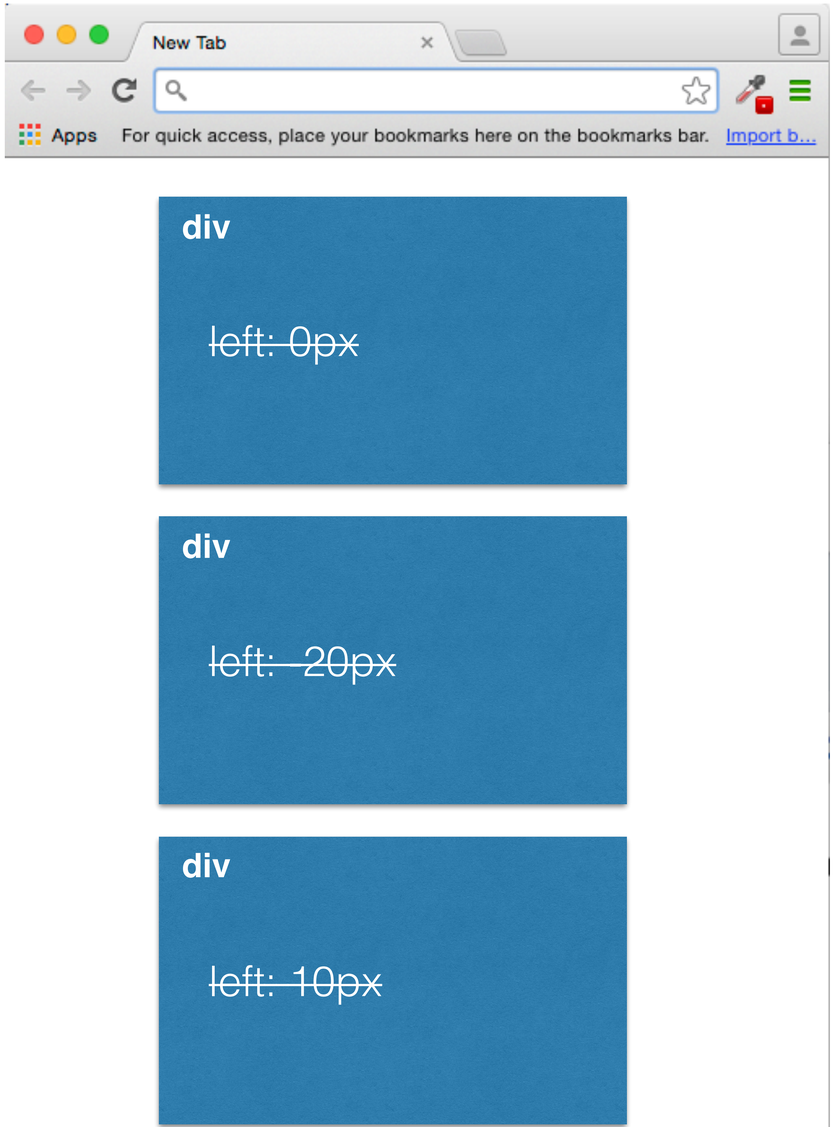
position: static

Static elements are not positioned in any particular way. they are laid down where they occur within the flow of the page.
They are unaffected by the css properties
top,left,bottom,right



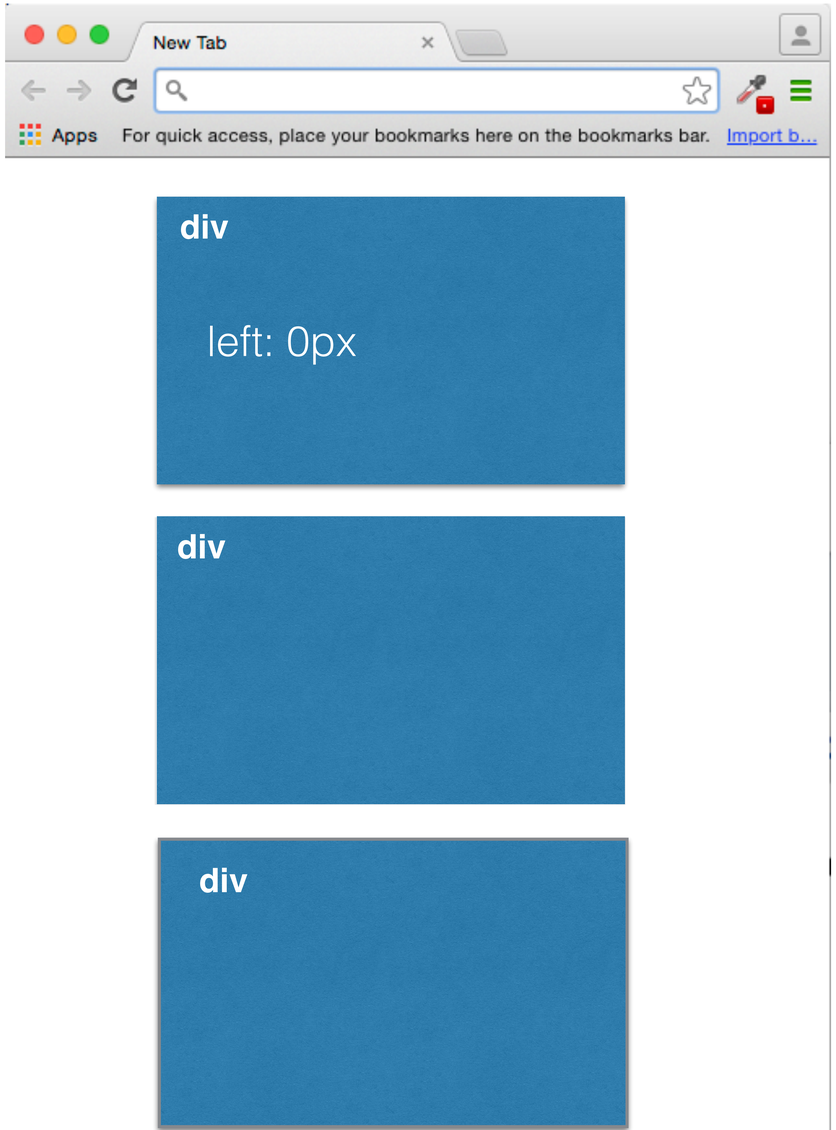
position: static

Static elements are not positioned in any particular way. they are laid down where they occur within the flow of the page.
They are unaffected by the css properties
top,left,bottom,right



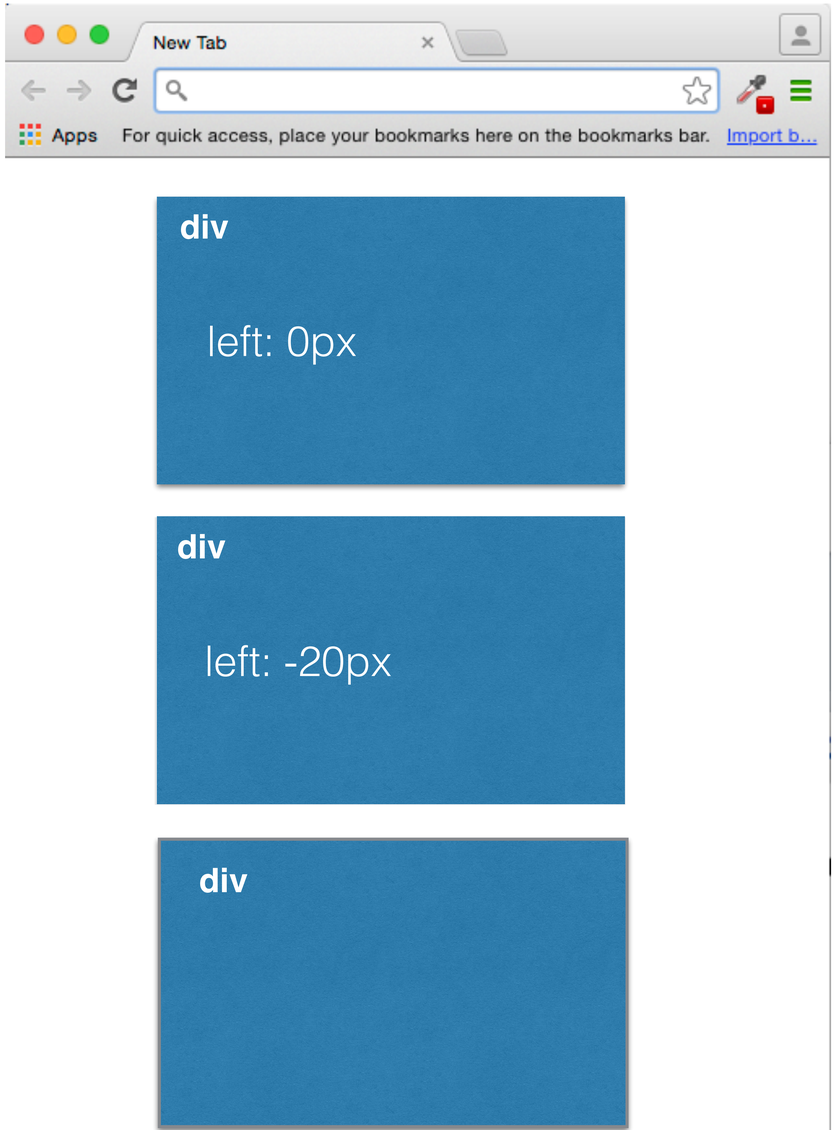
position: relative

Similar to *static,* but the elements can be positioned relatively to the space reserved for them in the flow, using the css properties *top,left,bottom,right.*
When doing so, the space allocated for them in the page won't be affected (leaving gaps when moving the element around). The element might overlap with other content around it



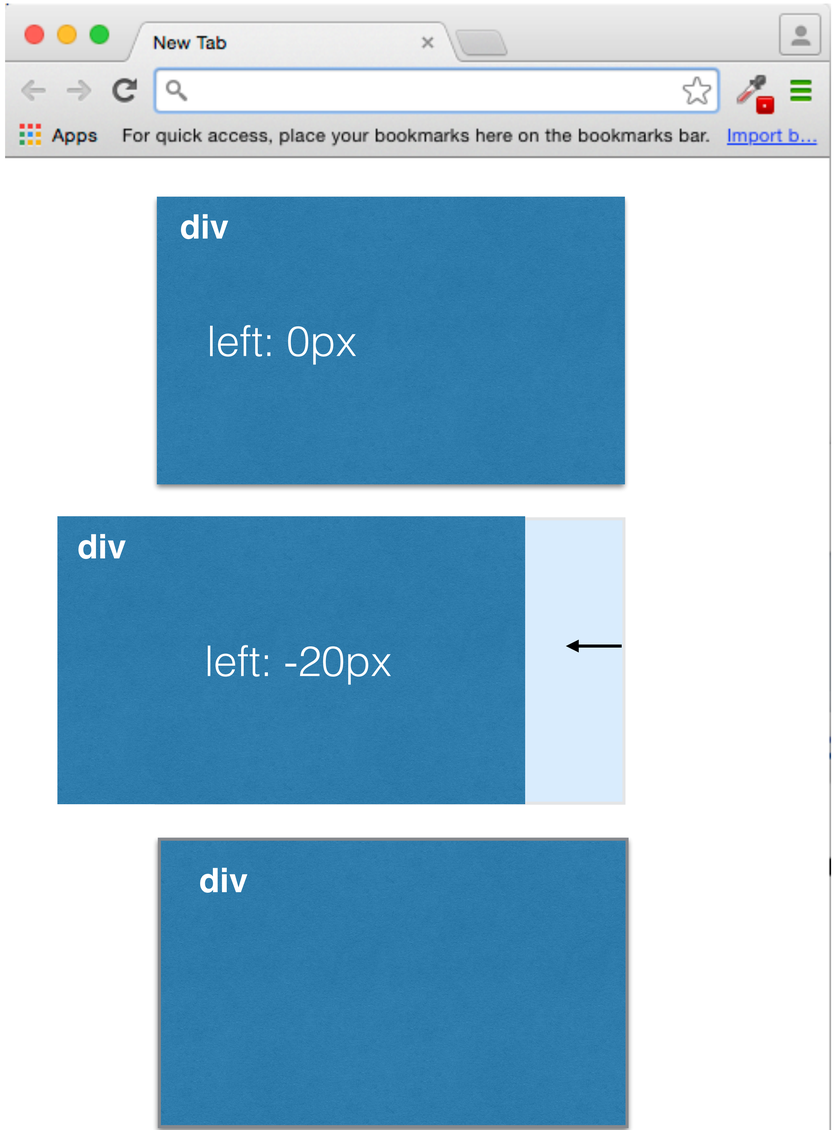
position: relative

Similar to *static,* but the elements can be positioned relatively to the space reserved for them in the flow, using the css properties *top,left,bottom,right.*
When doing so, the space allocated for them in the page won't be affected (leaving gaps when moving the element around). The element might overlap with other content around it



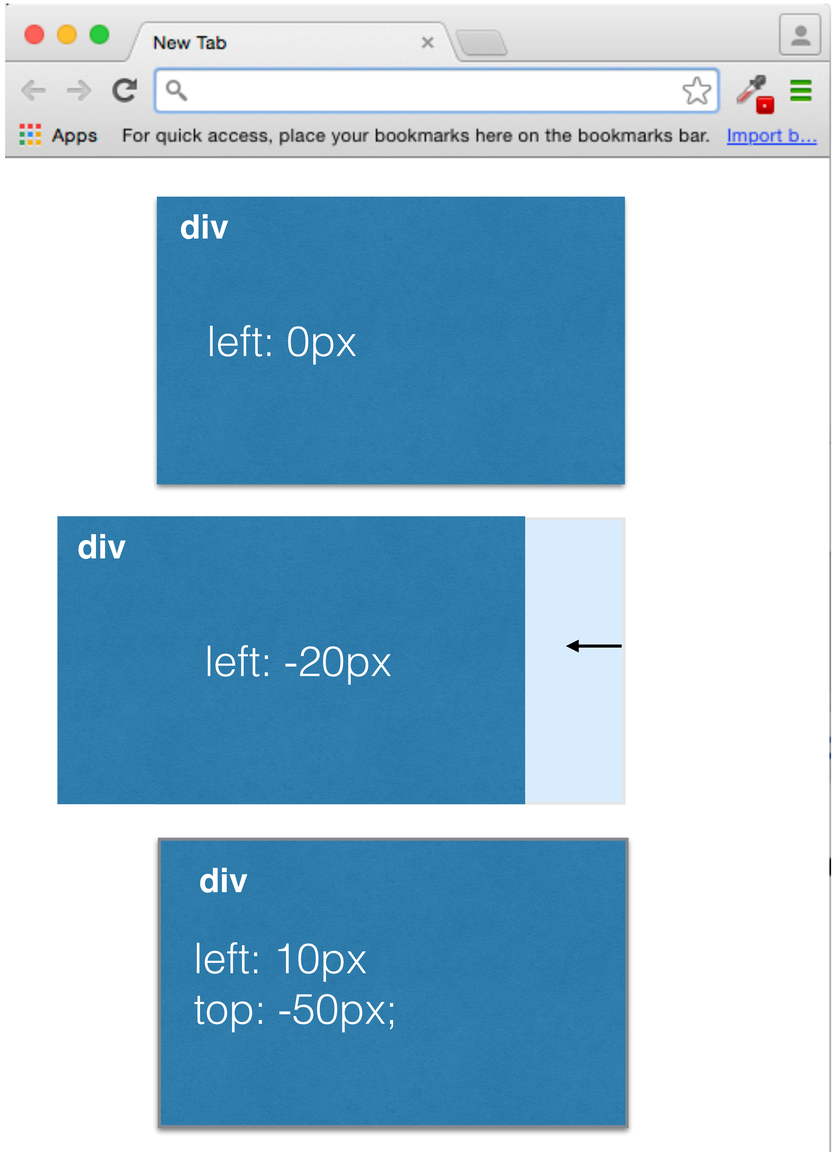
position: relative

Similar to *static,* but the elements can be positioned relatively to the space reserved for them in the flow, using the css properties *top,left,bottom,right.*
When doing so, the space allocated for them in the page won't be affected (leaving gaps when moving the element around). The element might overlap with other content around it



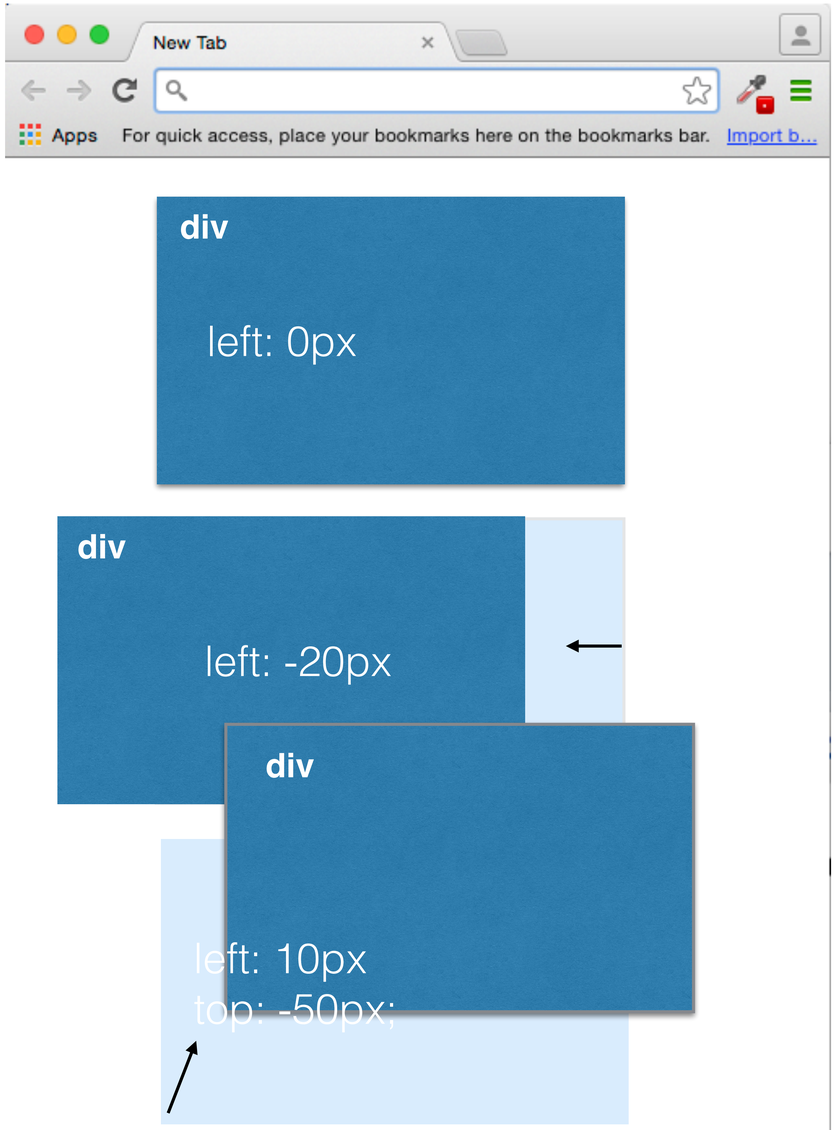
position: relative

Similar to *static,* but the elements can be positioned relatively to the space reserved for them in the flow, using the css properties *top,left,bottom,right.*
When doing so, the space allocated for them in the page won't be affected (leaving gaps when moving the element around). The element might overlap with other content around it



position: relative

Similar to *static,* but the elements can be positioned relatively to the space reserved for them in the flow, using the css properties *top,left,bottom,right.*
When doing so, the space allocated for them in the page won't be affected (leaving gaps when moving the element around). The element might overlap with other content around it

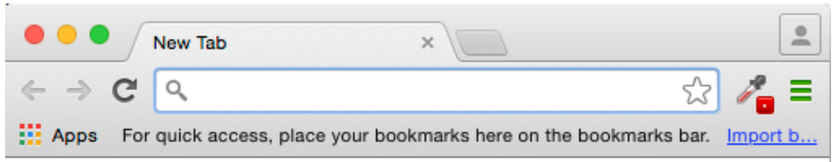


position: absolute (no positioned ancestor)

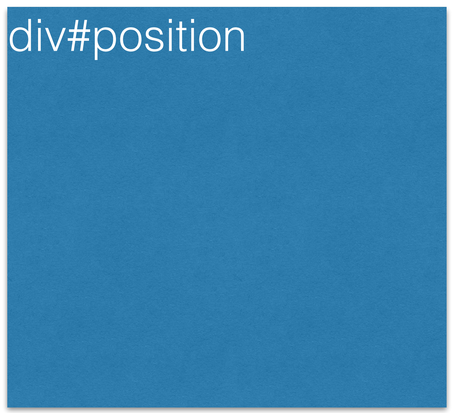
The elements is taken OUT of the flow (no space allocated for it on its parent).
the element is initially positioned where it would've been inserted if statically positioned.

If *left,top,right,bottom* properties are used, the element is positioned relative to its closest "positioned" (not static) ancestor. If no such ancestor exists, the element is positioned relative to the body of the page

```
div#position {
    position: static;
}
```



Lorem ipsum dolor sit amet, vix animal voluptatibus ad, ea quo illum delectus tincidunt, sed meis nostrum ne. Mel an latine aperiam, ea munere adolescens usu, nam id ullum choro dolor. Ius placerat qualisque definitiones in, eam altera disputationi no. Eam vero maiorum no, ad dico cotidieque sea. Nec sadipscing comprehensam no, eu reque cetero similique nec.



Praesent contentiones per in. Ei mea nobis commune pertinax, consul essent cu vix. Id vix eius veri, pro no purto veniam menandri.

Nisl abhorreant te per, ea suas deserunt indoctum est, has ullum tractatos eu. Mea ea tibique percipitur, dicta alterum vel ad. Vim ad eirmod mnesarchum, stet admodum at per, in illum eirmod argumentum qui. Novum dissentiet et nec. Pro vocibus democritum ne, eos in modus scaevola.

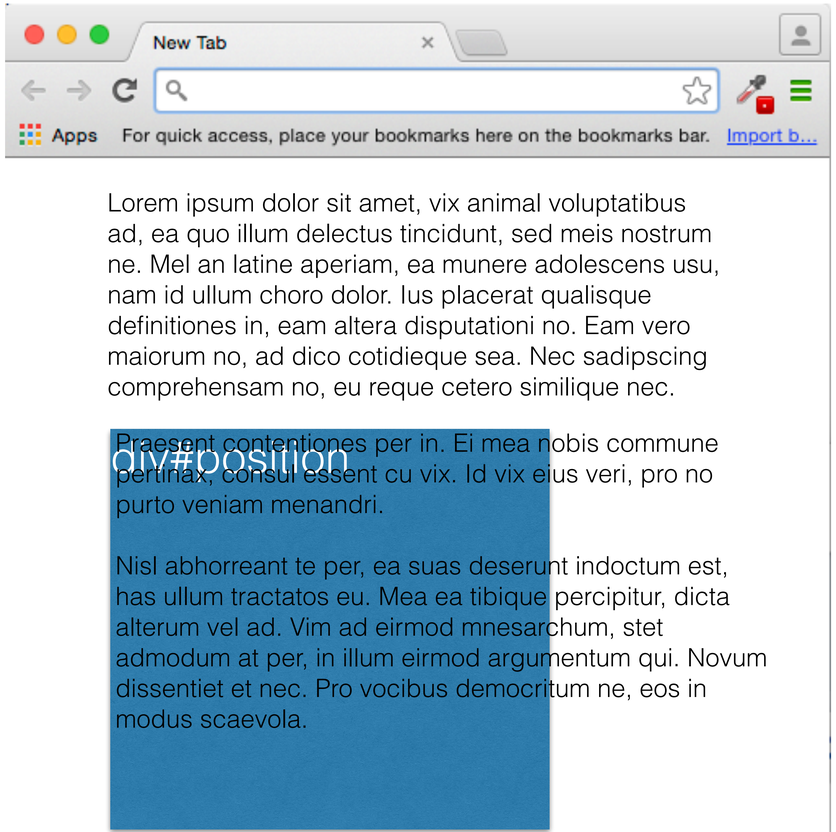
position: absolute (no positioned ancestor)

The elements is taken OUT of the flow (no space allocated for it on its parent).
the element is initially positioned where it would've been inserted if statically positioned.

If *left,top,right,bottom* properties are used, the element is positioned relative to its closest "positioned" (not static) ancestor. If no such ancestor exists, the element is positioned relative to the body of the page

```
div#position {
    position: absolute;
}
```

(the element is taken out of the flow and its space occupied from the element positioned after it in the document flow)



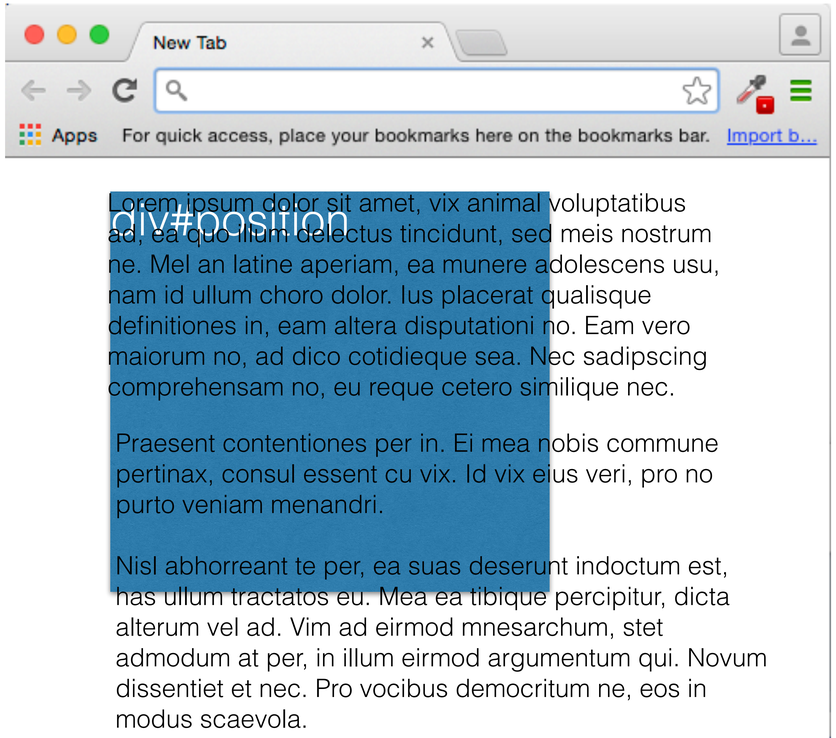
position: absolute (no positioned ancestor)

The elements is taken OUT of the flow (no space allocated for it on its parent).
the element is initially positioned where it would've been inserted if statically positioned.

If *left,top,right,bottom* properties are used, the element is positioned relative to its closest "positioned" (not static) ancestor. If no such ancestor exists, the element is positioned relative to the body of the page

```
div#position {
    position: absolute;
    top: 0;
}
```

without an ancestor with relative or absolute positioning, the element is relative to the body of the entire page (so, top:0 means it will be moved to the top of the entire page)



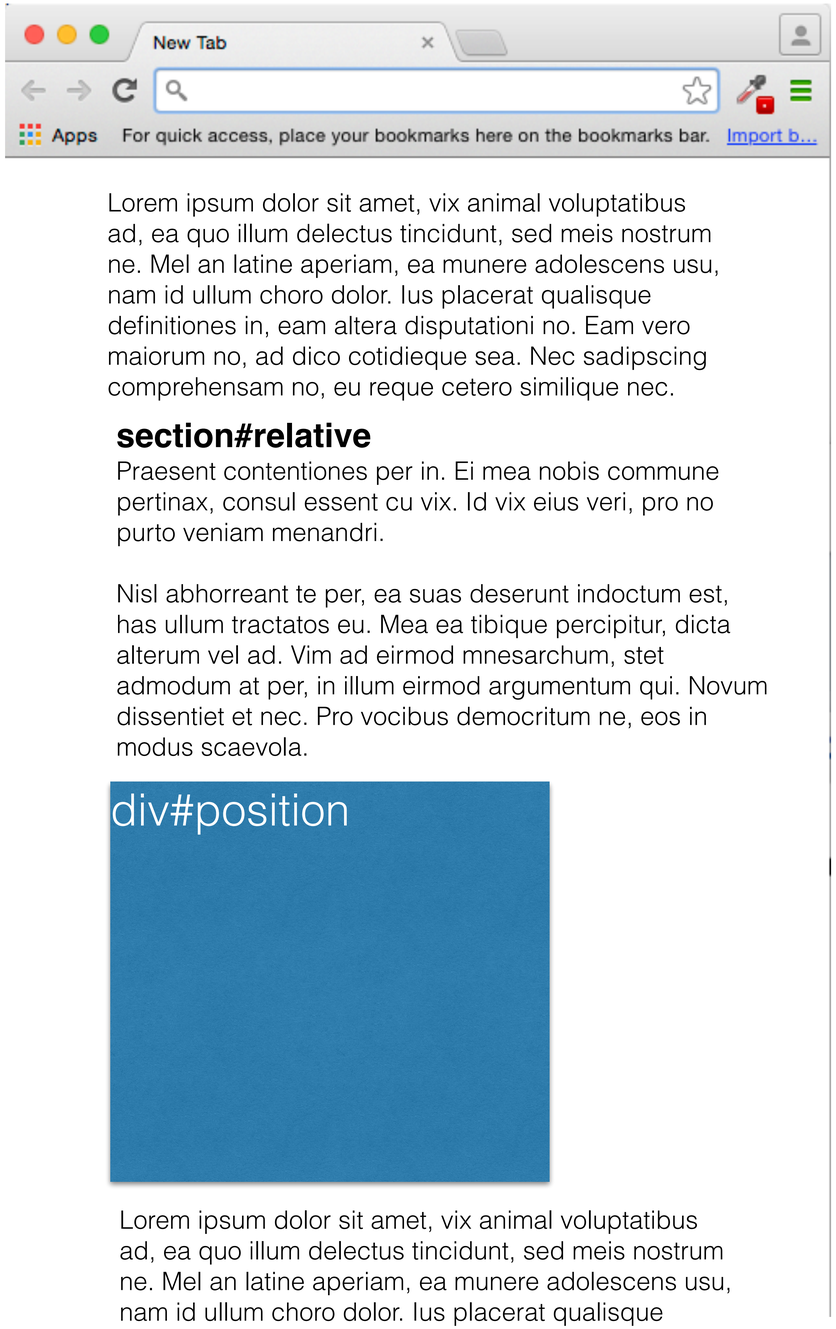
position: absolute (with positioned ancestor)

The elements is taken OUT of the flow (no space allocated for it on its parent).
the element is initially positioned where it would've been inserted if statically positioned.

If *left,top,right,bottom* properties are used, the element is positioned relative to its closest "positioned" (not static) ancestor.

```
section#relative { position: relative; }

div#position {
    position: static;
}
```



position: absolute (with positioned ancestor)

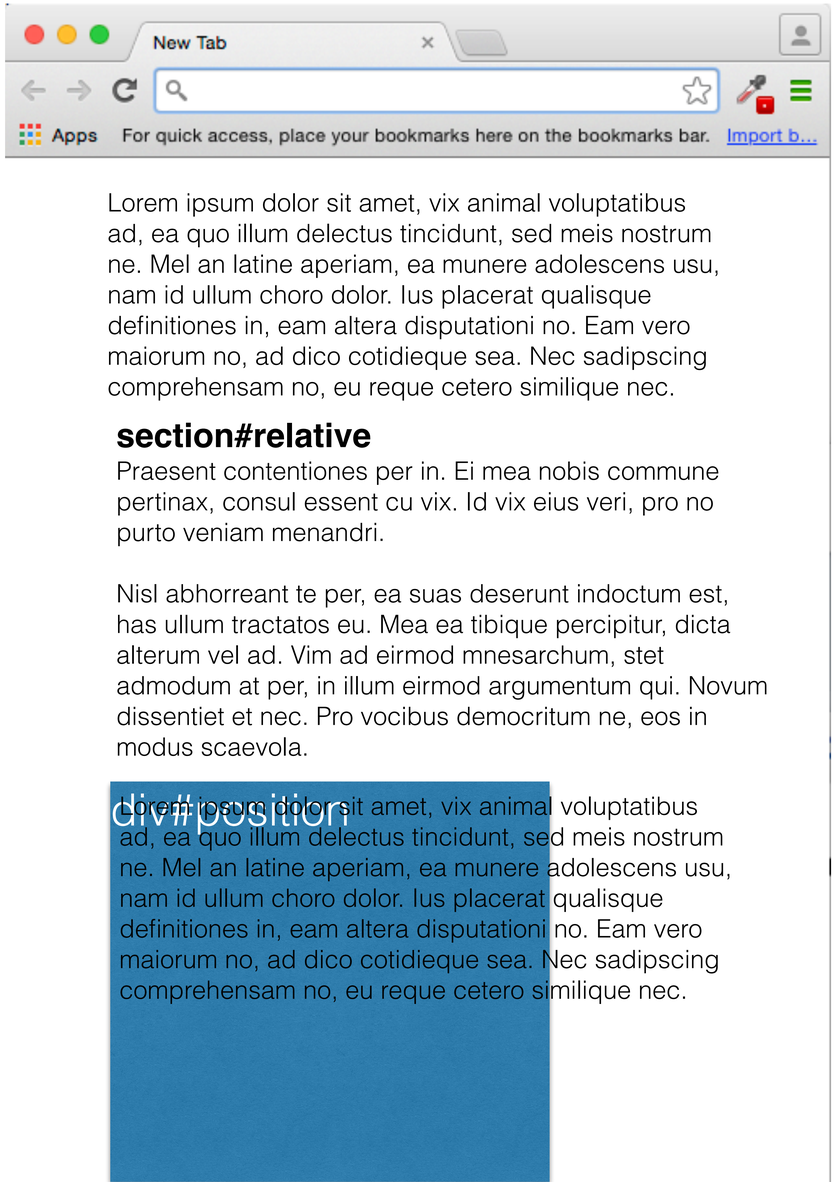
The elements is taken OUT of the flow (no space allocated for it on its parent).
the element is initially positioned where it would've been inserted if statically positioned.

If *left,top,right,bottom* properties are used, the element is positioned relative to its closest "positioned" (not static) ancestor.

```
section#relative { position: relative; }

div#position {
    position: absolute;
}
```

(the element is taken out of the flow and its space occupied from the element positioned after it in the document flow)



position: absolute (with positioned ancestor)

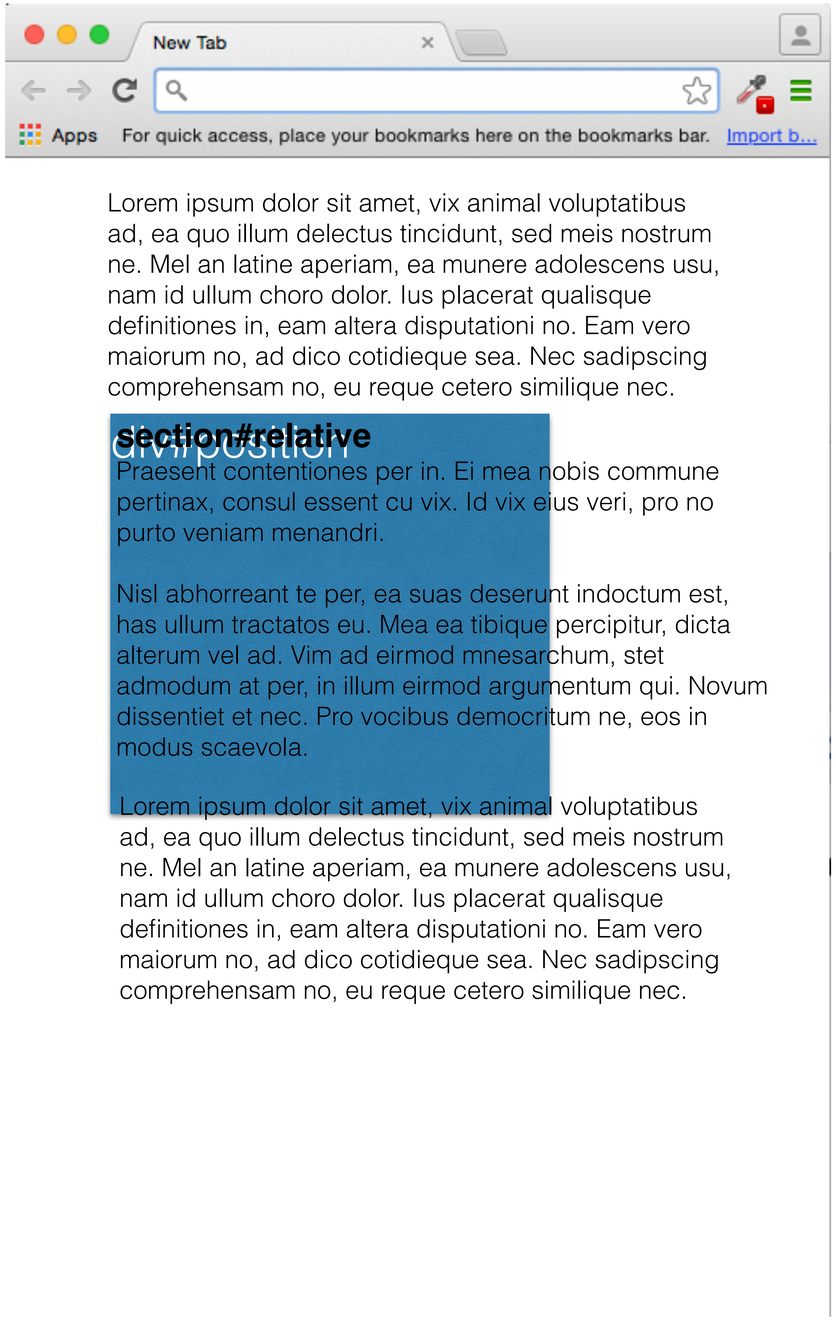
The elements is taken OUT of the flow (no space allocated for it on its parent).
the element is initially positioned where it would've been inserted if statically positioned.

If *left,top,right,bottom* properties are used, the element is positioned relative to its closest "positioned" (not static) ancestor.

```
section#relative { position: relative; }

div#position {
    position: absolute;
    top: 0;
}
```

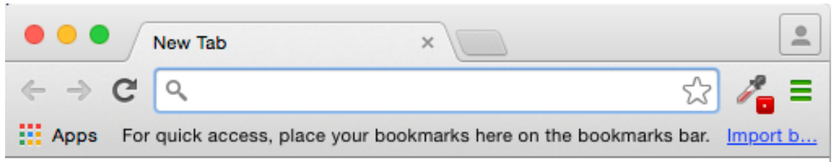
with a positioned ancestor (relative in this case), the element is then moved "relatively" to the position of its ancestor



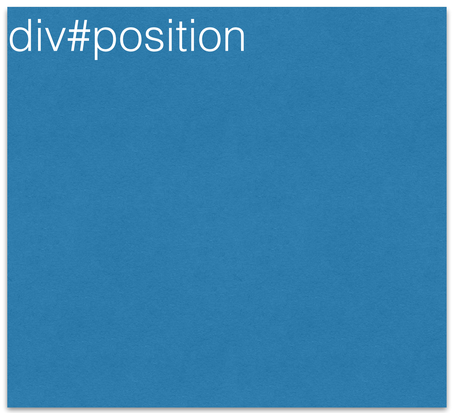
position: fixed

The elements is taken OUT of the flow (no space allocated for it on its parent).
The element is then positioned relatively to the viewport and its position doesn't change even when scrolling the page, resulting in the element overlapping with the content of the page

```
div#position {
    position: static;
}
```



Lorem ipsum dolor sit amet, vix animal voluptatibus ad, ea quo illum delectus tincidunt, sed meis nostrum ne. Mel an latine aperiam, ea munere adolescens usu, nam id ullum choro dolor. Ius placerat qualisque definitiones in, eam altera disputationi no. Eam vero maiorum no, ad dico cotidieque sea. Nec sadipscing comprehensam no, eu reque cetero similique nec.



Praesent contentiones per in. Ei mea nobis commune pertinax, consul essent cu vix. Id vix eius veri, pro no purto veniam menandri.

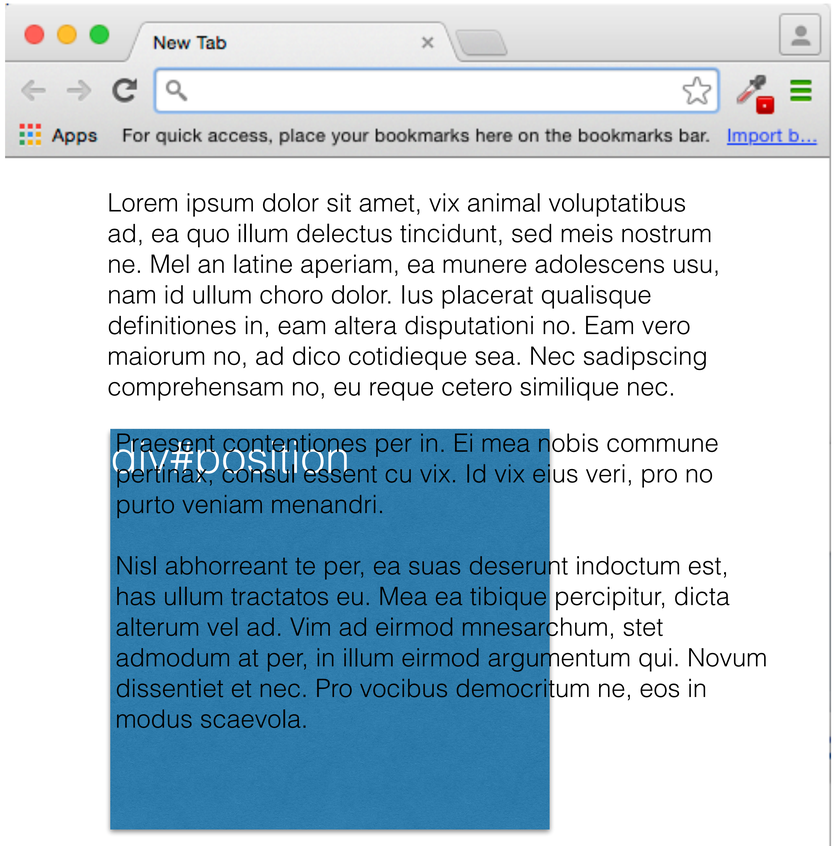
Nisl abhorreant te per, ea suas deserunt indoctum est, has ullum tractatos eu. Mea ea tibique percipitur, dicta alterum vel ad. Vim ad eirmod mnesarchum, stet admodum at per, in illum eirmod argumentum qui. Novum dissentiet et nec. Pro vocibus democritum ne, eos in modus scaevola.

position: fixed

The elements is taken OUT of the flow (no space allocated for it on its parent).
The element is then positioned relatively to the viewport and its position doesn't change even when scrolling the page, resulting in the element overlapping with the content of the page

```
div#position {
    position: fixed;
}
```

(the element is taken out of the flow and its space occupied from the element positioned after it in the document flow)

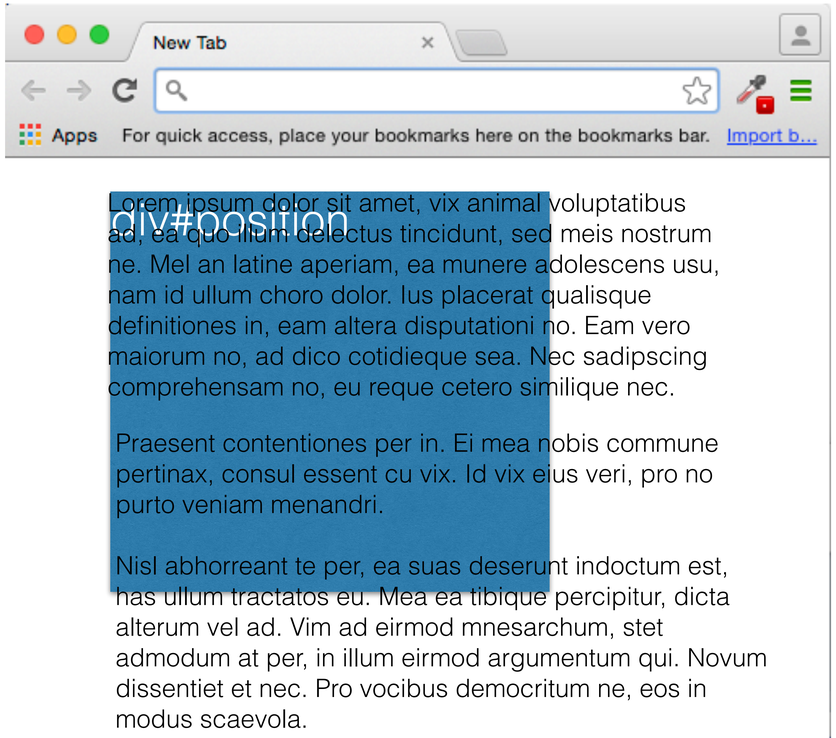


position: fixed

The elements is taken OUT of the flow (no space allocated for it on its parent).
The element is then positioned relatively to the viewport and its position doesn't change even when scrolling the page, resulting in the element overlapping with the content of the page

```
div#position {
    position: fixed;
    top: 0;
}
```

with or without a positioned ancestor, the element is always relative to entire viewport(so, top:0 means it will be moved to the top of the entire page)



position: fixed

The elements is taken OUT of the flow (no space allocated for it on its parent).
The element is then positioned relatively to the viewport and its position doesn't change even when scrolling the page, resulting in the element overlapping with the content of the page

```
div#position {
    position: fixed;
    top: 0;
}
```

with or without a positioned ancestor, the element is always relative to entire viewport(so, top:0 means it will be moved to the top of the entire page)