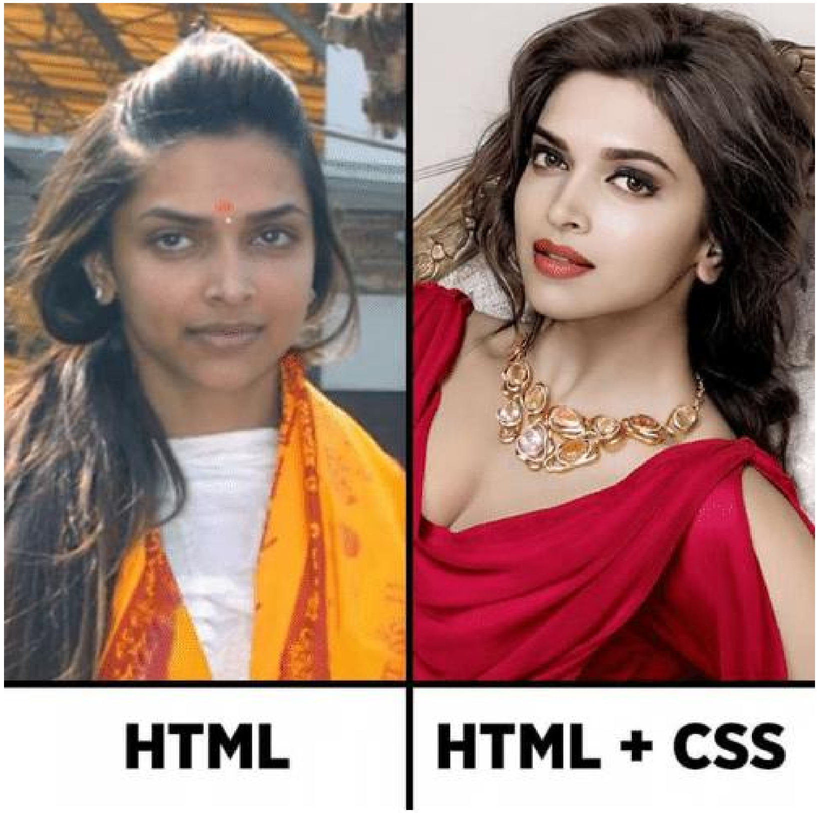
Web Programming
#3 CSS

Herman Kabetta

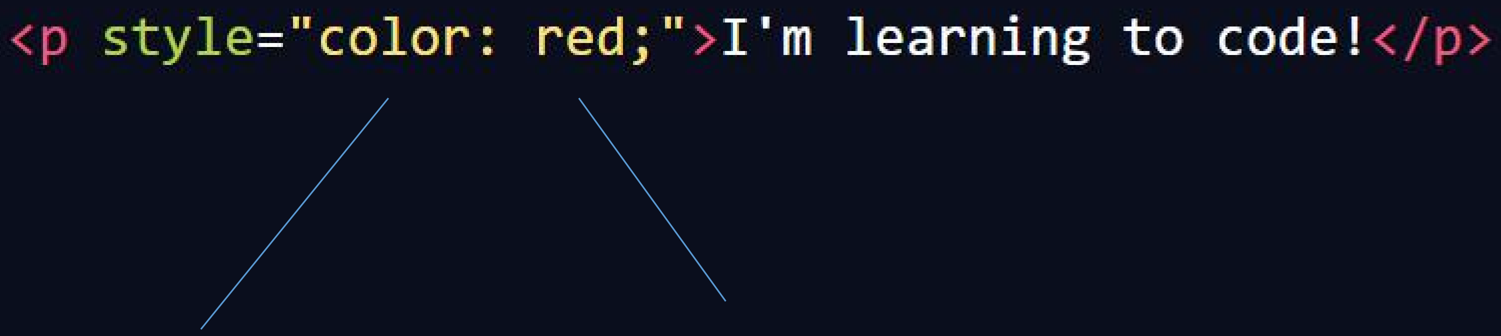
hermanka.github.io

What is CSS?

- **Cascading Style Sheets (CSS)** is a style sheet language used for describing the presentation of a document written in a markup language like HTML
- **CSS** is designed to enable the separation of presentation and content, including layout, colors, and fonts.



Inline Styles



```html
<p style="color: red;">I'm learning to code!</p>
```

property name

value

Inline Styles - Exercise

Download file

https://**github.com/hermanka/wplab**/blob/master/css/01.html

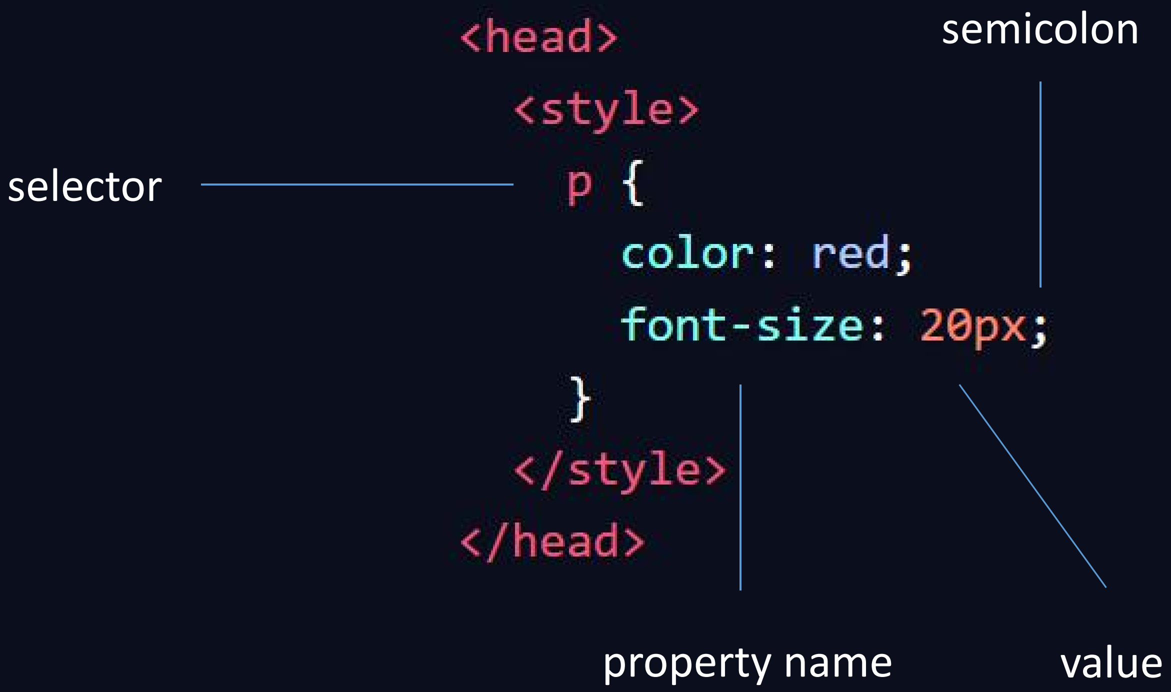
Pada paragraf pertama, atur style font-family ke Arial

More properties :

www.w3schools.com/css

The <style> Tag

- Inline styles are a fast way of styling HTML, but they also have limitations.



```
<head>
  <style>
    p {
      color: red;
      font-size: 20px;
    }
  </style>
</head>
```

The <style> Tag - Exercise

1. Tambahkan tag <style> di dalam tag <head> file 01.html. Pastikan inline style pada langkah sebelumnya telah dihapus.
2. Pindahkan properti pada inline style yang telah dihapus sebelumnya ke dalam tag <style>.

The .css file

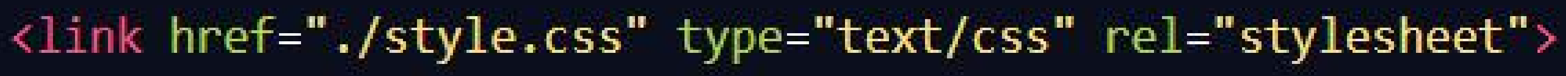
- Developers avoid mixing code by storing HTML and CSS code in separate files (HTML files contain only HTML code, and CSS files contain only CSS code).
- You can create a CSS file by using the .css file name extension, like so: **style.css**
- With a CSS file, you can write all the CSS code needed to style a page without sacrificing the readability and maintainability of your HTML file.

The .css file - Exercise

- Buatlah file baru bernama **style.css** di folder yang sama dengan file 01.html.
- Pada file 01.html, pindahkan kode CSS yang berada di dalam tag <style> kedalam file baru bernama **style.css.**
- Pastikan pula kode tag <style> telah dihapus.

Linking the CSS File

- When HTML and CSS code are in separate files, the files must be linked. Otherwise, the HTML file won’t be able to locate the CSS code, and the styling will not be applied.



```
<link href="./style.css" type="text/css" rel="stylesheet">
```

Linking the CSS File - Exercise

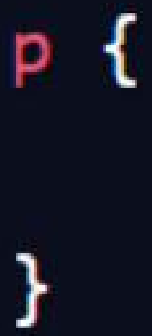
Membuat link menuju file style.css di dalam file 01.html

- Buatlah tag <link> di dalam tag <head>
- Tambahkan atribut href pada tag <link> dan isi dengan lokasi file style.css
- Tambahkan atribut type, isi dengan “text/css”.
- Terakhir, tambahkan atribut rel, isi dengan string “stylesheet”

⚠ **Perhatikan perubahan pada file 01.html, jika tidak ada kesalahan, maka paragraf pertama pada file 01.html seharusnya ber-font Arial.**

Tag Name Selector

- CSS can select HTML elements by using an element’s tag name. A tag name is the word (or character) between HTML angle brackets.
- For example, in HTML, the tag for a paragraph element is <p>. The CSS syntax for selecting <p> elements is:



```
p {

}
```

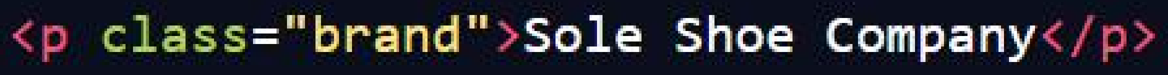
Tag Name Selector - Exercise

- Pada file **style.css**, tambahkan selector untuk tag <h1>.
- Di dalam kurung kurawal selector h1, tambahkan properti berikut :

color: maroon;

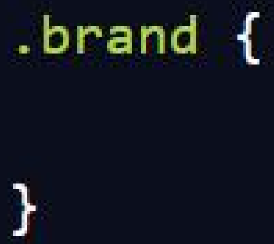
Class Name Selector

- CSS is not limited to selecting elements by tag name.
- HTML elements can have more than just a tag name, they can also have attributes.
- One common attribute is the class attribute. It’s also possible to select an element by its class attribute.
- Example class attribute in HTML FIle



```
<p class="brand">Sole Shoe Company</p>
```

- Put its selector in CSS File



```
.brand {

}
```

Class Name Selector - Exercise

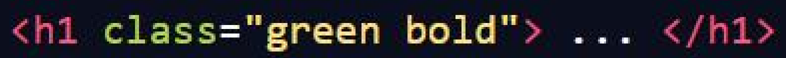
- Pada file **style.css**, buatlah selector untuk class title.
- tambahkan properti berikut :

  color: teal;

Multiple Classes

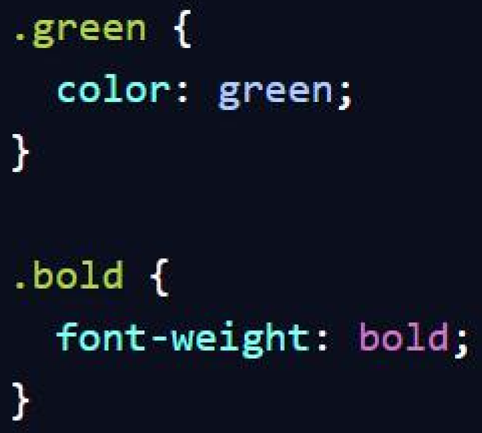
- It’s possible to add more than one class name to an HTML element’s class attribute.

HTML File



```
<h1 class="green bold"> ... </h1>
```

CSS File



```
.green {
  color: green;
}

.bold {
  font-weight: bold;
}
```

Multiple Classes - Exercise

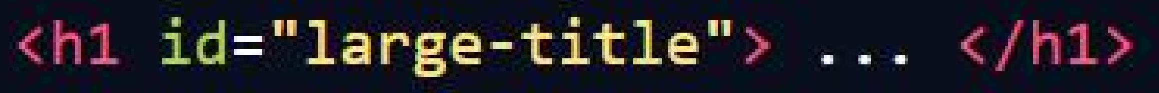
- Pada file **style.css**, tambahkan selector class “.uppercase”.
- Berikut properti didalamnya :

  text-transform: uppercase;

- Pada file 01.html, tambahkan class baru pada tag h1 bersebelahan dengan class title (pisahkan dengan spasi).

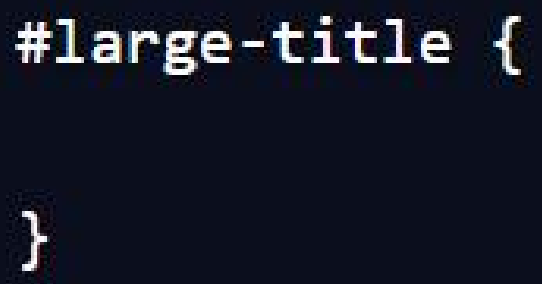
ID Name Selector

- If an HTML element needs to be styled uniquely (no matter what classes are applied to the element), we can add an ID to the element.
- IDs override the styles of tags and classes.
- ID attribute in HTML File



```
<h1 id="large-title"> ... </h1>
```

- Put its selector in CSS File



```
#large-title {

}
```

ID Name Selector - Exercise

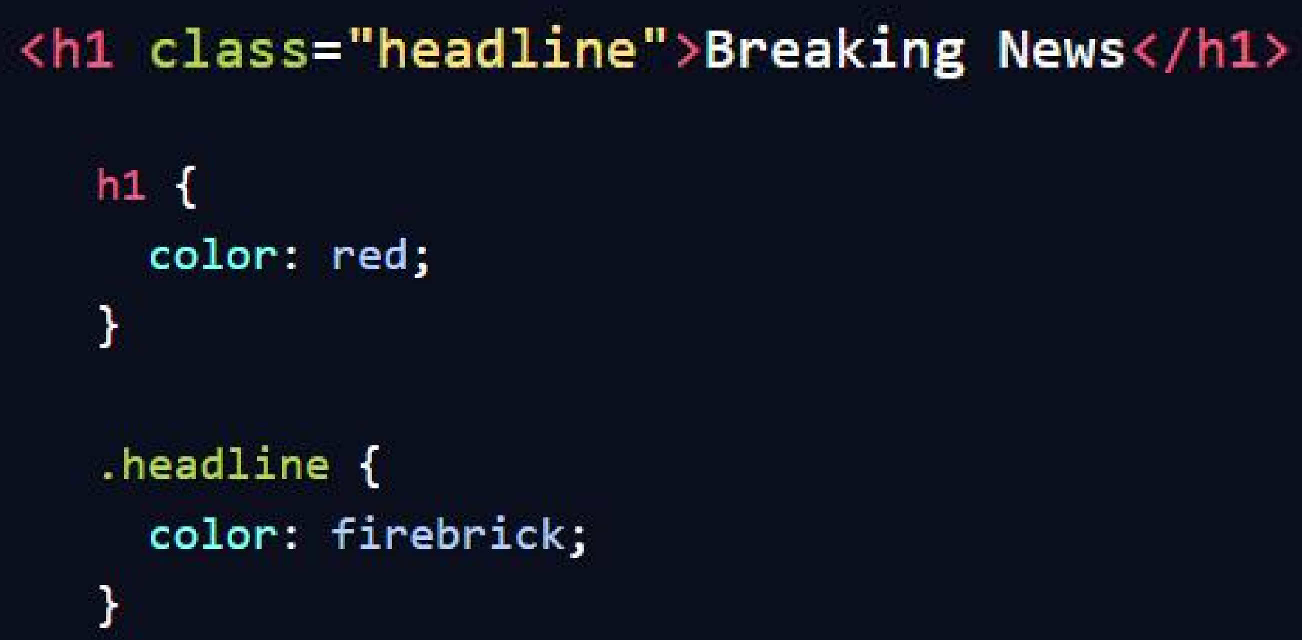
- Pada file **style.css**, tambahkan selector ID dengan nama "**article-title**".
- Dengan properti sbb :

  font-family: cursive;

  text-transform: capitalize;
- Pada file 01.html, tambahkan atribut id pada tag h1 dengan nilai "article-title"

Specificity

- Specificity is the order by which the browser decides which CSS styles will be displayed.
- IDs are the most specific selector in CSS, followed by classes, and finally, tags.



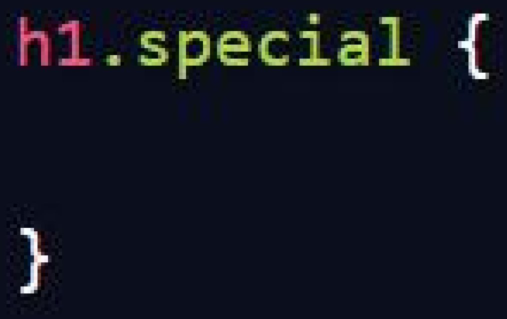
```
<h1 class="headline">Breaking News</h1>

    h1 {
      color: red;
    }

    .headline {
      color: firebrick;
    }
```

Chaining Selectors

- When writing CSS rules, it’s possible to require an HTML element to have two or more CSS selectors at the same time.



```
h1.special {


}
```

- There was a .special class for <h1> elements
- If a <p> element also had a class of .special, the rule in the example would not style the paragraph.

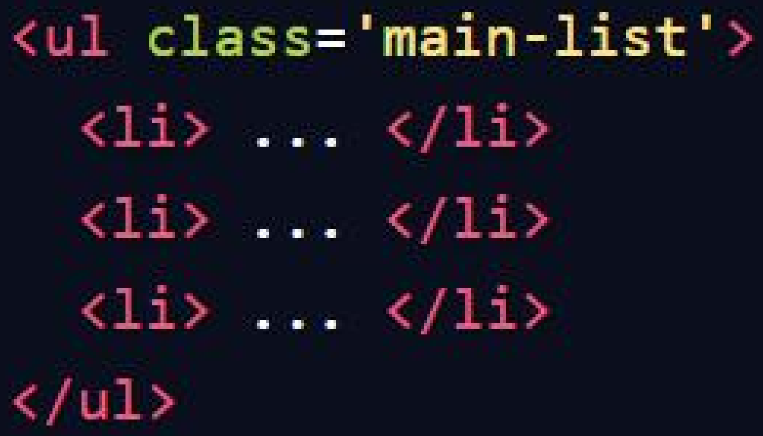
Chaining Selectors - Exercise

- Pada file **style.css**, tuliskan CSS selector untuk tag **h2** dengan class **destination**
- Dengan properti sbb :

  font-family: cursive;

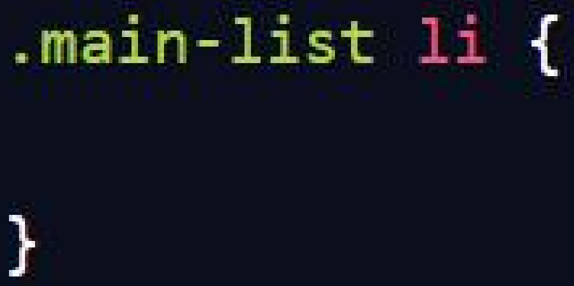
Nested Elements

- In addition to chaining selectors to select elements, CSS also supports selecting elements that are nested within other HTML elements.



```
<ul class='main-list'>
  <li> ... </li>
  <li> ... </li>
  <li> ... </li>
</ul>
```

separated by a space



```
.main-list li {

}
```

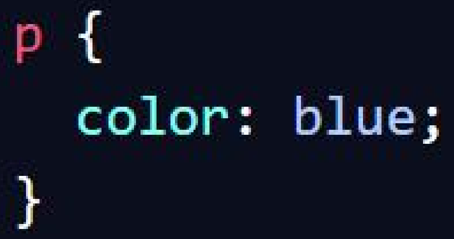
Nested Elements - Exercise

- Pada file **01.html**, perhatikan tag <h5> yang berisi teks “Top Attractions”.
- Kita akan merubah warna teks tersebut dengan merujuk pada parent tag-nya.
- Jika diperhatikan, tag **<h5>** tersebut berada di dalam element dengan class **description.**
- Buka file **style.css**, kemudian tambahkan selector yang mengarah pada tag **<h5>** yang berada di dalam element dengan class **description**.
- Tambahkan properti sbb :

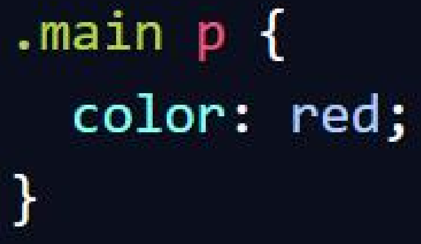
  color: teal;

Chaining and Specificity

- Adding more than one tag, class, or ID to a CSS selector increases the specificity of the CSS selector.



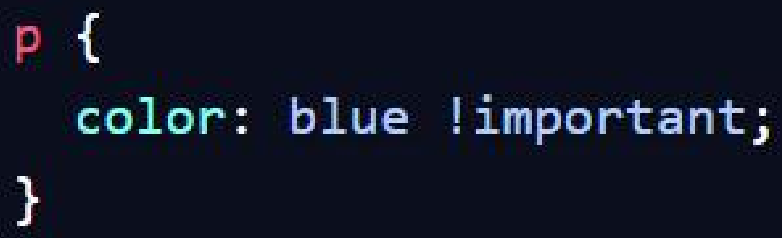
```
p {
  color: blue;
}
```



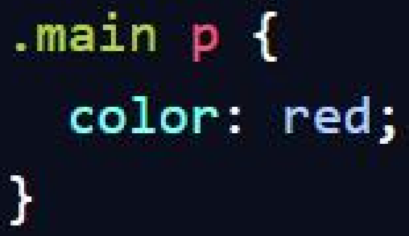
```
.main p {
  color: red;
}
```

Important

- Frustating design?!
- There is one thing that is even more specific than IDs: **!important**



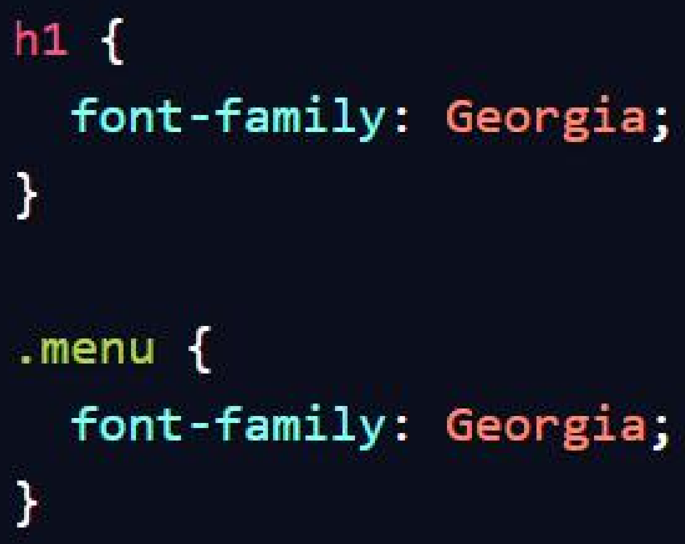
```css
p {
  color: blue !important;
}
```



```css
.main p {
  color: red;
}
```

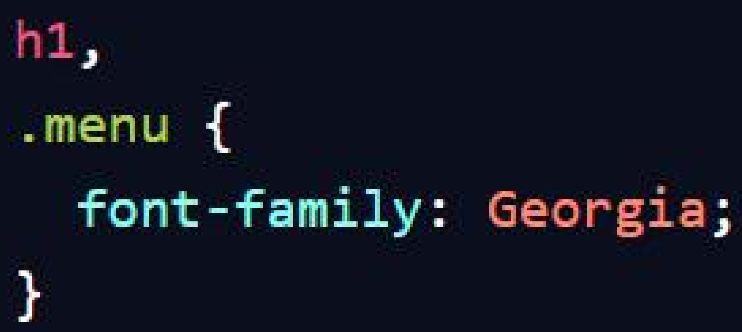
Multiple Selectors

- In order to make CSS more concise, it’s possible to add CSS styles to multiple CSS selectors all at once.
- This prevents writing repetitive code.



```css
h1 {
  font-family: Georgia;
}

.menu {
  font-family: Georgia;
}
```

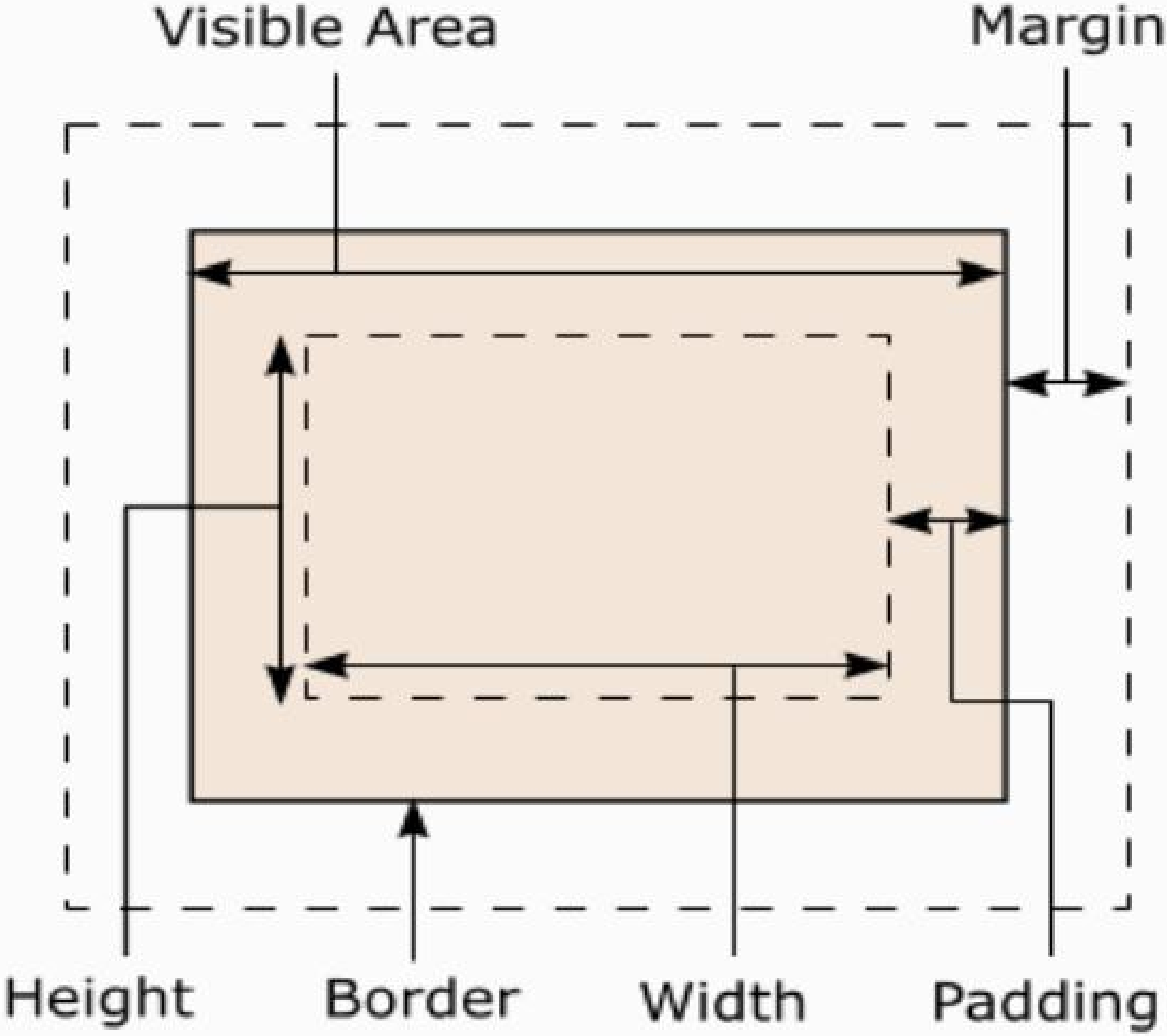


```css
h1,
.menu {
  font-family: Georgia;
}
```

Komponen CSS | Properti

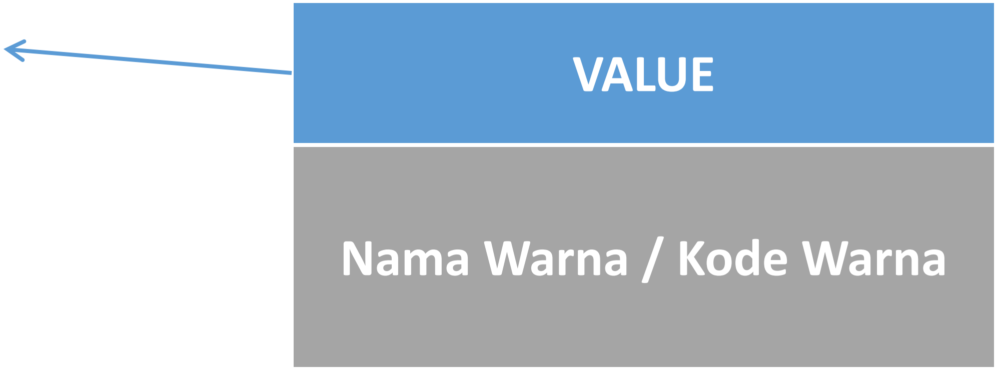
Ada beberapa tipe yang berbeda dari nama properti yang dapat dikelompokkan menjadi beberapa jenis, diantaranya :

- Text style
- Text layout
- Background
- Border
- Margin
- Padding
- Page layout
- Element type
- User interface


Visible Area
Margin
Height
Border
Width
Padding

Properti Text Style

- color
- font-weight
- font-family
- font-size
- font-style
- text-decoration
- text-transform



VALUE

Nama Warna / Kode Warna

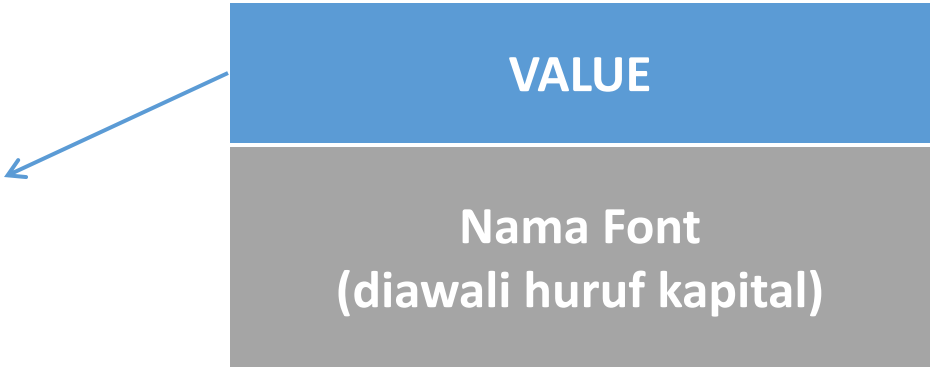
Properti Text Style

- color
- font-weight
- font-family
- font-size
- font-style
- text-decoration
- text-transform

VALUE
Normal, Bold, Bolder, Lighter, Angka [100 … 900]

Properti Text Style

- color
- font-weight
- font-family
- font-size
- font-style
- text-decoration
- text-transform

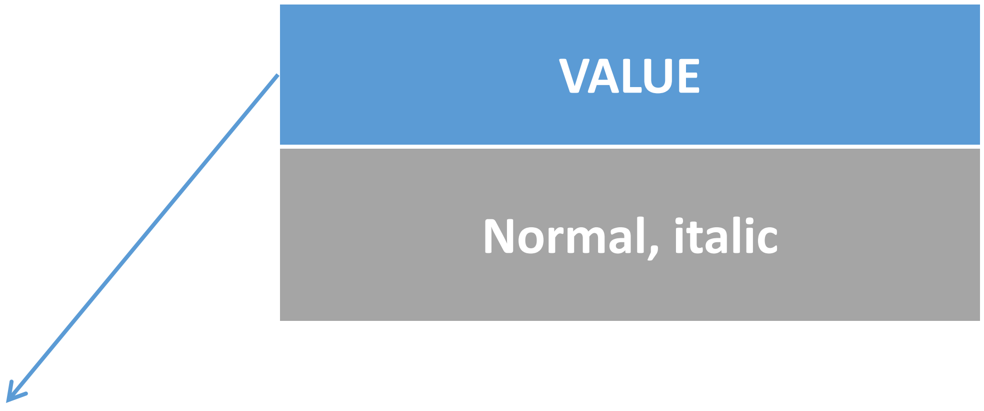


VALUE

Nama Font
(diawali huruf kapital)

Properti Text Style

- color
- font-weight
- font-family
- font-size
- font-style
- text-decoration
- text-transform



VALUE

Normal, italic

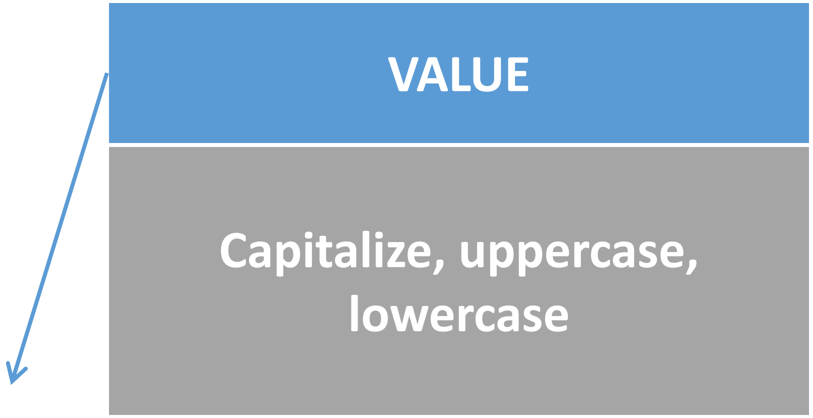
Properti Text Style

- color
- font-weight
- font-family
- font-size
- font-style
- text-decoration
- text-transform

VALUE
none, underline, overline, line though, blink

Properti Text Style

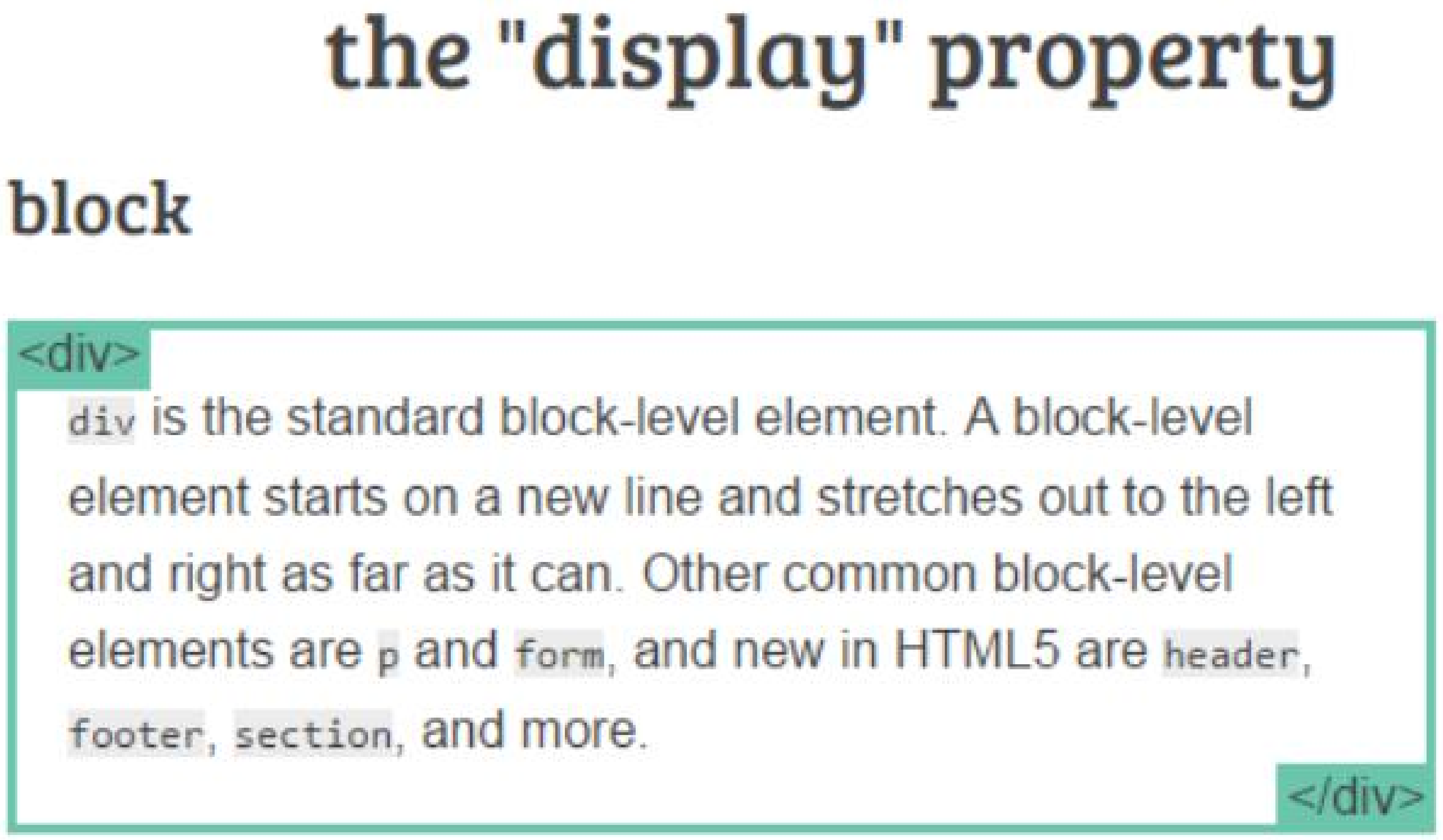
- color
- font-weight
- font-family
- font-size
- font-style
- text-decoration
- text-transform



VALUE

Capitalize, uppercase, lowercase

Learn CSS Layout

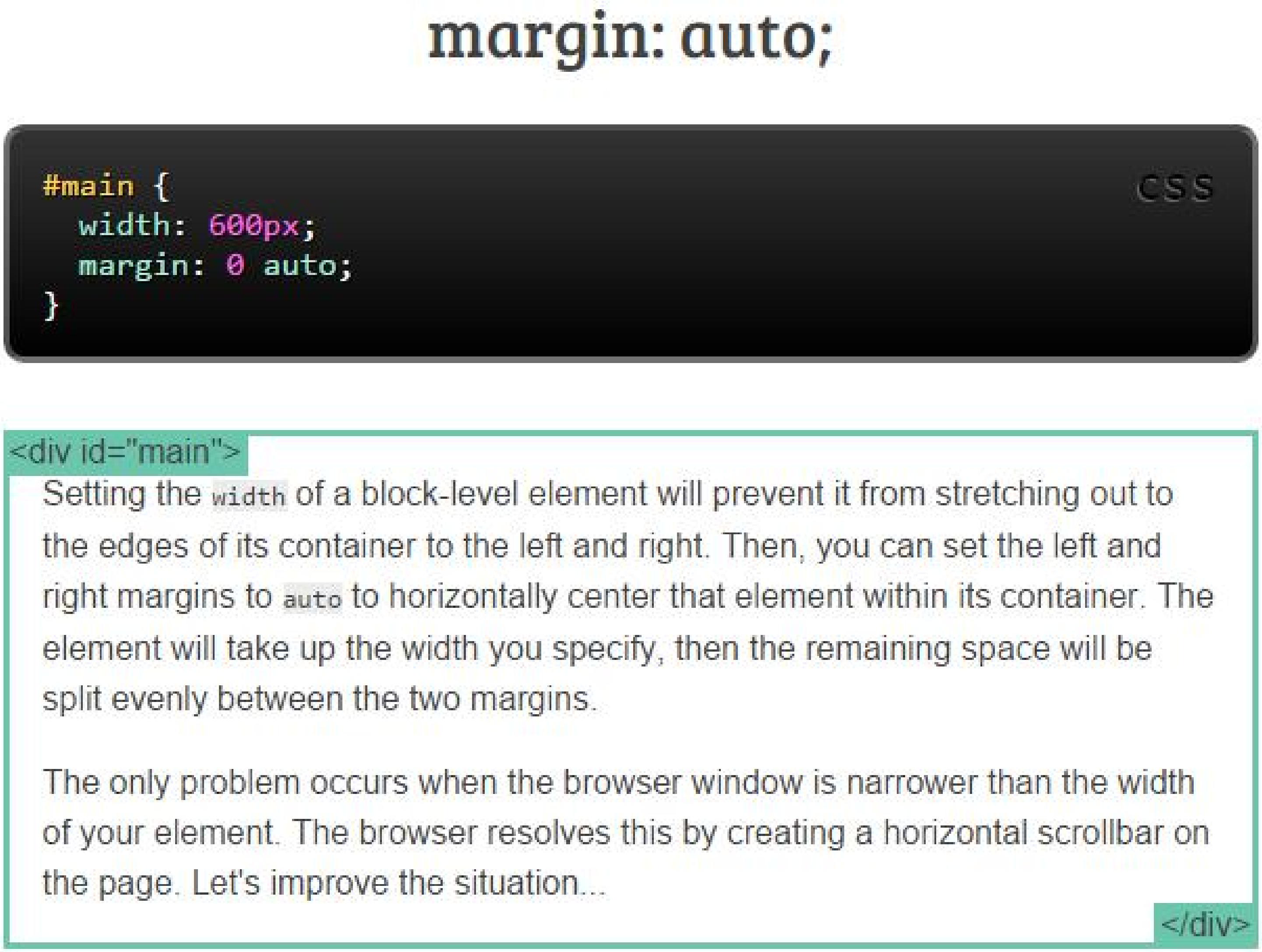


the "display" property

block

`div` is the standard block-level element. A block-level element starts on a new line and stretches out to the left and right as far as it can. Other common block-level elements are `p` and `form`, and new in HTML5 are `header`, `footer`, `section`, and more.

Learn CSS Layout



margin: auto;

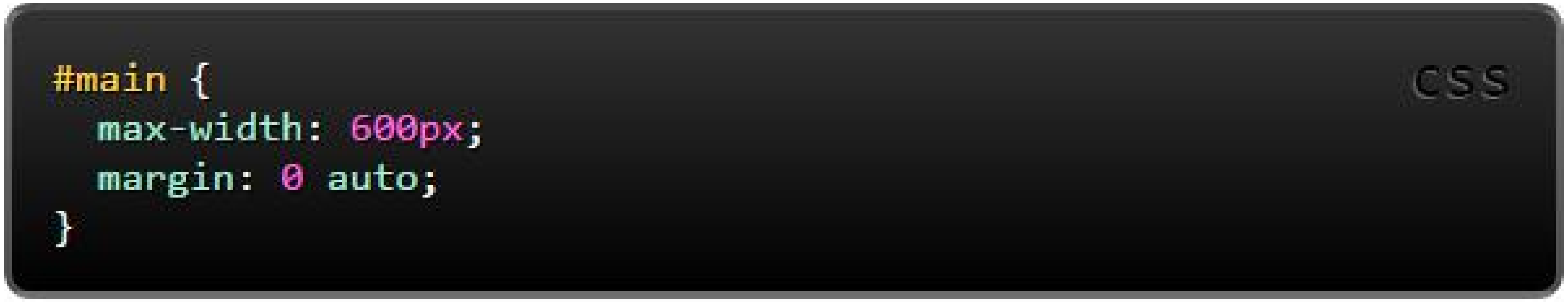
```css
#main {
  width: 600px;
  margin: 0 auto;
}
```

Setting the `width` of a block-level element will prevent it from stretching out to the edges of its container to the left and right. Then, you can set the left and right margins to `auto` to horizontally center that element within its container. The element will take up the width you specify, then the remaining space will be split evenly between the two margins.

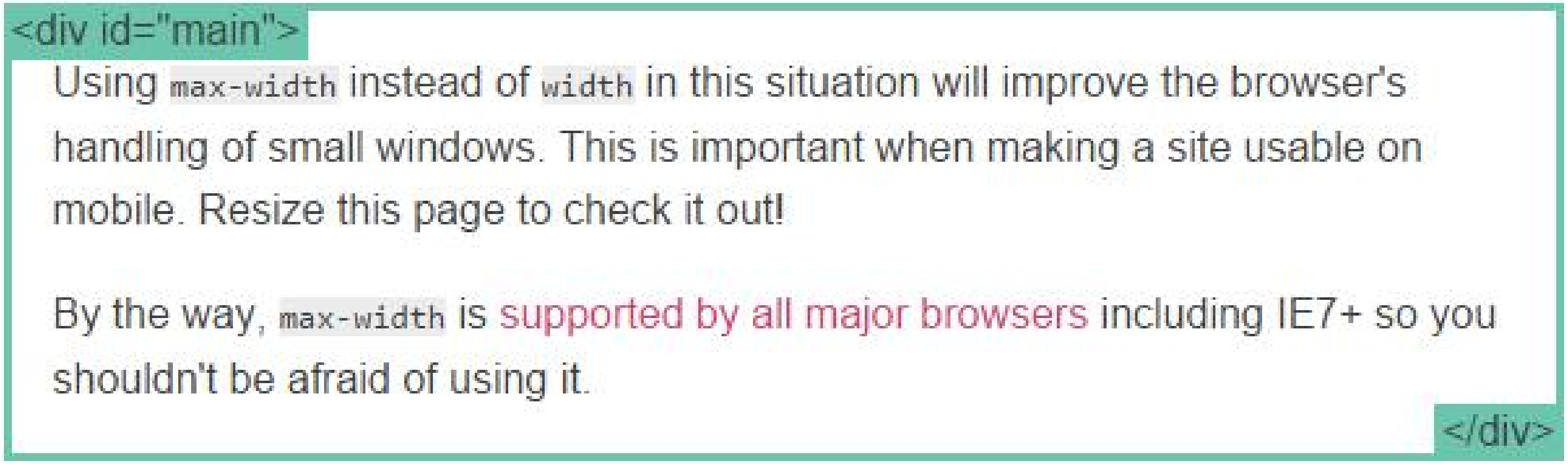
The only problem occurs when the browser window is narrower than the width of your element. The browser resolves this by creating a horizontal scrollbar on the page. Let's improve the situation...

Learn CSS Layout

max-width



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



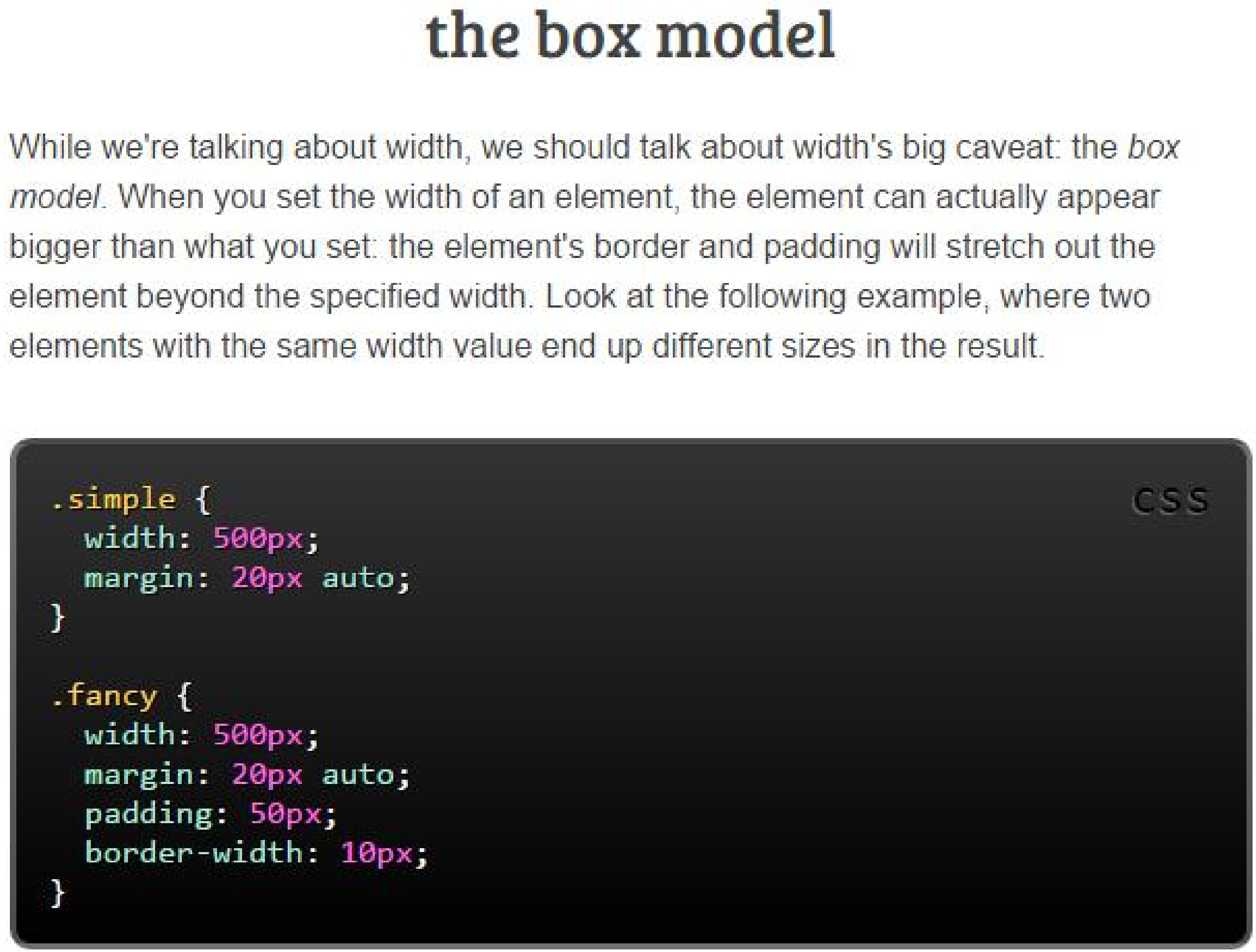
<div id="main">

Using `max-width` instead of `width` in this situation will improve the browser's handling of small windows. This is important when making a site usable on mobile. Resize this page to check it out!

By the way, `max-width` is supported by all major browsers including IE7+ so you shouldn't be afraid of using it.

</div>

Learn CSS Layout



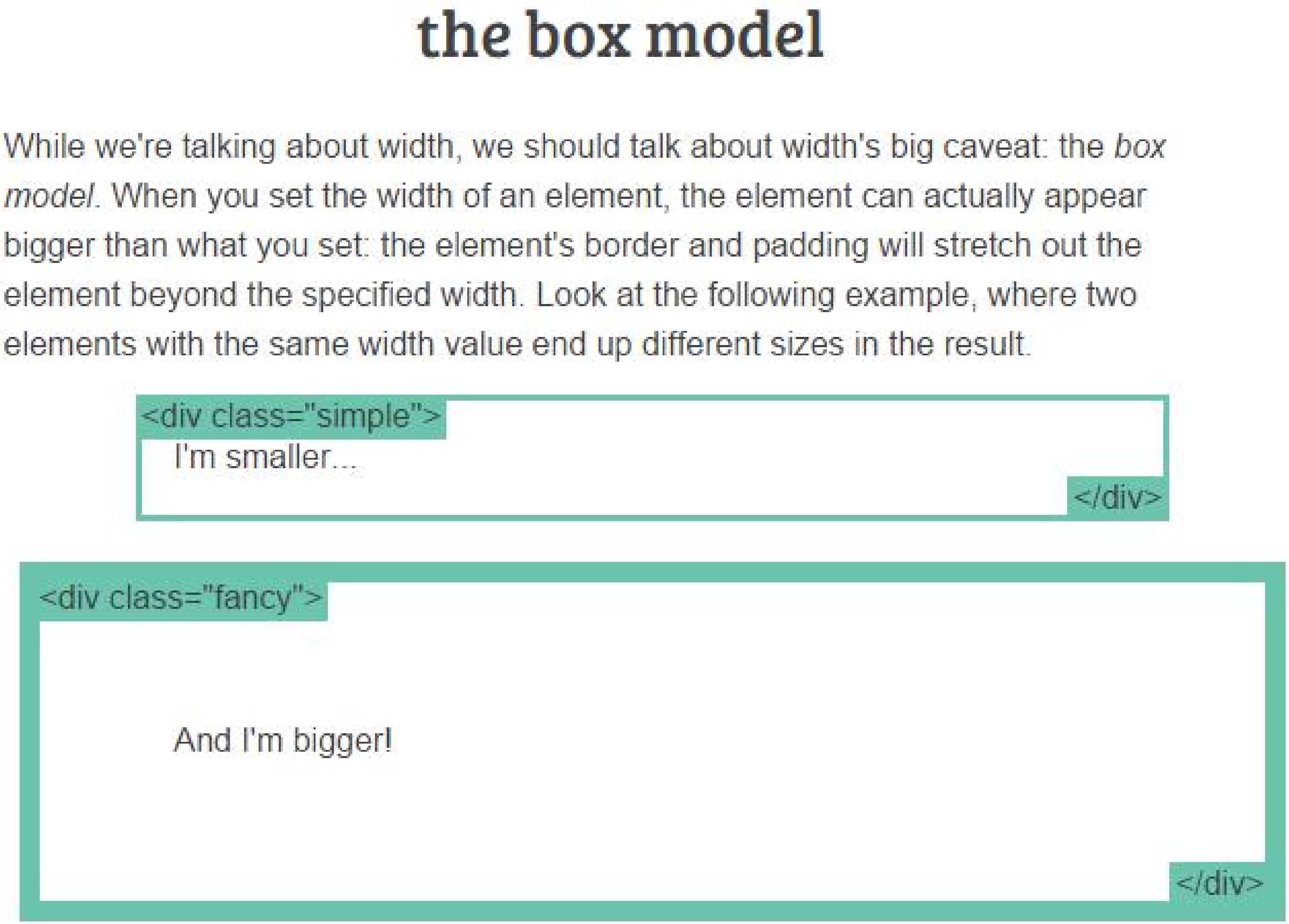
the box model

While we're talking about width, we should talk about width's big caveat: the *box model*. When you set the width of an element, the element can actually appear bigger than what you set: the element's border and padding will stretch out the element beyond the specified width. Look at the following example, where two elements with the same width value end up different sizes in the result.

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

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

Learn CSS Layout



the box model

While we're talking about width, we should talk about width's big caveat: the *box model*. When you set the width of an element, the element can actually appear bigger than what you set: the element's border and padding will stretch out the element beyond the specified width. Look at the following example, where two elements with the same width value end up different sizes in the result.

<div class="simple">

I'm smaller...

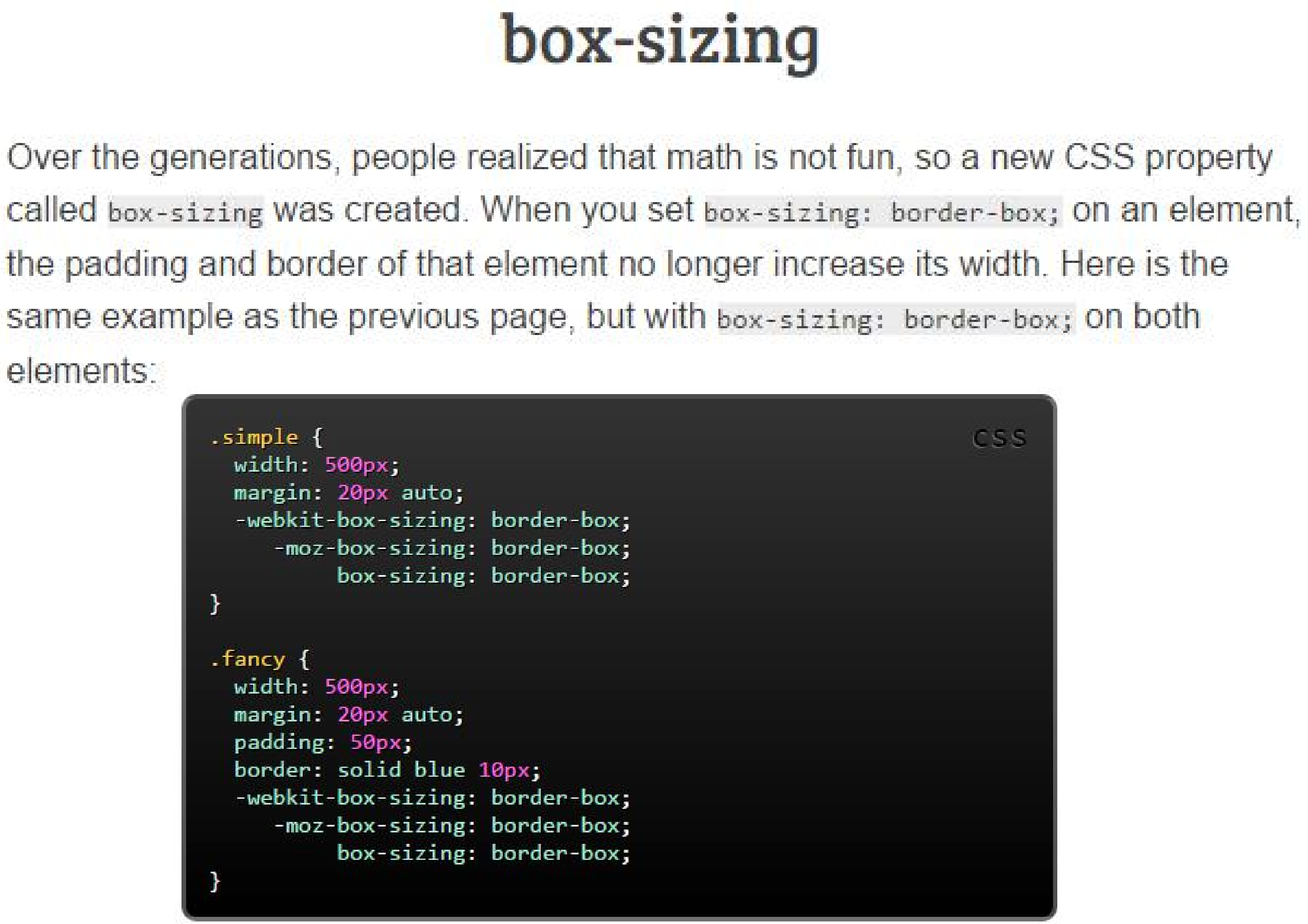
</div>

<div class="fancy">

And I'm bigger!

</div>

Learn CSS Layout



box-sizing

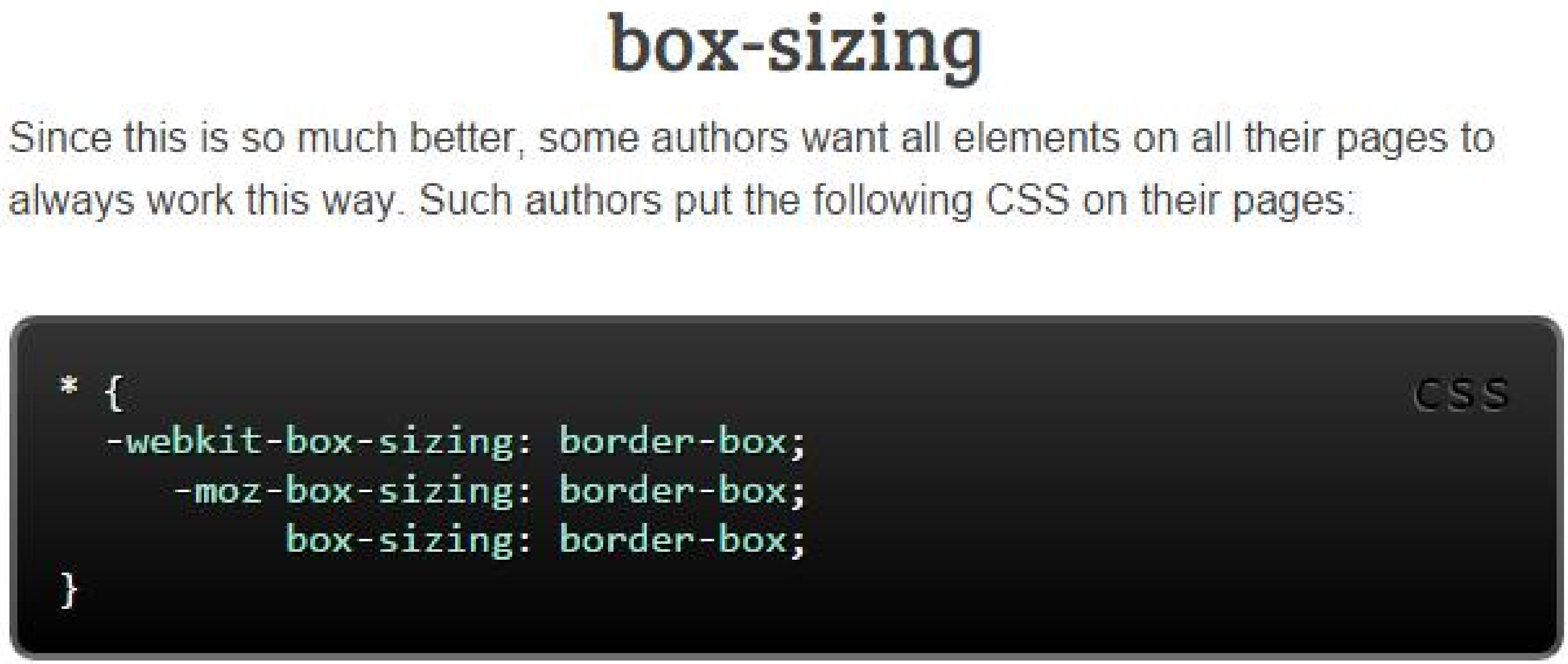
Over the generations, people realized that math is not fun, so a new CSS property called `box-sizing` was created. When you set `box-sizing: border-box;` on an element, the padding and border of that element no longer increase its width. Here is the same example as the previous page, but with `box-sizing: border-box;` on both elements:

CSS

```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;
}
```

Learn CSS Layout

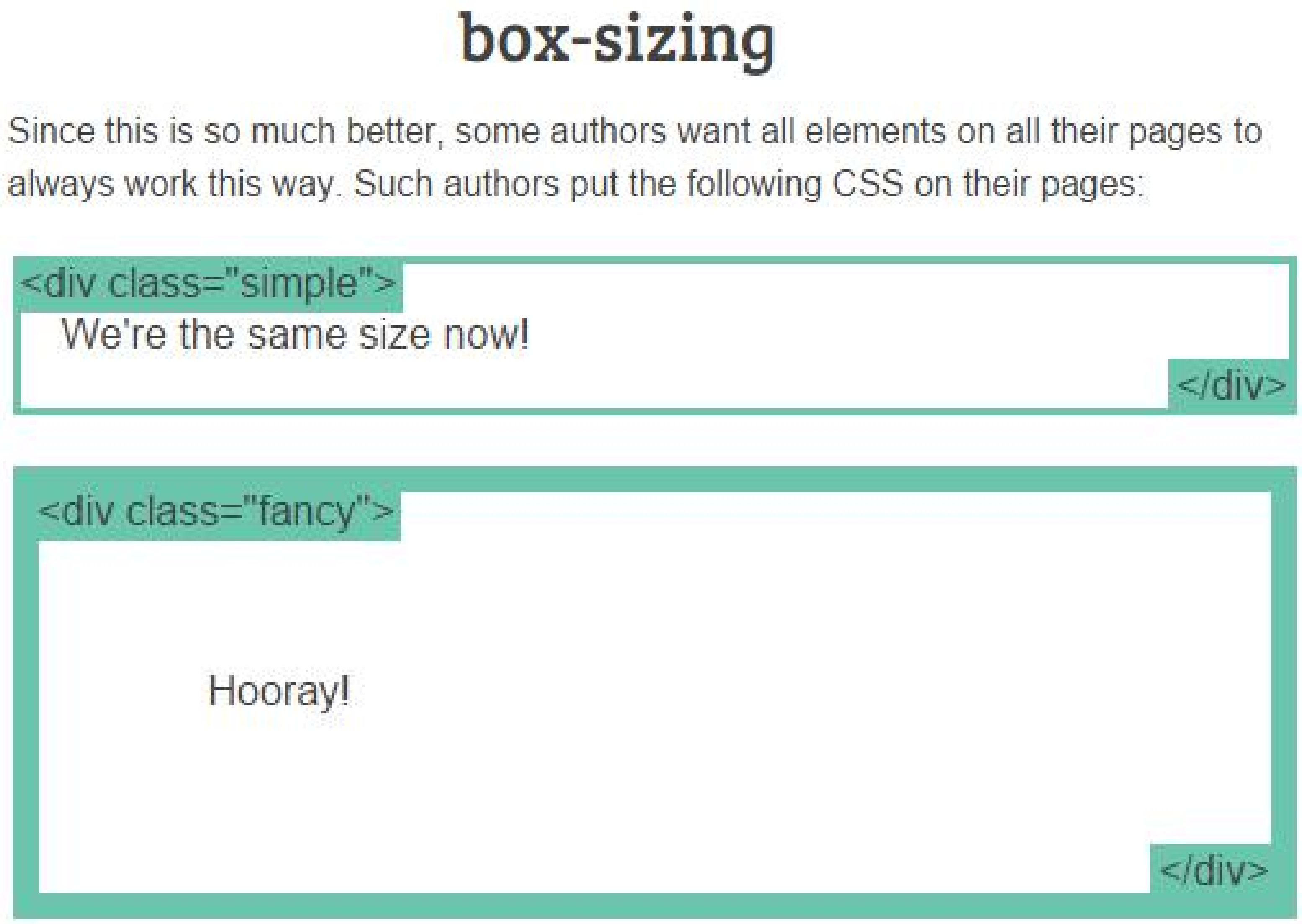


box-sizing

Since this is so much better, some authors want all elements on all their pages to always work this way. Such authors put the following CSS on their pages:

CSS

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



box-sizing

Since this is so much better, some authors want all elements on all their pages to always work this way. Such authors put the following CSS on their pages:

<div class="simple">

We're the same size now!

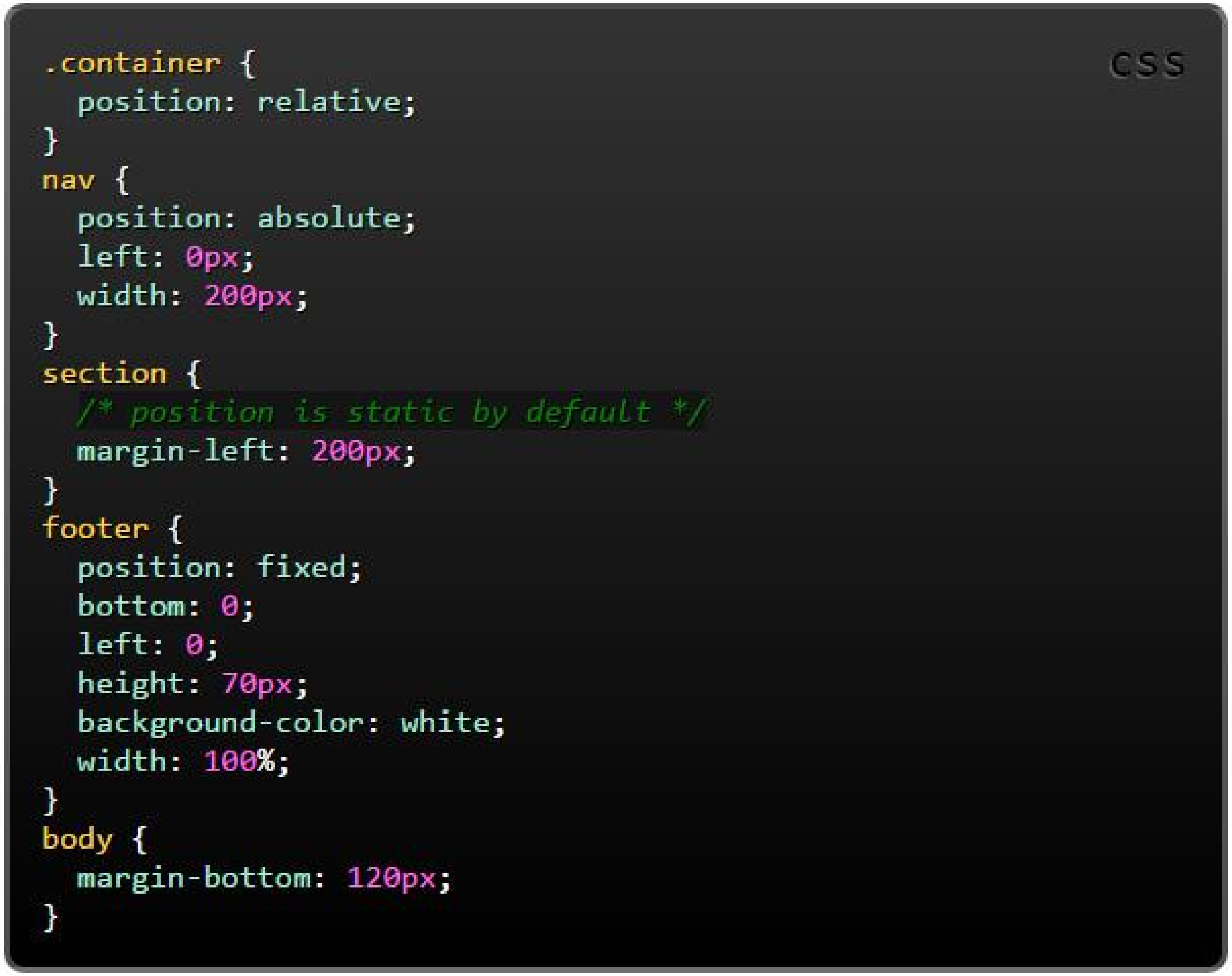
</div>

<div class="fancy">

Hooray!

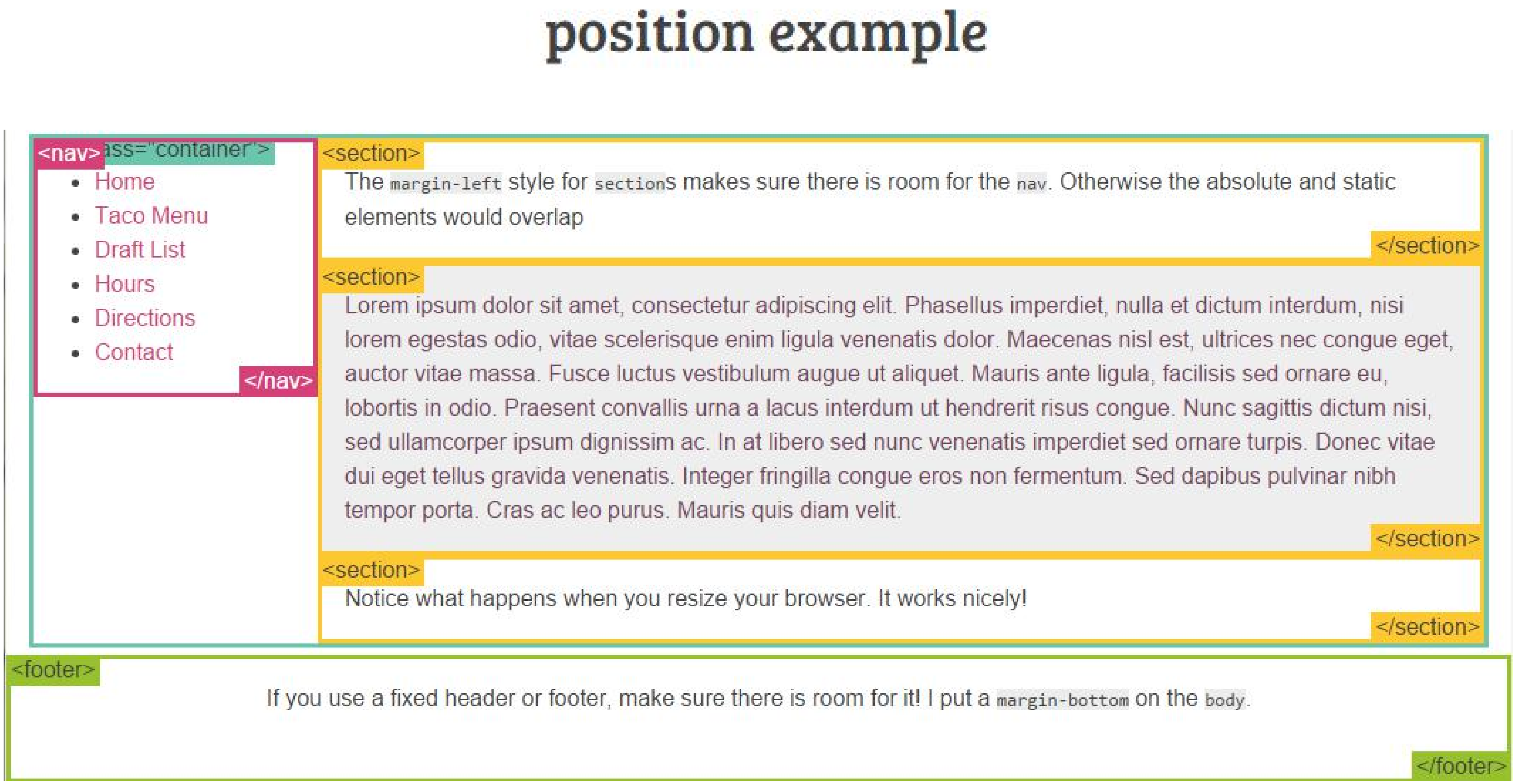
</div>

position example



```css
.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;
}
```

Learn CSS Layout



position example

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

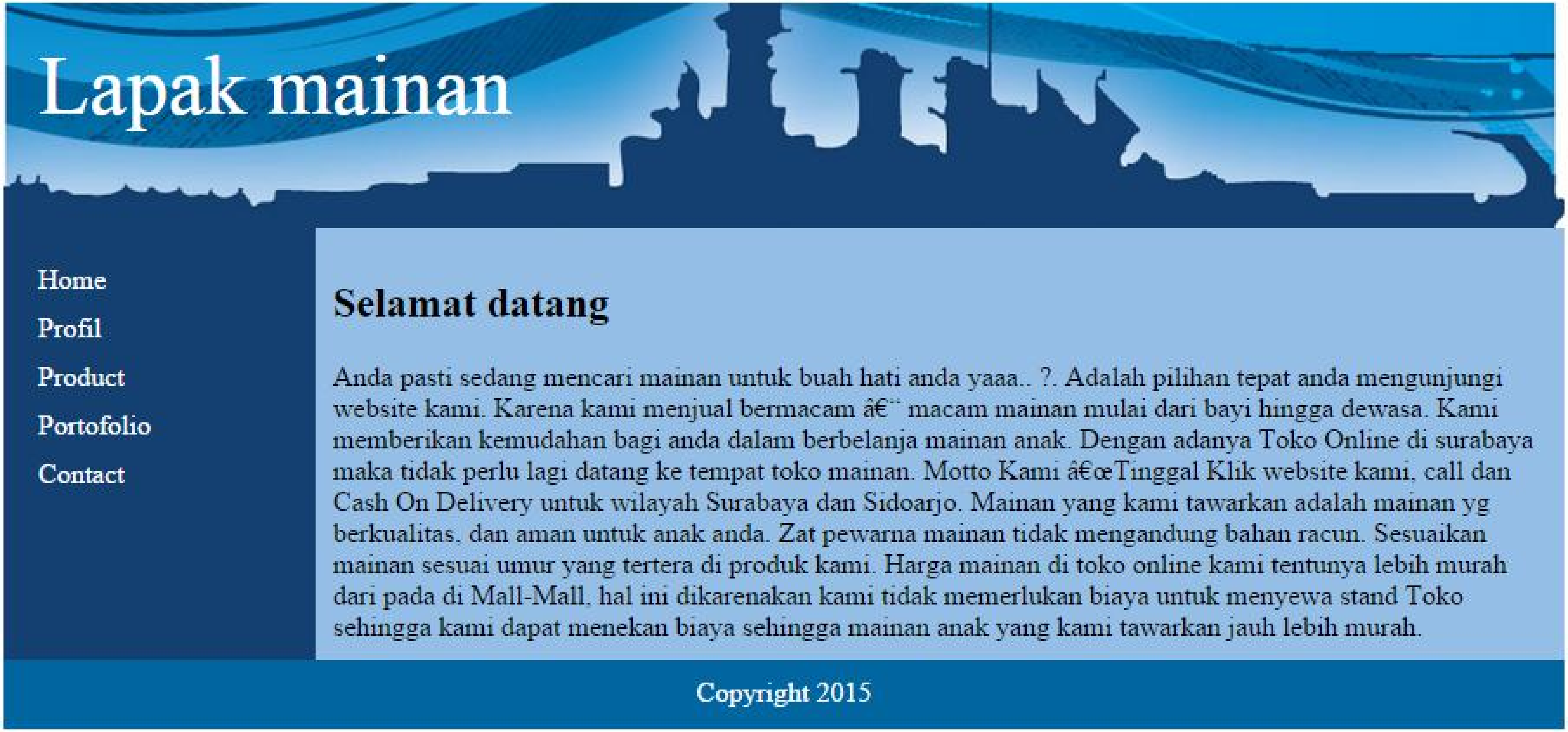
The `margin-left` style for `section`s makes sure there is room for the `nav`. Otherwise the absolute and static elements would overlap

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`.

Latihan



Lapak mainan

- Home
- Profil
- Product
- Portofolio
- Contact

Selamat datang

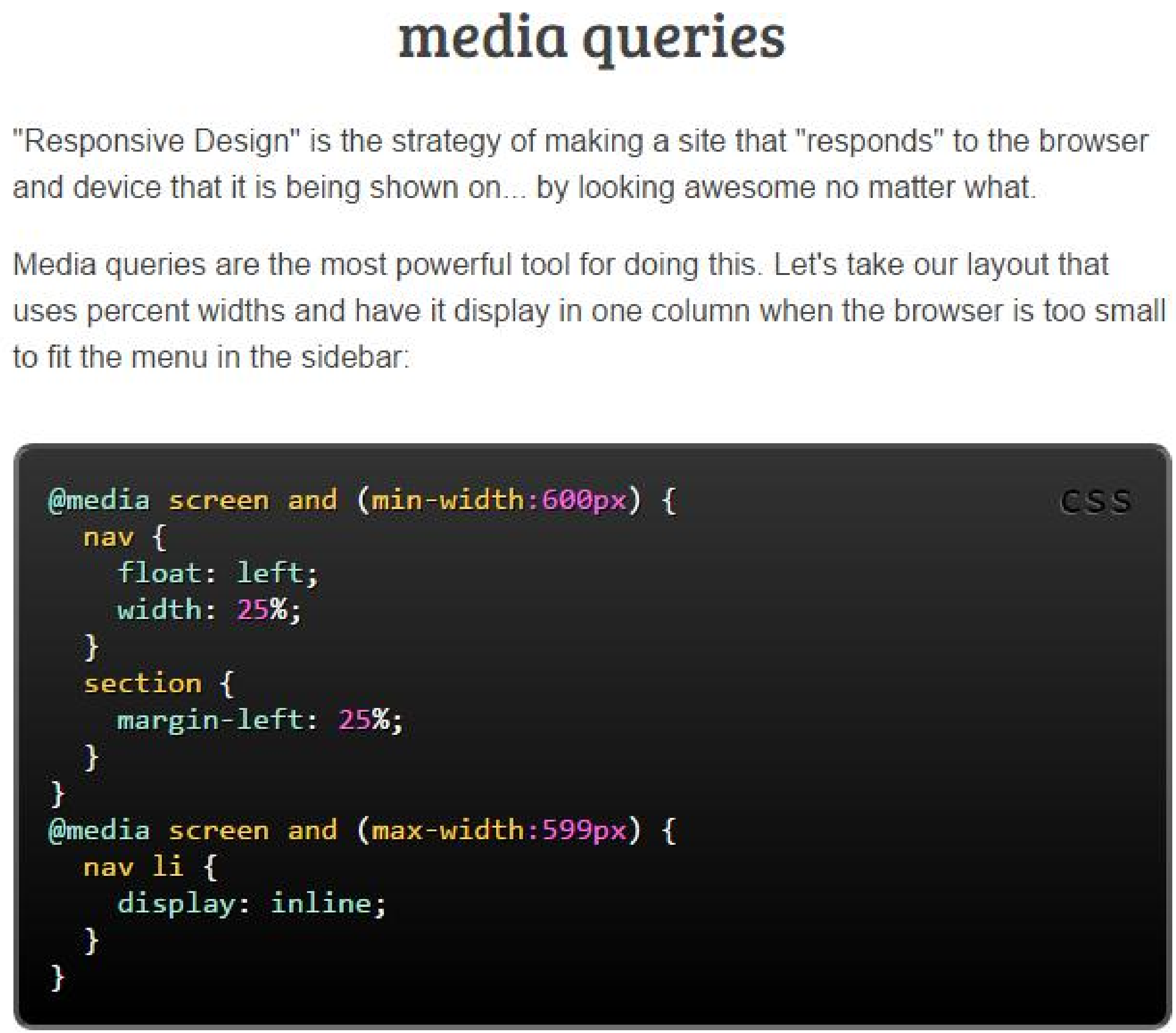
Anda pasti sedang mencari mainan untuk buah hati anda yaaa.. ?. Adalah pilihan tepat anda mengunjungi website kami. Karena kami menjual bermacam â€" macam mainan mulai dari bayi hingga dewasa. Kami memberikan kemudahan bagi anda dalam berbelanja mainan anak. Dengan adanya Toko Online di surabaya maka tidak perlu lagi datang ke tempat toko mainan. Motto Kami â€œTinggal Klik website kami, call dan Cash On Delivery untuk wilayah Surabaya dan Sidoarjo. Mainan yang kami tawarkan adalah mainan yg berkualitas, dan aman untuk anak anda. Zat pewarna mainan tidak mengandung bahan racun. Sesuaikan mainan sesuai umur yang tertera di produk kami. Harga mainan di toko online kami tentunya lebih murah dari pada di Mall-Mall, hal ini dikarenakan kami tidak memerlukan biaya untuk menyewa stand Toko sehingga kami dapat menekan biaya sehingga mainan anak yang kami tawarkan jauh lebih murah.

Copyright 2015

Responsive Design

Mobile
Tablet
Desktop

Responsive Design



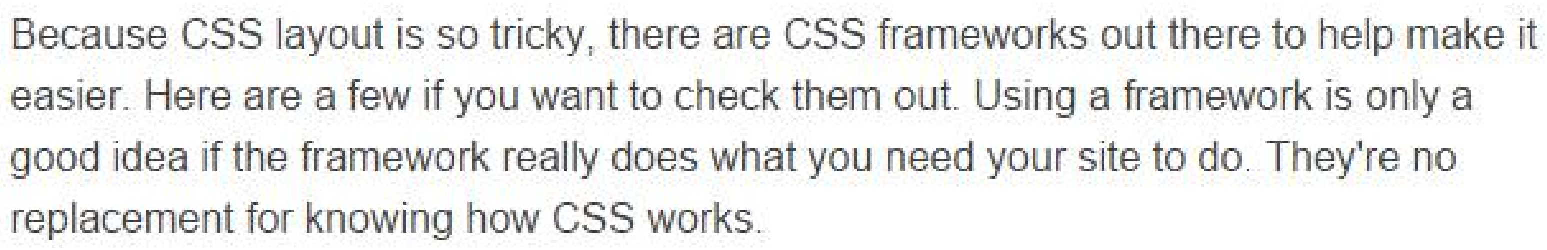
media queries

"Responsive Design" is the strategy of making a site that "responds" to the browser and device that it is being shown on... by looking awesome no matter what.

Media queries are the most powerful tool for doing this. Let's take our layout that uses percent widths and have it display in one column when the browser is too small to fit the menu in the sidebar:

```css
@media screen and (min-width:600px) {
  nav {
    float: left;
    width: 25%;
  }
  section {
    margin-left: 25%;
  }
}
@media screen and (max-width:599px) {
  nav li {
    display: inline;
  }
}
```

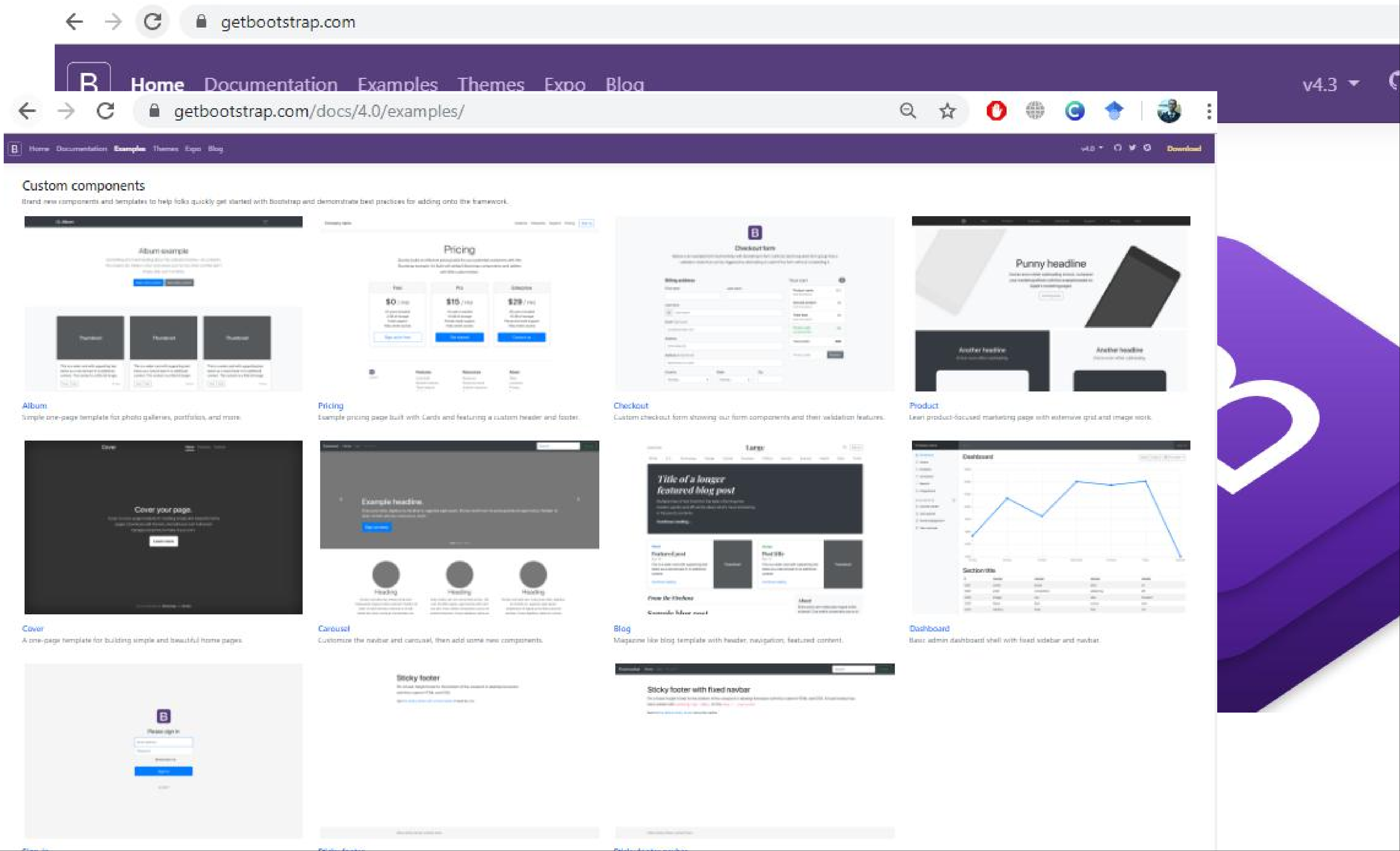
CSS Framework



Because CSS layout is so tricky, there are CSS frameworks out there to help make it easier. Here are a few if you want to check them out. Using a framework is only a good idea if the framework really does what you need your site to do. They're no replacement for knowing how CSS works.

blueprint
unsemantic
‹SUSY›
Foundation
BlueTrip
elasticss
Kube
GroundworkCSS
Bootstrap
Gumby
FRAMEWORK
Semantic UI
P
CSS
Pure.css

Bootstrap



getbootstrap.com
B Home Documentation Examples Themes Expo Blog
v4.3
getbootstrap.com/docs/4.0/examples/
B Home Documentation Examples Themes Expo Blog
v4.0
Download

Custom components

Brand new components and templates to help folks quickly get started with Bootstrap and demonstrate best practices for adding onto the framework.

Album
Simple one-page template for photo galleries, portfolios, and more.

Pricing
Example pricing page built with Cards and featuring a custom header and footer.

Checkout
Custom checkout form showing our form components and their validation features.

Product
Lean product-focused marketing page with extensive grid and image work.

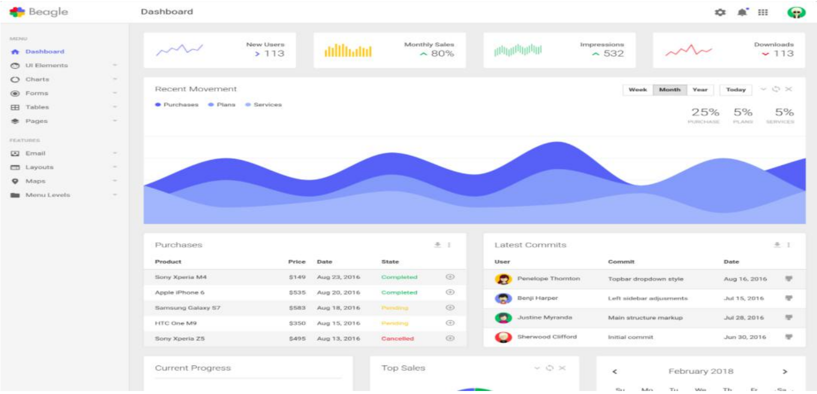
Cover
A one-page template for building simple and beautiful home pages.

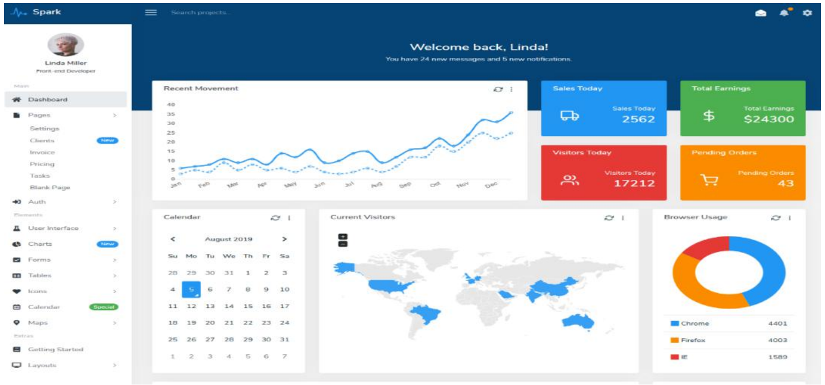
Carousel
Customize the navbar and carousel, then add some new components.

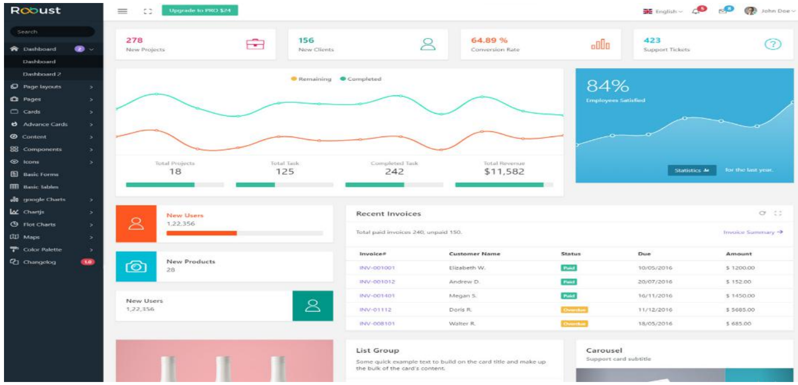
Blog
Magazine like blog template with header, navigation, featured content.

Dashboard
Basic admin dashboard shell with fixed sidebar and navbar.

Using Template







Latihan

- Pelajari dokumentasi Bootstrap.
- Kemudian buat ulang halaman web pada latihan sebelumnya menggunakan Bootstrap.