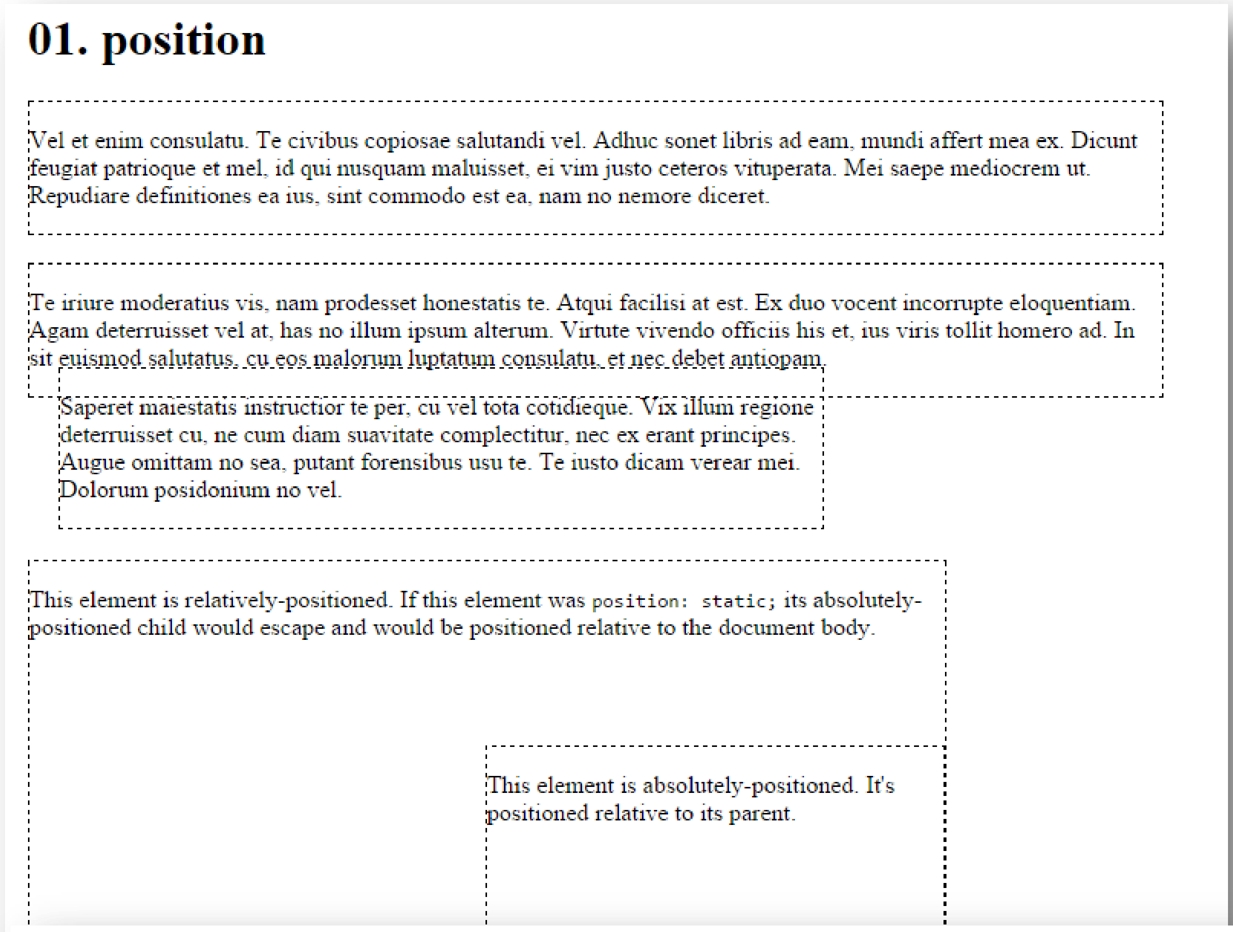
CSS Layout Part I

Web Development

Lab 06: Learn CSS Layout Part 1



01. position

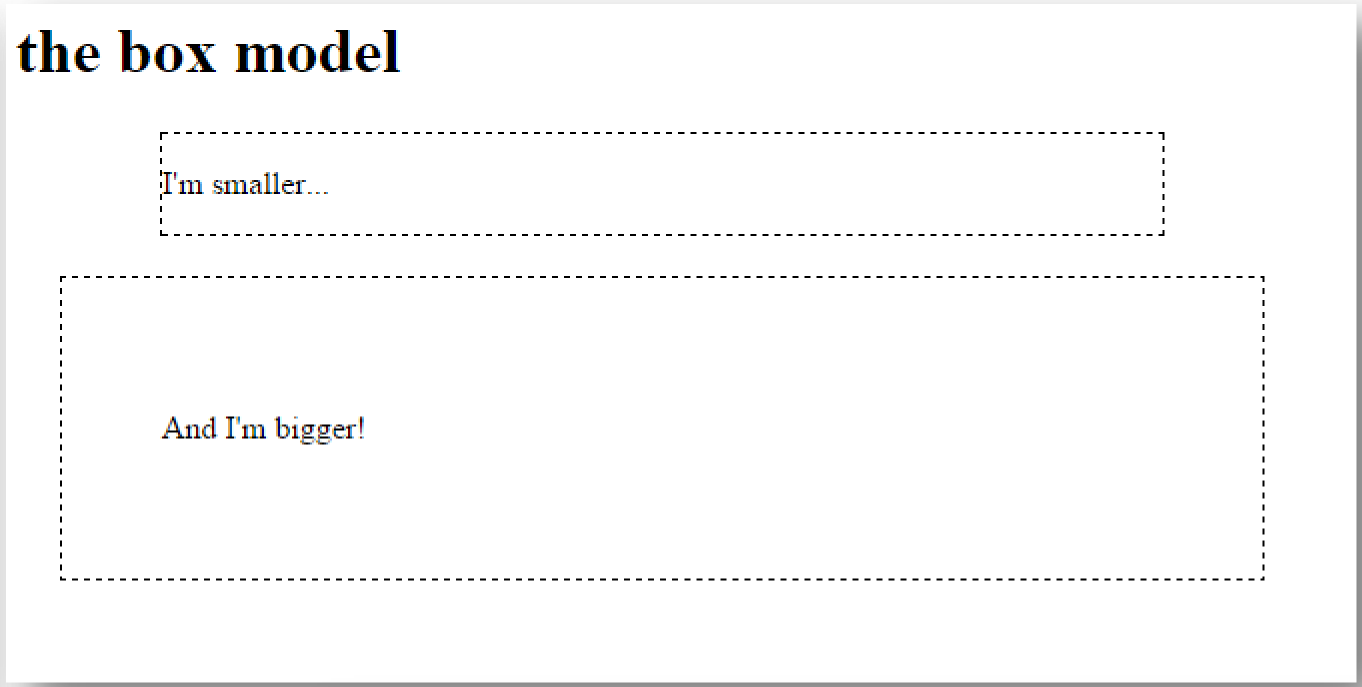
Vel et enim consulatu. Te civibus copiosae salutandi vel. Adhuc sonet libris ad eam, mundi affert mea ex. Dicunt feugiat patrioque et mel, id qui nusquam maluisset, ei vim justo ceteros vituperata. Mei saepe mediocrem ut. Repudiare definitiones ea ius, sint commodo est ea, nam no nemore diceret.

Te iriure moderatius vis, nam prodesset honestatis te. Atqui facilisi at est. Ex duo vocent incorrupte eloquentiam. Agam deterruisset vel at, has no illum ipsum alterum. Virtute vivendo officiis his et, ius viris tollit homero ad. In sit euismod salutatus, cu eos malorum luptatum consulatu, et nec debet antiopam.

Saperet maiestatis instructior te per, cu vel tota cotidieque. Vix illum regione deterruisset cu, ne cum diam suavitate complectitur, nec ex erant principes. Augue omittam no sea, putant forensibus usu te. Te iusto dicam verear mei. Dolorum posidonium no vel.

This element is relatively-positioned. If this element was `position: static;` its absolutely-positioned child would escape and would be positioned relative to the document body.

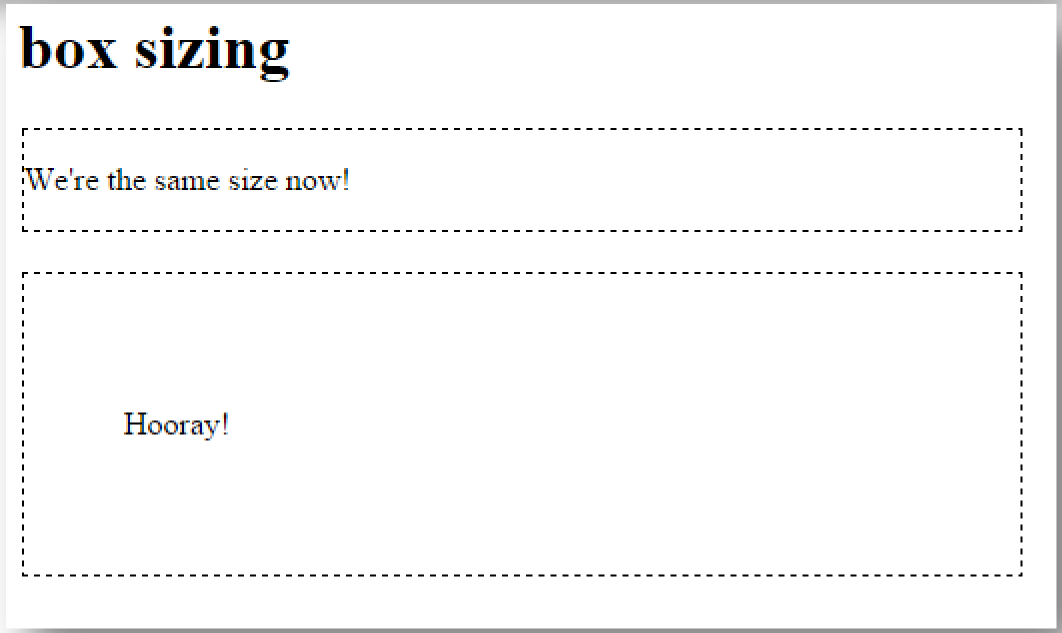
This element is absolutely-positioned. It's positioned relative to its parent.



the box model

I'm smaller...

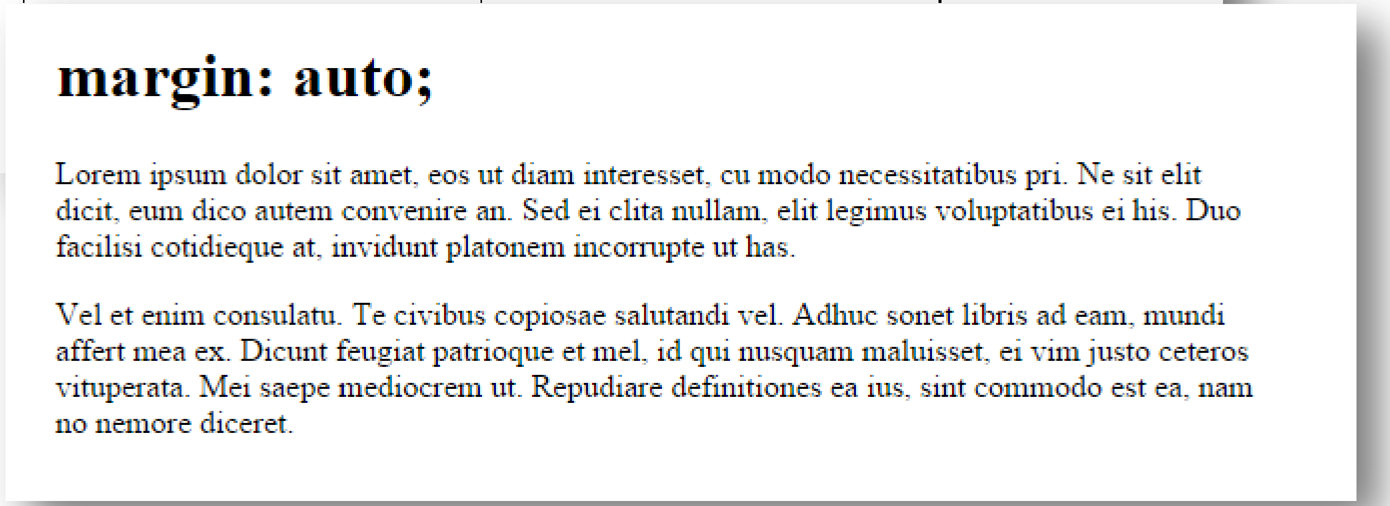
And I'm bigger!



box sizing

We're the same size now!

Hooray!



margin: auto;

Lorem ipsum dolor sit amet, eos ut diam interesset, cu modo necessitatibus pri. Ne sit elit dicit, eum dico autem convenire an. Sed ei clita nullam, elit legimus voluptatibus ei his. Duo facilisi cotidieque at, invidunt platonem incorrupte ut has.

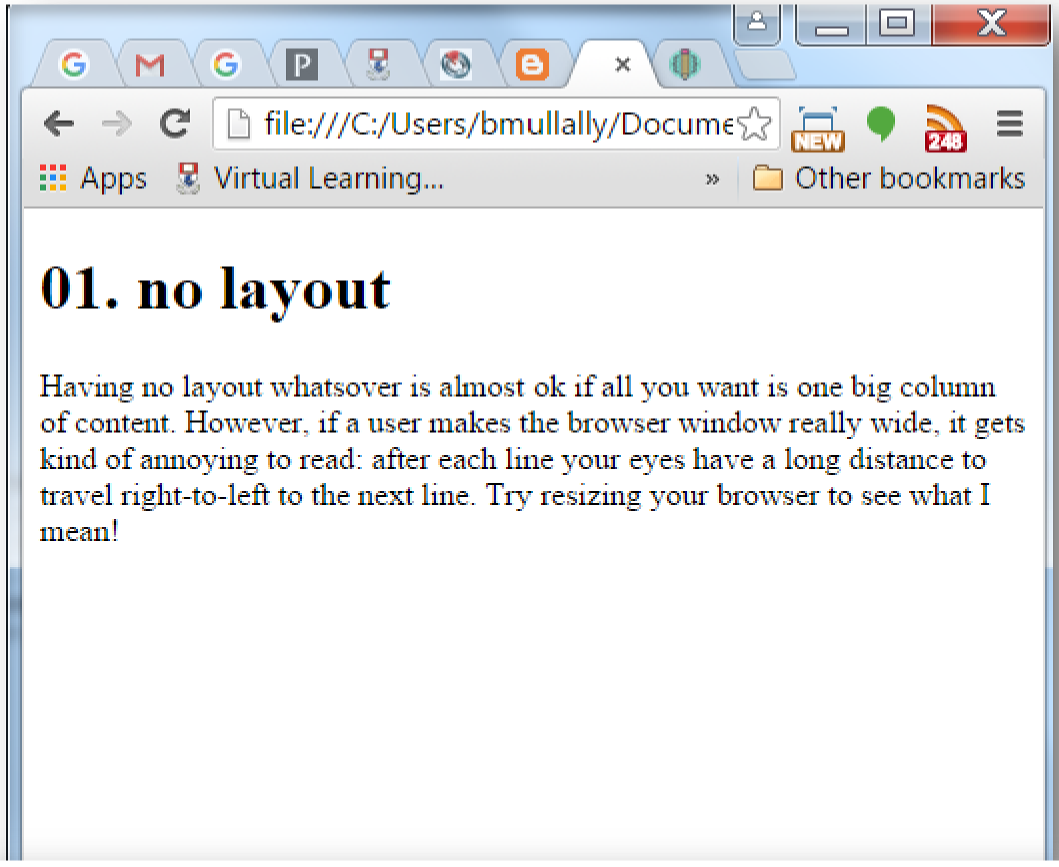
Vel et enim consulatu. Te civibus copiosae salutandi vel. Adhuc sonet libris ad eam, mundi affert mea ex. Dicunt feugiat patrioque et mel, id qui nusquam maluisset, ei vim justo ceteros vituperata. Mei saepe mediocrem ut. Repudiare definitiones ea ius, sint commodo est ea, nam no nemore diceret.

CSS Part 1

- No layout
- The “display” property
- Margin: auto;
- Max-width
- The box model
- Box-sizing
- Position
- Float
- Clear

No Layout

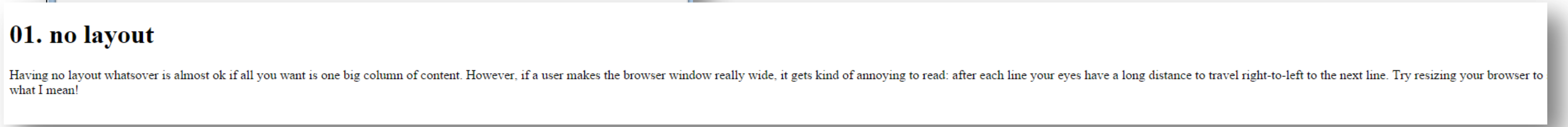
- No layout is ok if you want one big column of content.
- What happens if you make the browser really wide?



01. no layout

Having no layout whatsover is almost ok if all you want is one big column of content. However, if a user makes the browser window really wide, it gets kind of annoying to read: after each line your eyes have a long distance to travel right-to-left to the next line. Try resizing your browser to see what I mean!

Your eyes have to travel a long distance right to left to the next line.



01. no layout

Having no layout whatsover is almost ok if all you want is one big column of content. However, if a user makes the browser window really wide, it gets kind of annoying to read: after each line your eyes have a long distance to travel right-to-left to the next line. Try resizing your browser to what I mean!

“display” property

`display` is CSS’s most important property for controlling layout.

Every element has a default display value depending on what type of element it is. The default for most elements is usually `block` or `inline`.

block

`<div>` is the standard block level element. It starts on a new line and stretches to the left and right as far as it can. Others include `<p>` and new in HTML5 are `<header>` `<footer>` `<section>` and more.

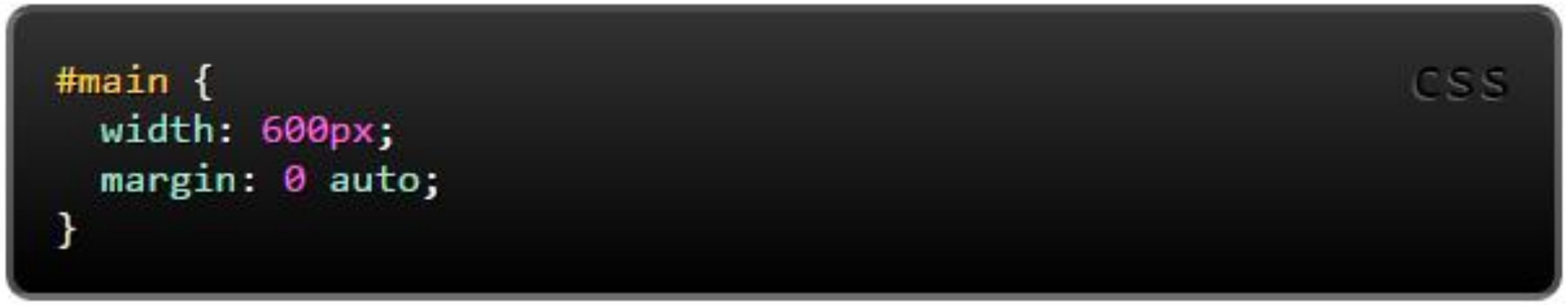
inline

`<span>` is the standard inline element. The `<a>` element is the most common inline element.

none

Another common display value is `none` . This is different from `visibility.` Setting `display` to `none` will render the page as though the element does not exist. `visibility: hidden;` will hide the element, but the element will take up the space it would usually.

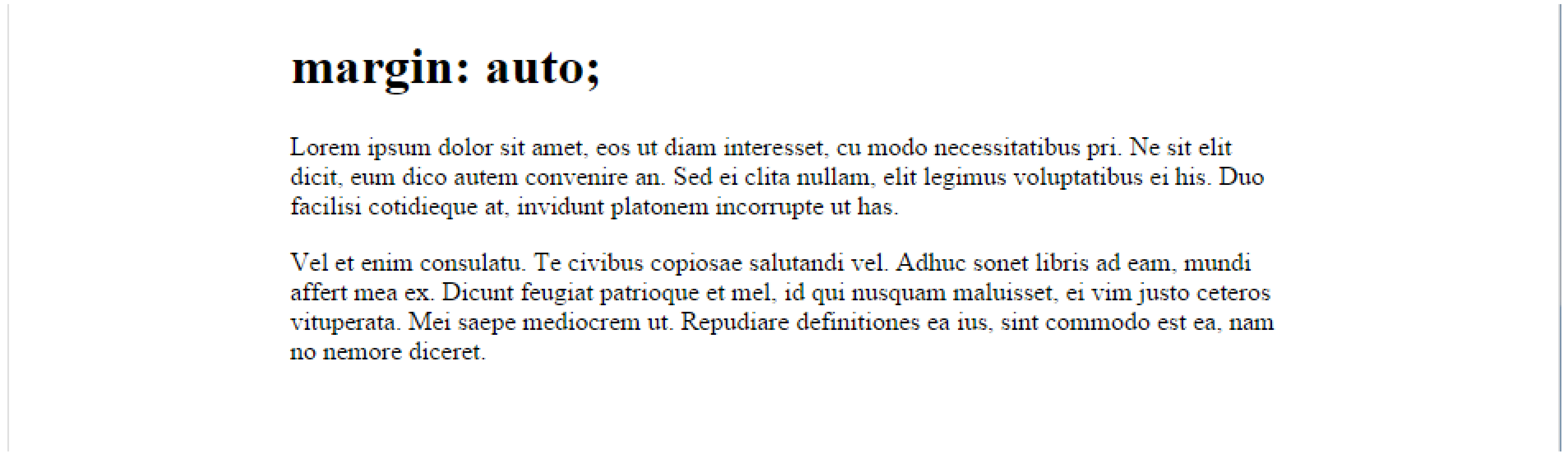
margin: auto;



```css
#main {
  width: 600px;
  margin: 0 auto;
}
```

CSS

- Setting the width of a block-level element will stop it stretching out to the edges left and right. Then set the margin to auto left and right. This horizontally centres that element within its container.
- The only problem is when the browser window is narrower than the width you set. What will the browser do?

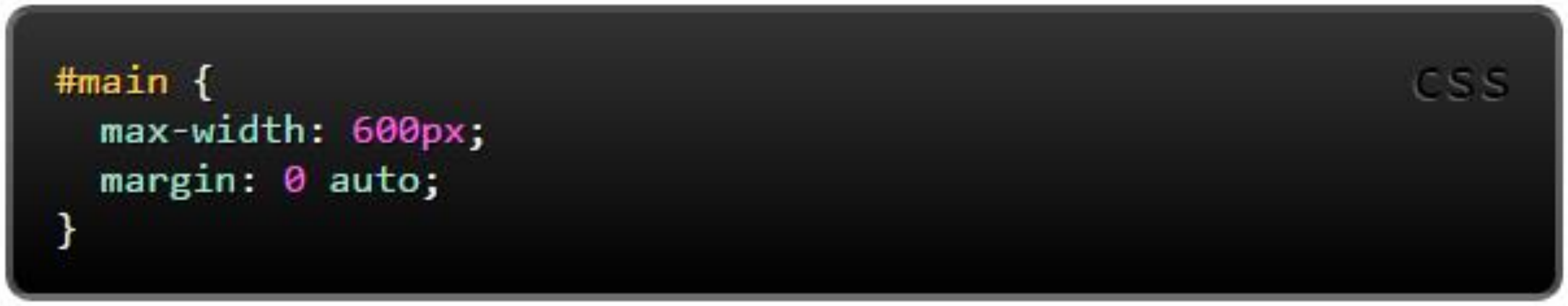


margin: auto;

Lorem ipsum dolor sit amet, eos ut diam interesset, cu modo necessitatibus pri. Ne sit elit dicit, eum dico autem convenire an. Sed ei clita nullam, elit legimus voluptatibus ei his. Duo facilisi cotidieque at, invidunt platonem incorrupte ut has.

Vel et enim consulatu. Te civibus copiosae salutandi vel. Adhuc sonet libris ad eam, mundi affert mea ex. Dicunt feugiat patrioque et mel, id qui nusquam maluisset, ei vim justo ceteros vituperata. Mei saepe mediocrem ut. Repudiare definitiones ea ius, sint commodo est ea, nam no nemore diceret.

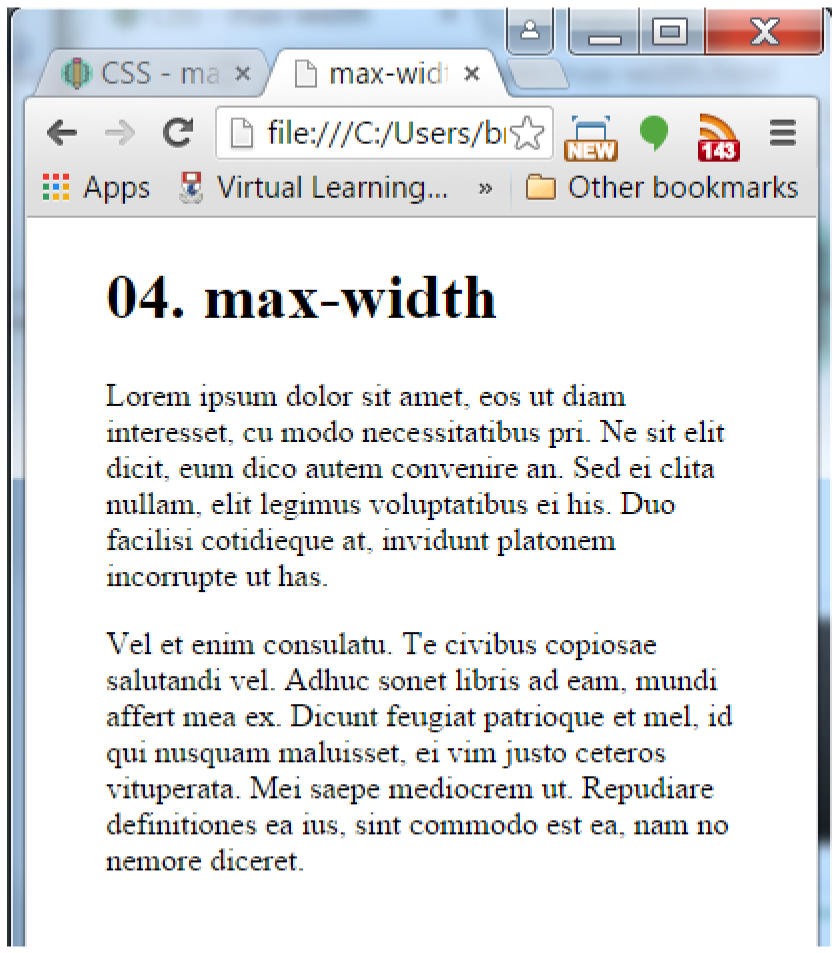
max-width



```css
#main {
  max-width: 600px;
  margin: 0 auto;
}
```

- To prevent a horizontal scroll bar appearing we can use the `max-width` property. For a site that needs to be usable on a mobile it is important to use `max-width`.

- `max-width` is supported by all major browsers including IE7



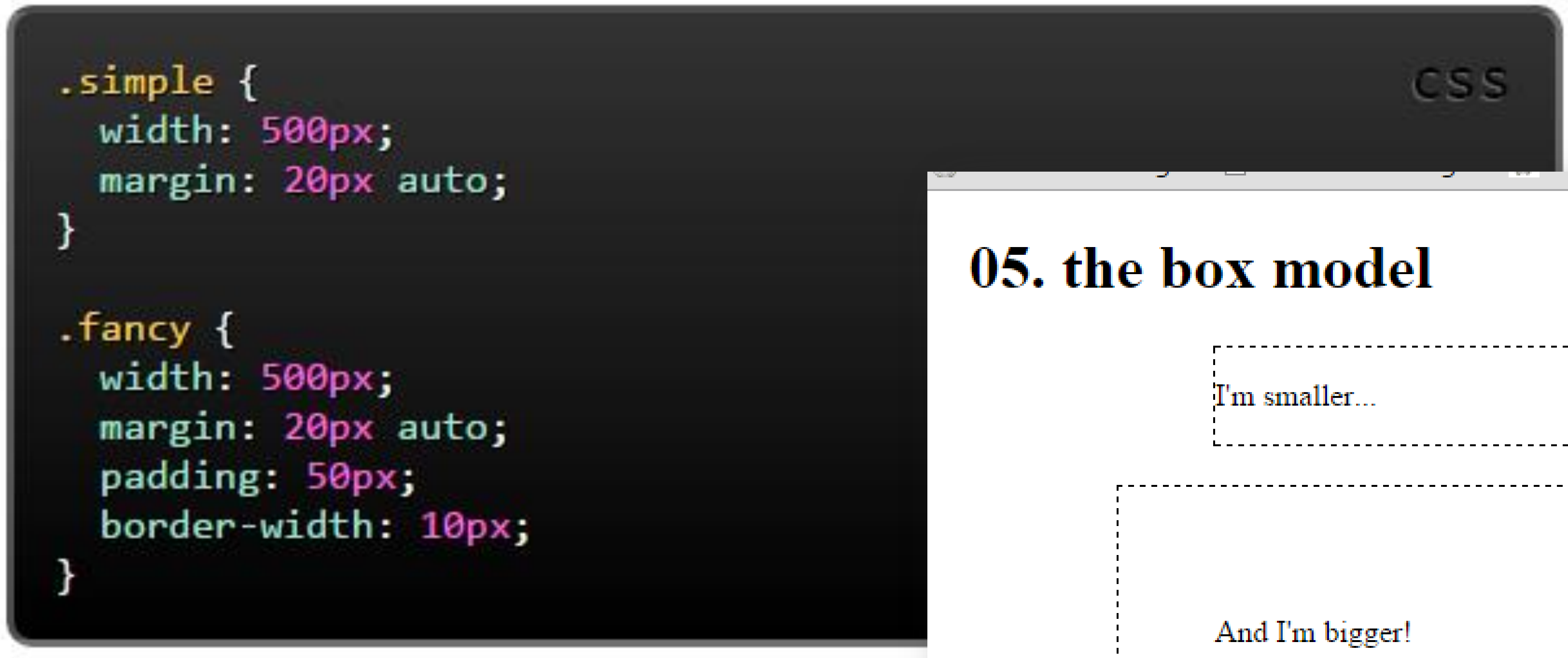
04. max-width

Lorem ipsum dolor sit amet, eos ut diam interesset, cu modo necessitatibus pri. Ne sit elit dicit, eum dico autem convenire an. Sed ei clita nullam, elit legimus voluptatibus ei his. Duo facilisi cotidieque at, invidunt platonem incorrupte ut has.

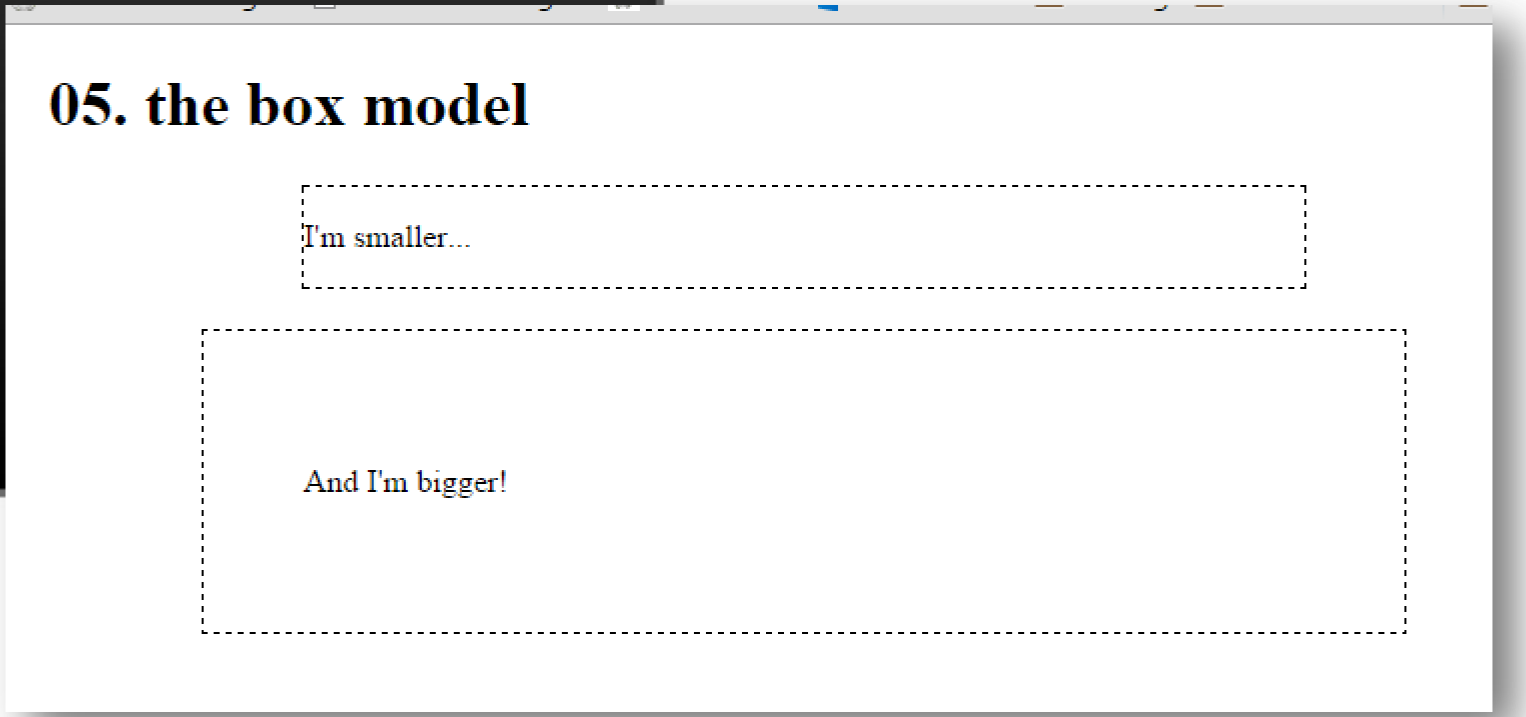
Vel et enim consulatu. Te civibus copiosae salutandi vel. Adhuc sonet libris ad eam, mundi affert mea ex. Dicunt feugiat patrioque et mel, id qui nusquam maluisset, ei vim justo ceteros vituperata. Mei saepe mediocrem ut. Repudiare definitiones ea ius, sint commodo est ea, nam no nemore diceret.

the box model

- the difficulty with setting width is the box model. When you set the width of an element, it can actually appear bigger because of the border and padding.
- These two elements have their width set the same but they end up rendered as different sizes.
- The solution has been to write smaller width value than wanted, but we don't have to do that any more.



```css
.simple {
  width: 500px;
  margin: 20px auto;
}

.fancy {
  width: 500px;
  margin: 20px auto;
  padding: 50px;
  border-width: 10px;
}
```



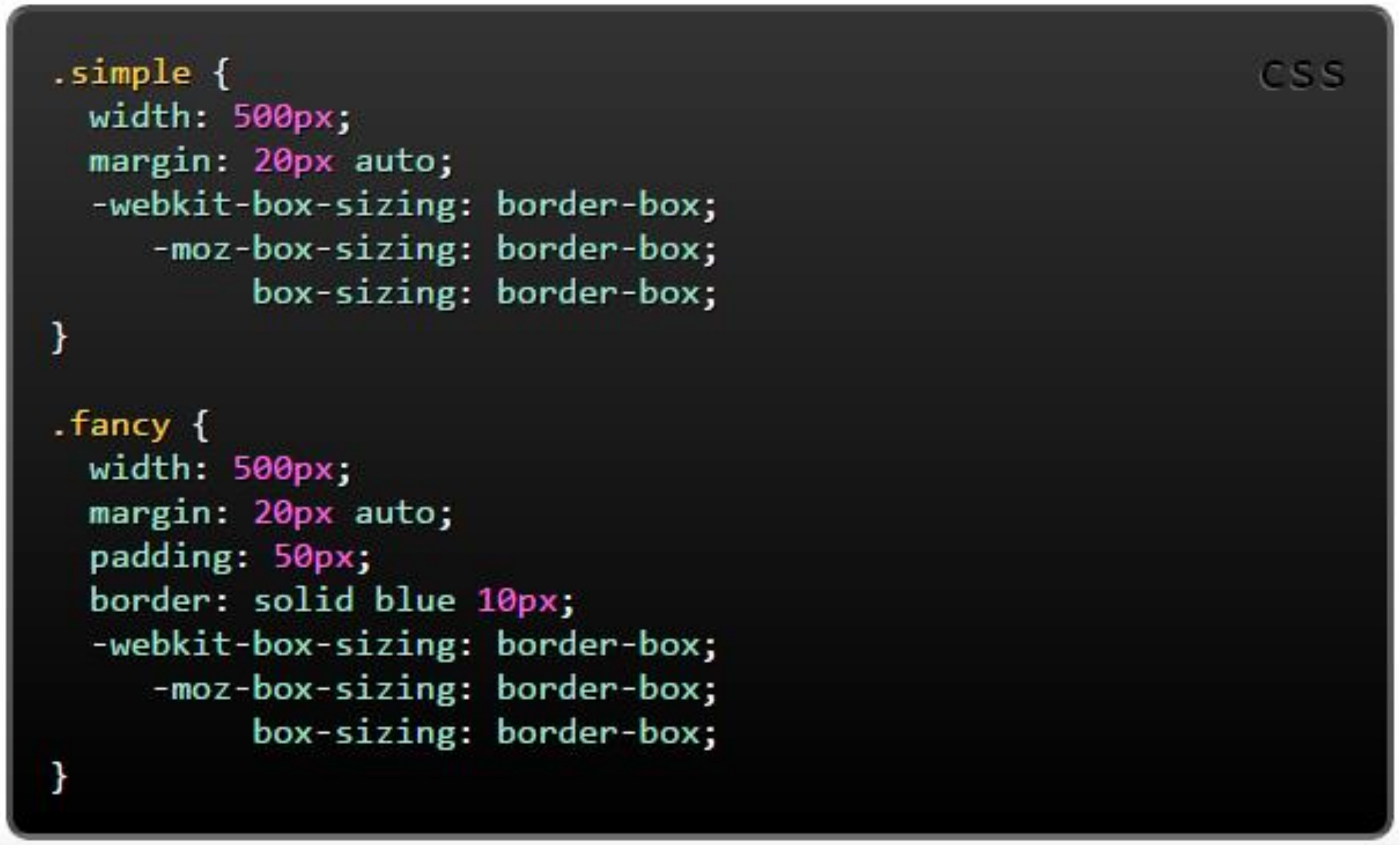
05. the box model

I'm smaller...

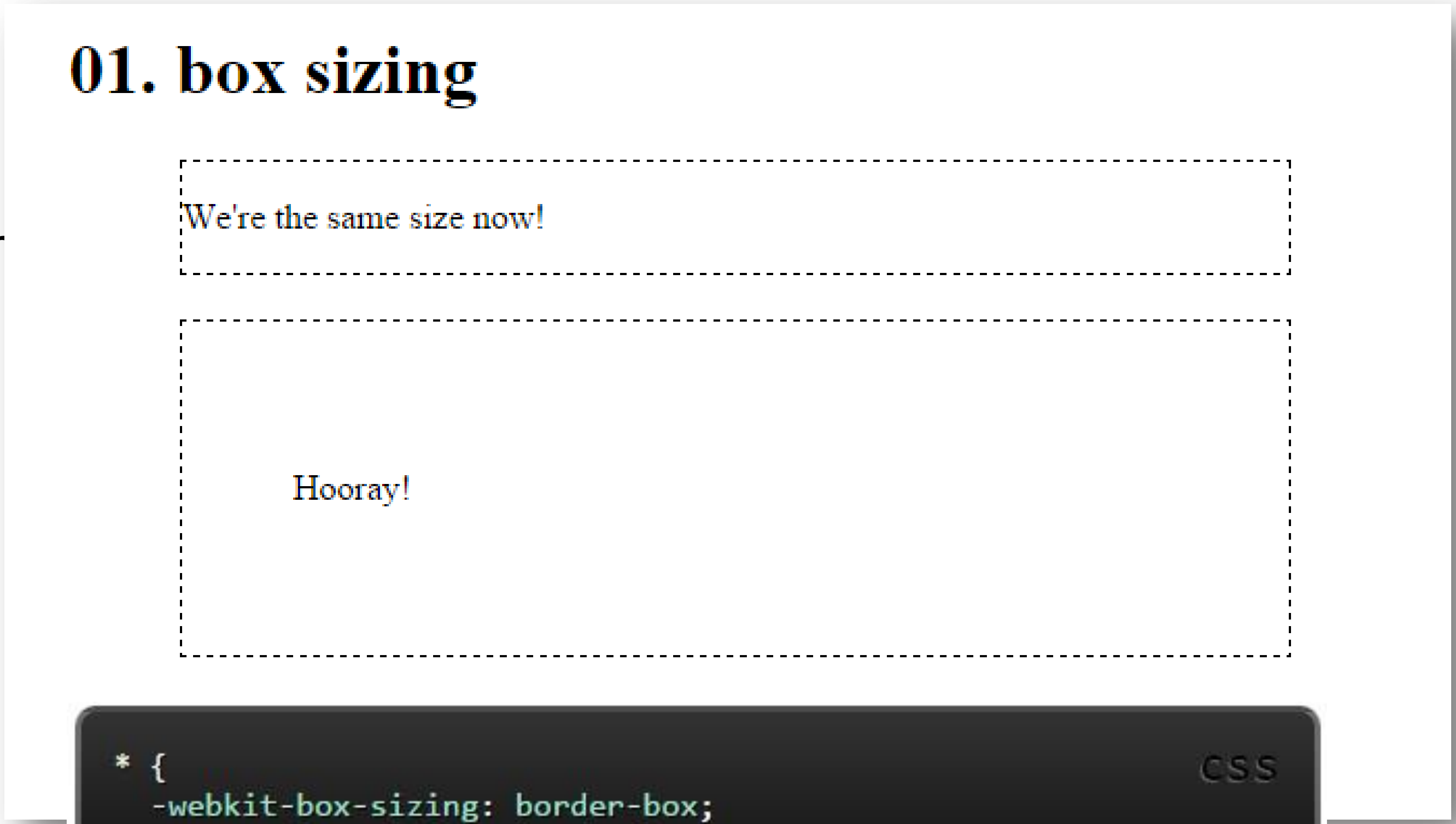
And I'm bigger!

box-sizing

- A new CSS property called `box-sizing` was created. When you set `box-sizing: border-box;` on an element the padding and border of that element no longer increases its width.
- Here we have set the `box-sizing: border-box;` on both elements.
- Since this is so much better, some authors want all elements on their pages to work this way.
- You should use –webkit- and –moz- prefixes



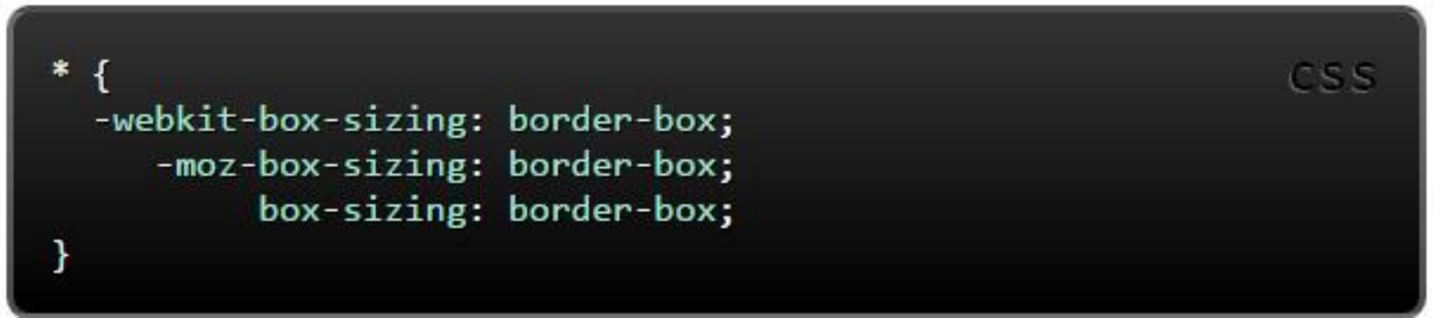
```css
.simple {
  width: 500px;
  margin: 20px auto;
  -webkit-box-sizing: border-box;
     -moz-box-sizing: border-box;
          box-sizing: border-box;
}

.fancy {
  width: 500px;
  margin: 20px auto;
  padding: 50px;
  border: solid blue 10px;
  -webkit-box-sizing: border-box;
     -moz-box-sizing: border-box;
          box-sizing: border-box;
}
```



01. box sizing

We're the same size now!

Hooray!

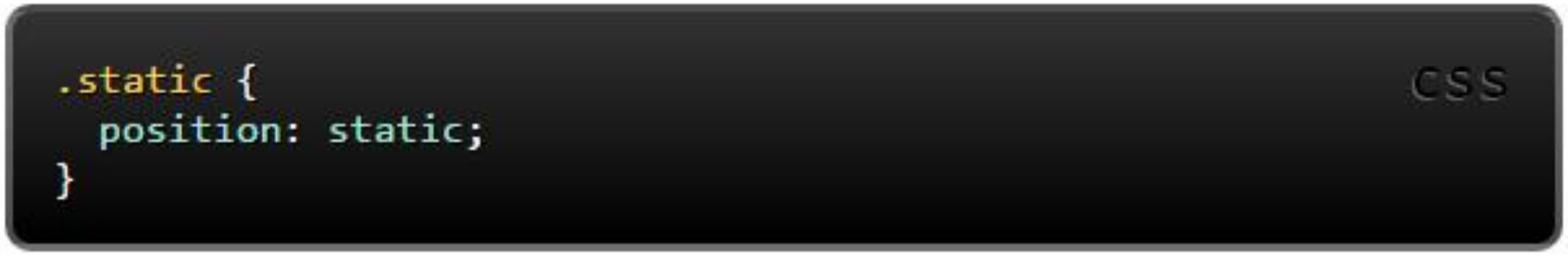


```css
* {
  -webkit-box-sizing: border-box;
     -moz-box-sizing: border-box;
          box-sizing: border-box;
}
```

position

- In order to make more complex layouts, we need to discuss the `position` property. It has a bunch of possible values, and their names make no sense and are impossible to remember.

- **Static**



```css
.static {
  position: static;
}
```

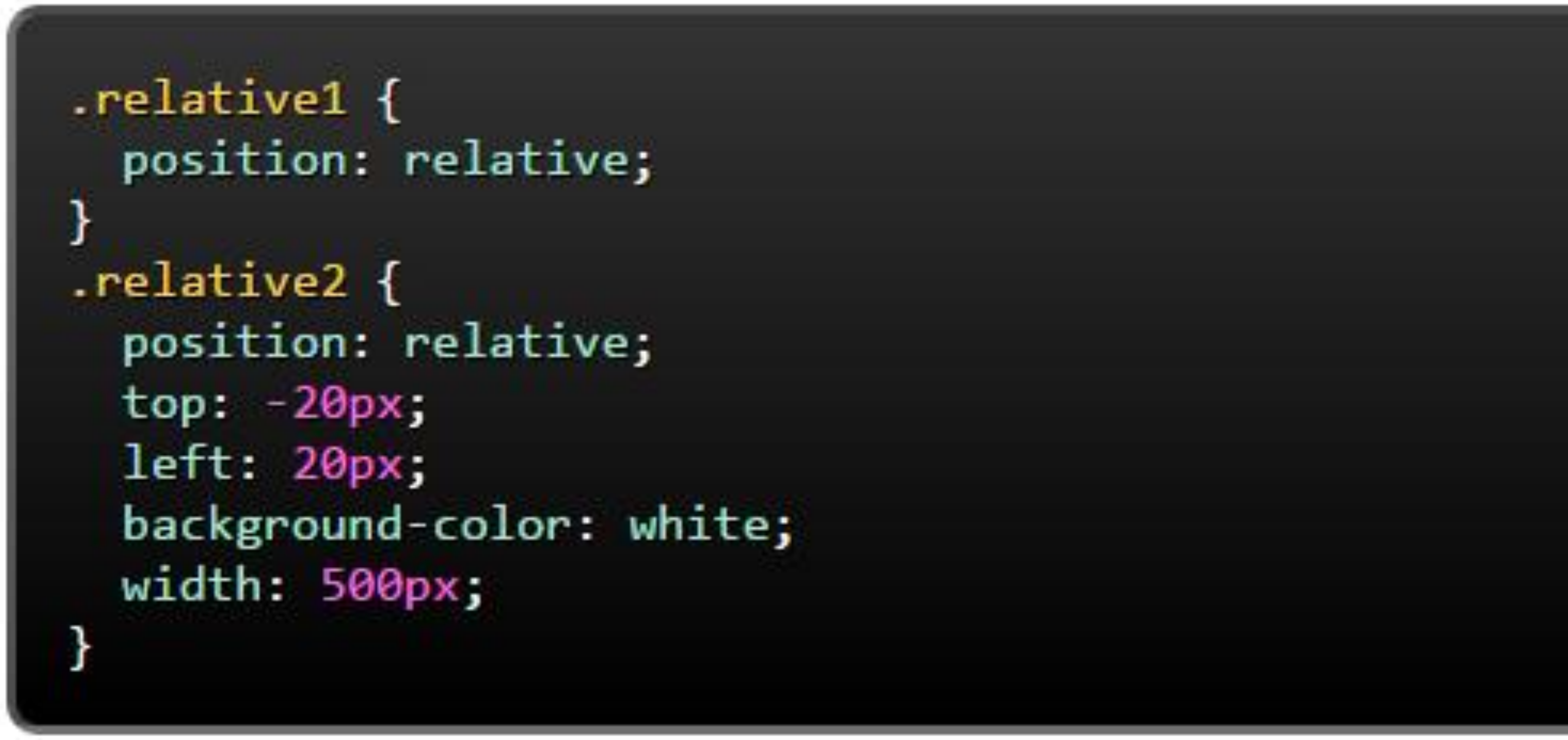
- static is the default value. An element with `position: static;` is not positioned in any special way.

- A static element is said to be *not positioned* and an element with its position set to anything else is said to be *positioned*.

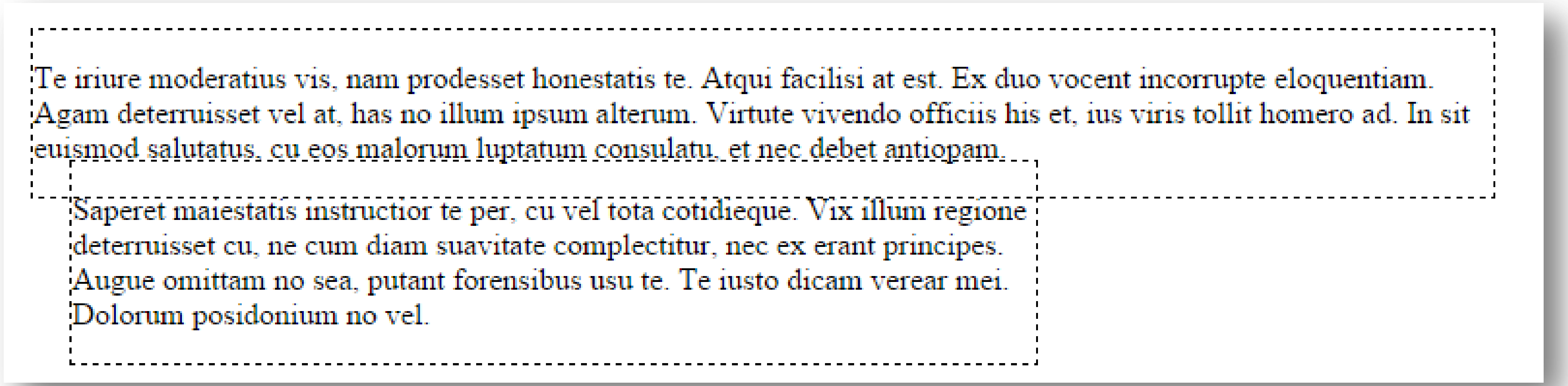
position

- **relative**

- relative behaves the same was as static unless you add some extra properties to adjust the element away from its normal position. Other content will not be adjusted to fit into any gap left by that element.



```
.relative1 {
  position: relative;
}
.relative2 {
  position: relative;
  top: -20px;
  left: 20px;
  background-color: white;
  width: 500px;
}
```



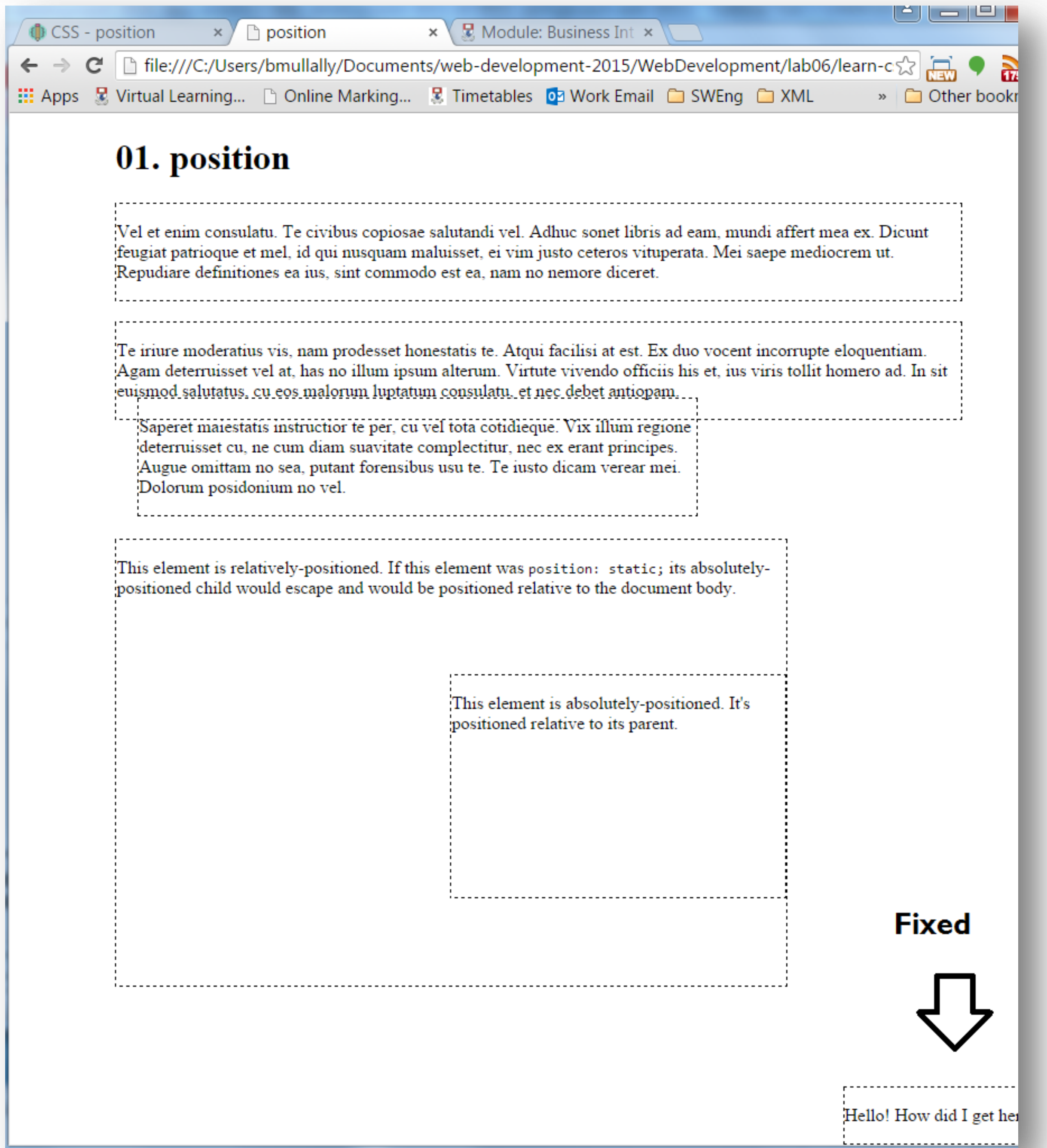
Te iriure moderatius vis, nam prodesset honestatis te. Atqui facilisi at est. Ex duo vocent incorrupte eloquentiam. Agam deterruisset vel at, has no illum ipsum alterum. Virtute vivendo officiis his et, ius viris tollit homero ad. In sit euismod salutatus, cu eos malorum luptatum consulatu, et nec debet antiopam.

Saperet maiestatis instructior te per, cu vel tota cotidieque. Vix illum regione deterruisset cu, ne cum diam suavitate complectitur, nec ex erant principes. Augue omittam no sea, putant forensibus usu te. Te iusto dicam verear mei. Dolorum posidonium no vel.

position

```css
.fixed {
  position: fixed;
  bottom: 0;
  right: 0;
  width: 200px;
  background-color: white;
}
```

- **fixed**
- A **fixed** element is positioned **relative** to the viewport, which means it always stays in the same place even if the page is scrolled.
- As with **relative** the **top**, **right**, **bottom**, and **left** properties are used.
- A **fixed** element does not leave a gap in the page where it would normally have been located.



01. position

Vel et enim consulatu. Te civibus copiosae salutandi vel. Adhuc sonet libris ad eam, mundi affert mea ex. Dicunt feugiat patrioque et mel, id qui nusquam maluisset, ei vim justo ceteros vituperata. Mei saepe mediocrem ut. Repudiare definitiones ea ius, sint commodo est ea, nam no nemore diceret.

Te iriure moderatius vis, nam prodesset honestatis te. Atqui facilisi at est. Ex duo vocent incorrupte eloquentiam. Agam deterruisset vel at, has no illum ipsum alterum. Virtute vivendo officiis his et, ius viris tollit homero ad. In sit euismod salutatus, cu eos malorum luptatum consulatu, et nec debet antiopam.

Saperet maiestatis instructior te per, cu vel tota cotidieque. Vix illum regione deterruisset cu, ne cum diam suavitate complectitur, nec ex erant principes. Augue omittam no sea, putant forensibus usu te. Te iusto dicam verear mei. Dolorum posidonium no vel.

This element is relatively-positioned. If this element was `position: static;` its absolutely-positioned child would escape and would be positioned relative to the document body.

This element is absolutely-positioned. It's positioned relative to its parent.

Fixed

⇩

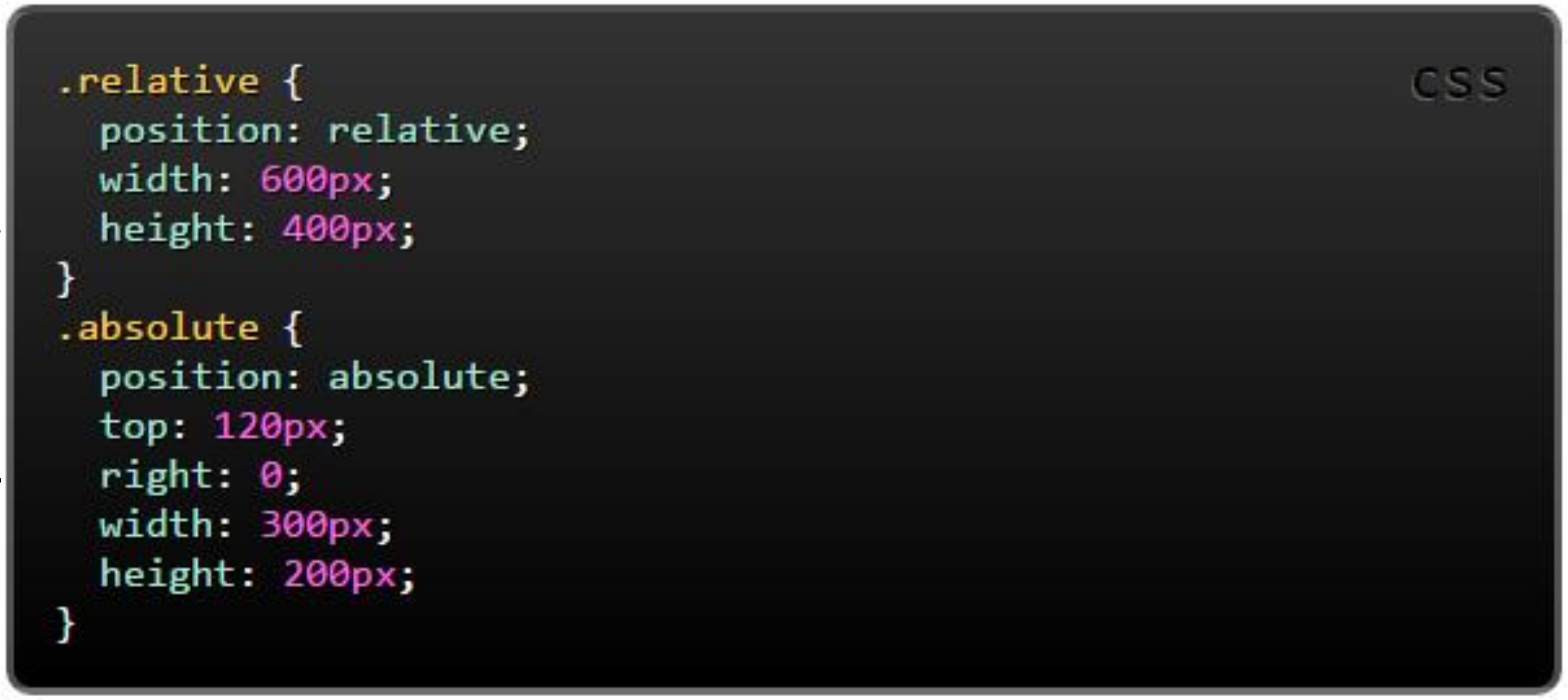
Hello! How did I get her

position

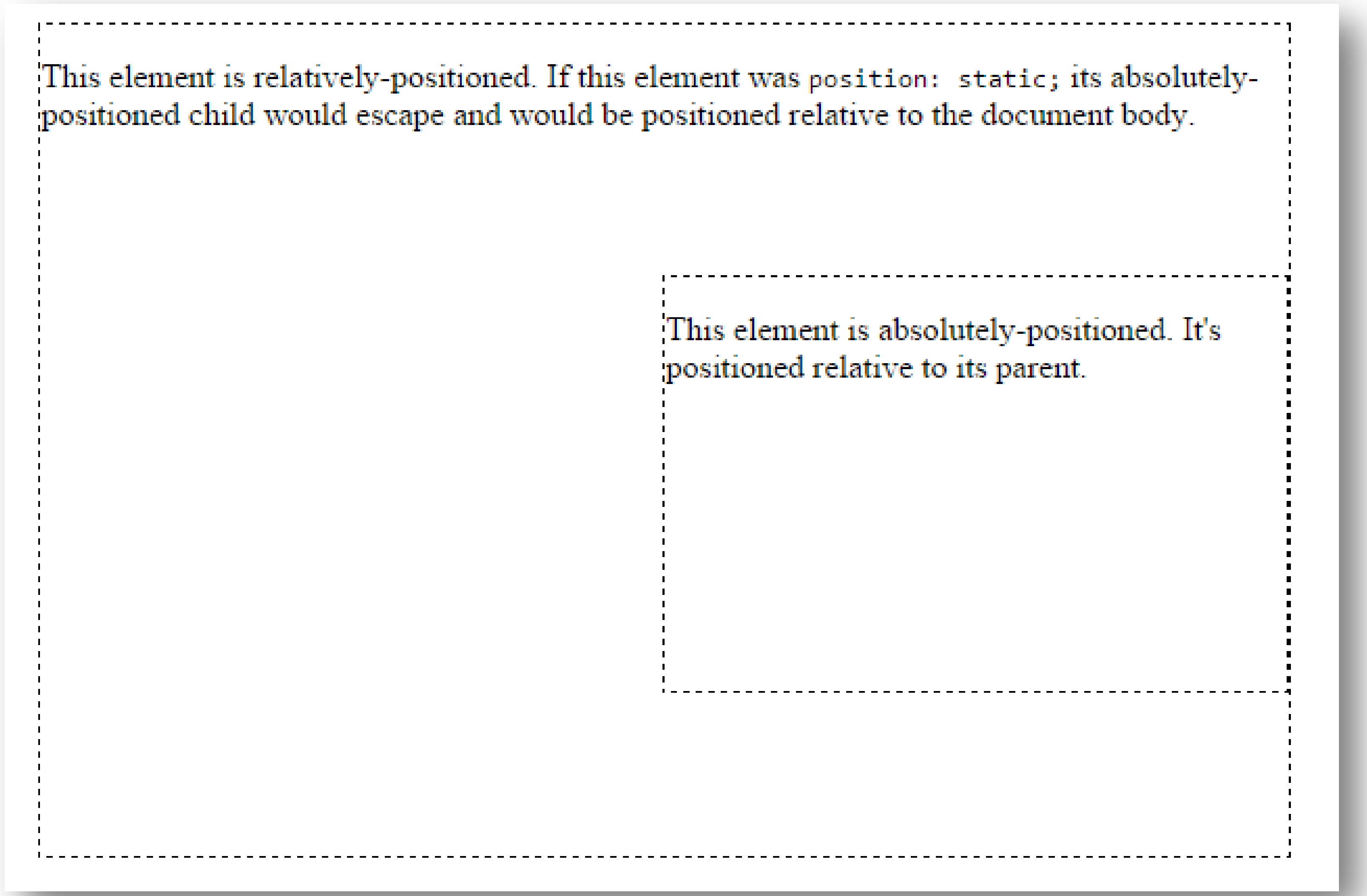
- **absolute**

- absolute is the trickiest position value. absolute behaves like fixed except relative to the nearest positioned ancestor instead of relative to the viewport. If an absolutely positioned element has no positioned ancestors, it uses the document body, and still moves along the page scrolling.

- This is tricky, but essential for creating great CSS layouts.



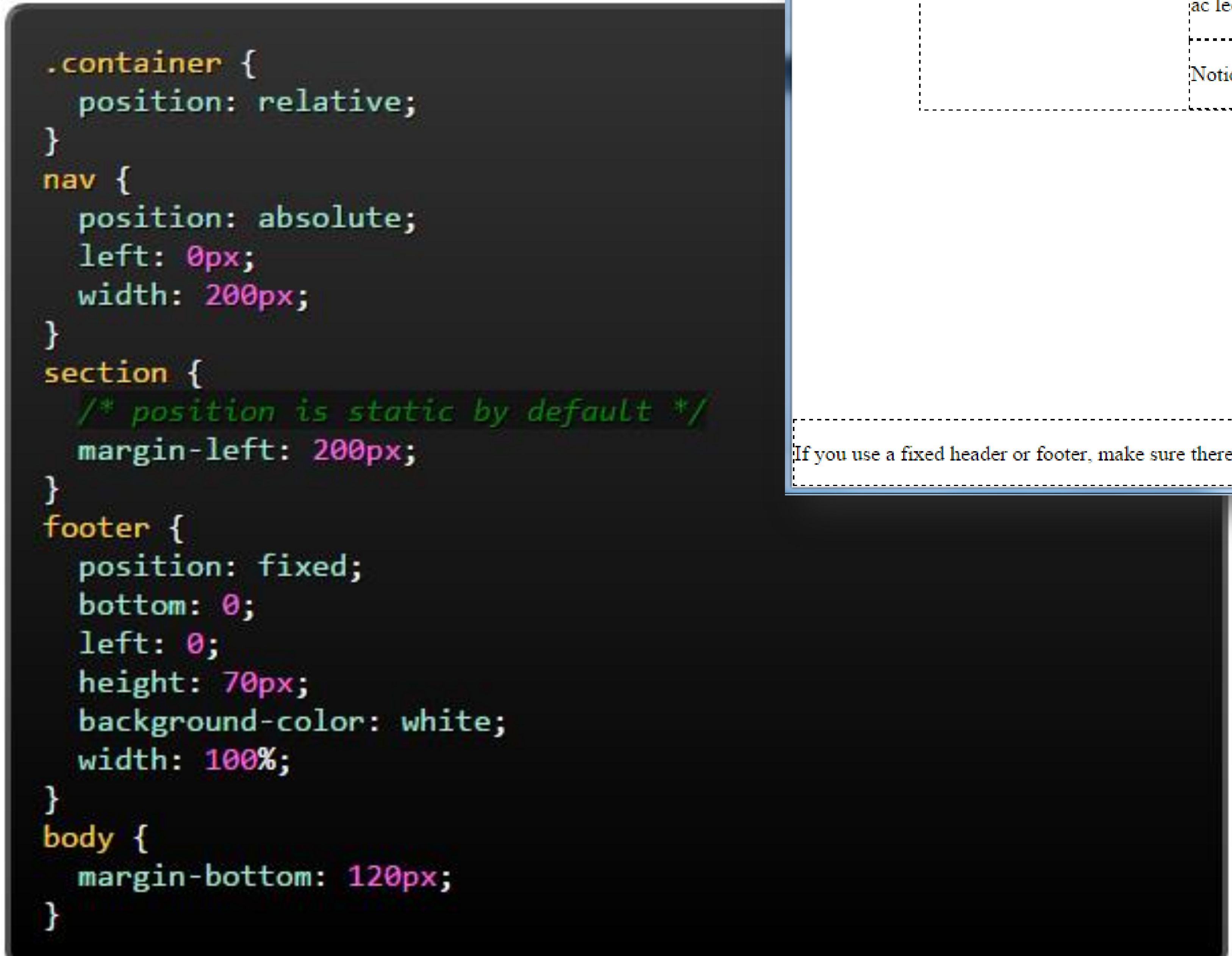
```css
.relative {
  position: relative;
  width: 600px;
  height: 400px;
}
.absolute {
  position: absolute;
  top: 120px;
  right: 0;
  width: 300px;
  height: 200px;
}
```



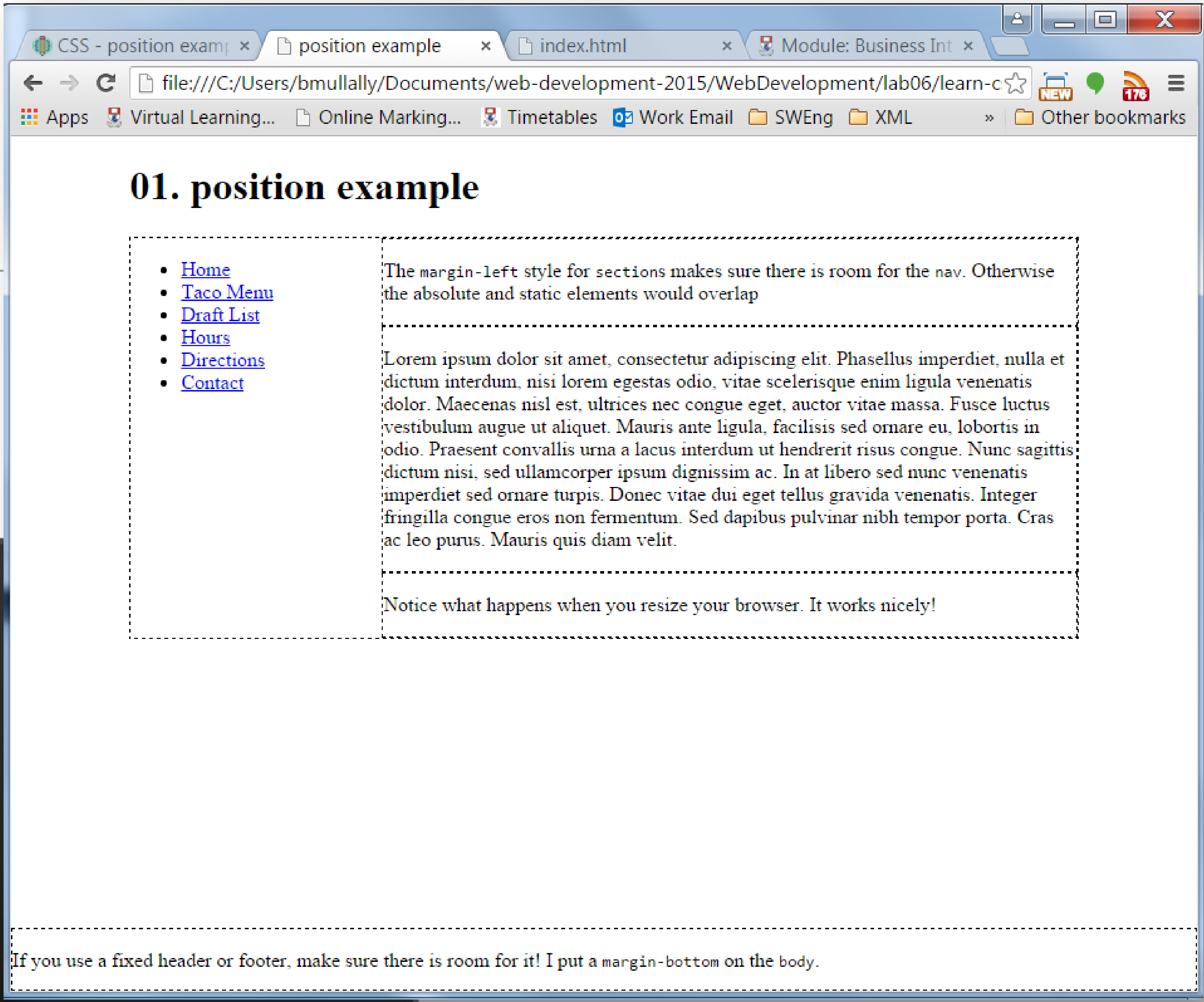
This element is relatively-positioned. If this element was `position: static;` its absolutely-positioned child would escape and would be positioned relative to the document body.

This element is absolutely-positioned. It's positioned relative to its parent.

position example



```
.container {
  position: relative;
}
nav {
  position: absolute;
  left: 0px;
  width: 200px;
}
section {
  /* position is static by default */
  margin-left: 200px;
}
footer {
  position: fixed;
  bottom: 0;
  left: 0;
  height: 70px;
  background-color: white;
  width: 100%;
}
body {
  margin-bottom: 120px;
}
```



01. position example

- Home
- Taco Menu
- Draft List
- Hours
- Directions
- Contact

The `margin-left` style for `sections` makes sure there is room for the `nav`. Otherwise the absolute and static elements would overlap

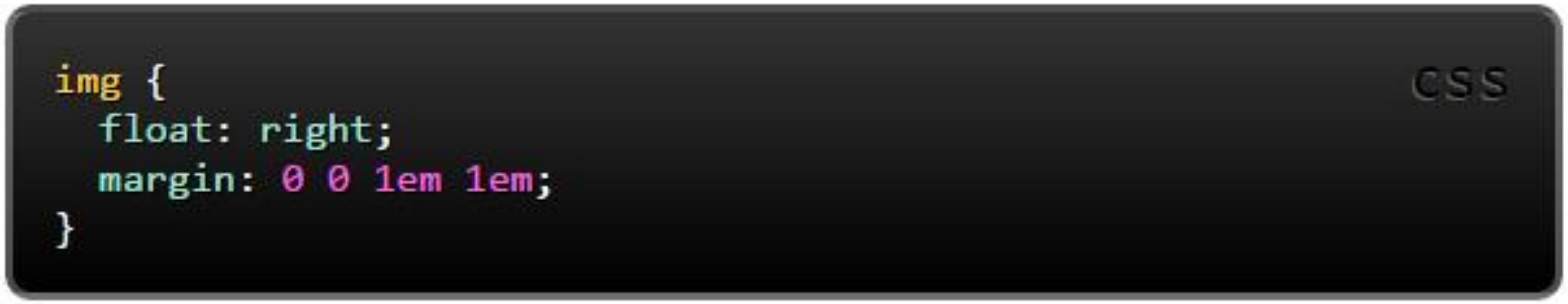
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.

Notice what happens when you resize your browser. It works nicely!

If you use a fixed header or footer, make sure there is room for it! I put a `margin-bottom` on the `body`.

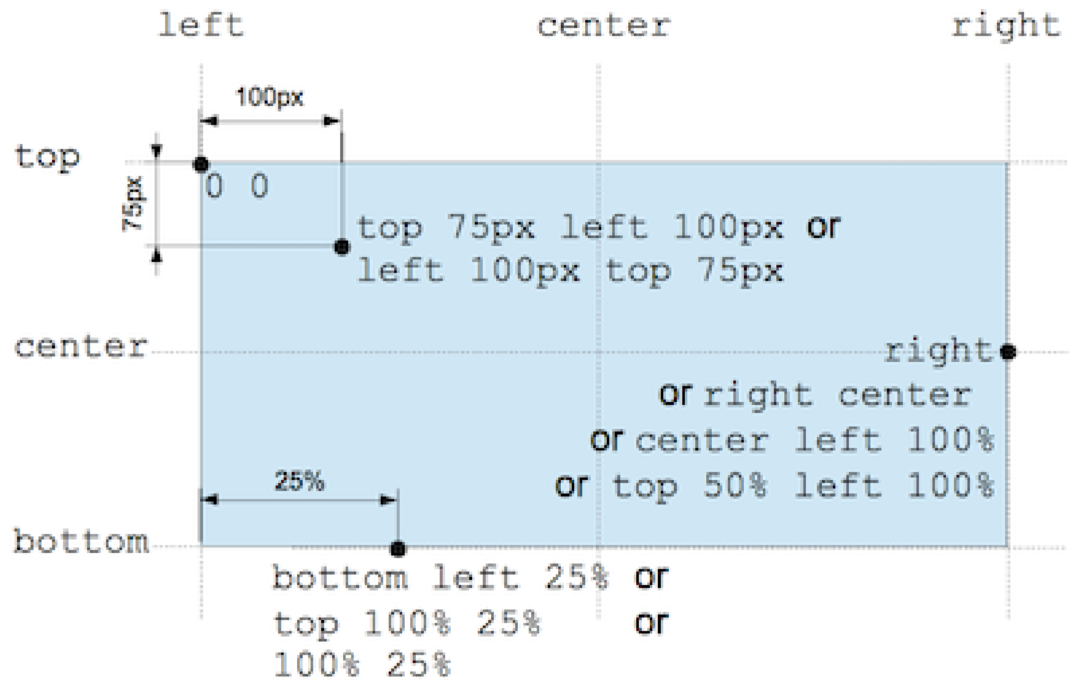
float

- another CSS property used for layout is float. Float is intended for wrapping text around images:



```css
img {
  float: right;
  margin: 0 0 1em 1em;
}
```

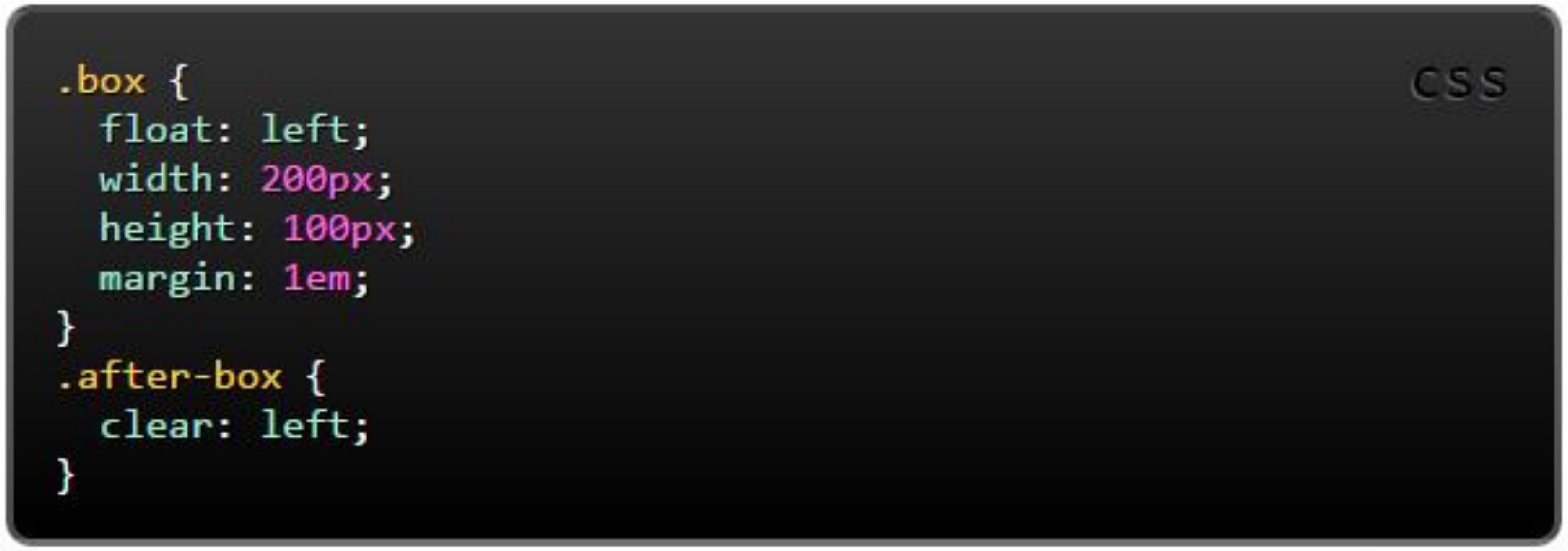
the position CSS data type denotes a coordinate in a 2D space used to set a location relative to a box. A specific coordinate can be given by a two keywords, with specific offsets. A keyword represent one edge of the element's box or the medium line between two edges: left, right, top, bottom or center (which represents either the center between the left and right edges, or the center between the top or bottom edges, depending on the context). An offset can be either a relative value, expressed as a percentage, or an absolute length value. Positive values are offset towards the right or towards the bottom, whichever is suitable. Negative values are offset in the other.



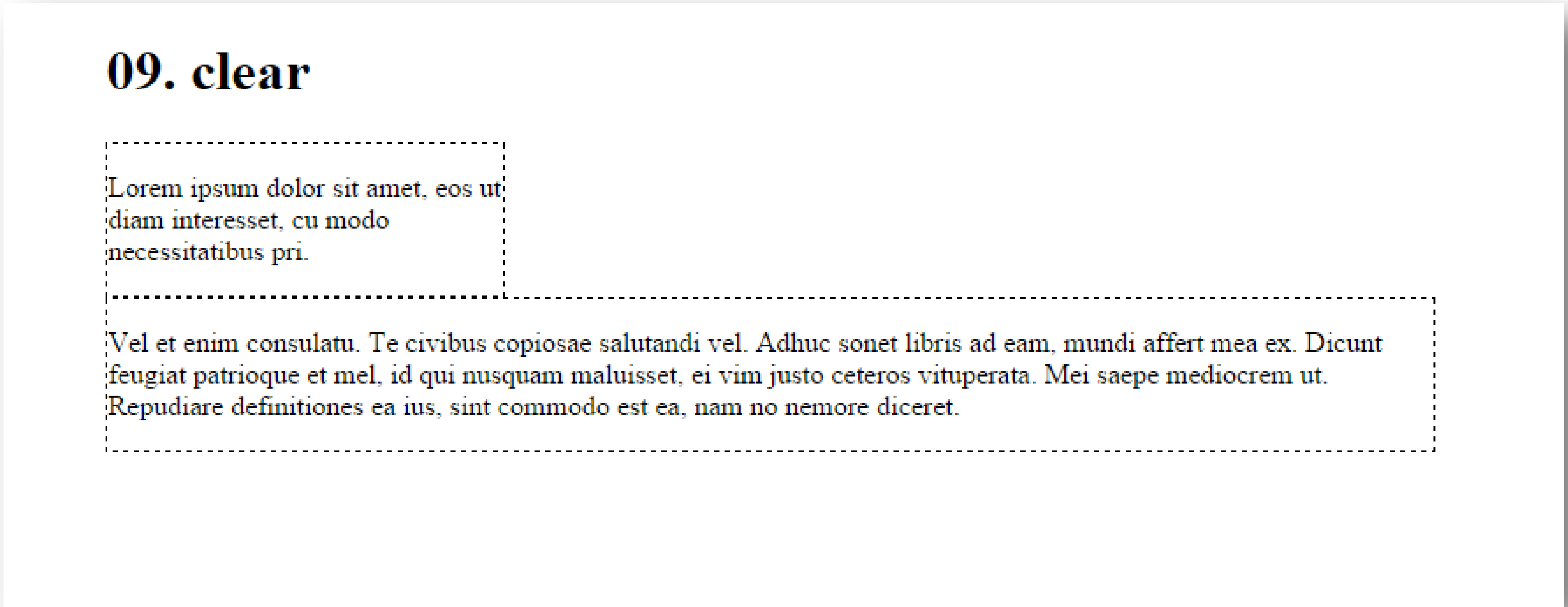
clear

- the clear property is important for controlling the behaviour of floats.

You use the value left to clear elements floated to the left. You can also clear right and both.



```css
.box {
  float: left;
  width: 200px;
  height: 100px;
  margin: 1em;
}
.after-box {
  clear: left;
}
```



09. clear

Lorem ipsum dolor sit amet, eos ut diam interesset, cu modo necessitatibus pri.

Vel et enim consulatu. Te civibus copiosae salutandi vel. Adhuc sonet libris ad eam, mundi affert mea ex. Dicunt feugiat patrioque et mel, id qui nusquam maluisset, ei vim justo ceteros vituperata. Mei saepe mediocrem ut. Repudiare definitiones ea ius, sint commodo est ea, nam no nemore diceret.

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

eLearning support unit