CSS[3] Fundamentals

PV219, Spring 2020

Agenda

- History
- What is New
- Basics (Syntax, Shorthand, Box Model)
- Selectors
- Examples

History

- CSS initially released in December 1996
- CSS2 which was released in 1998 became a recommendation June 7th 2011
- CSS3 was published in June 1999

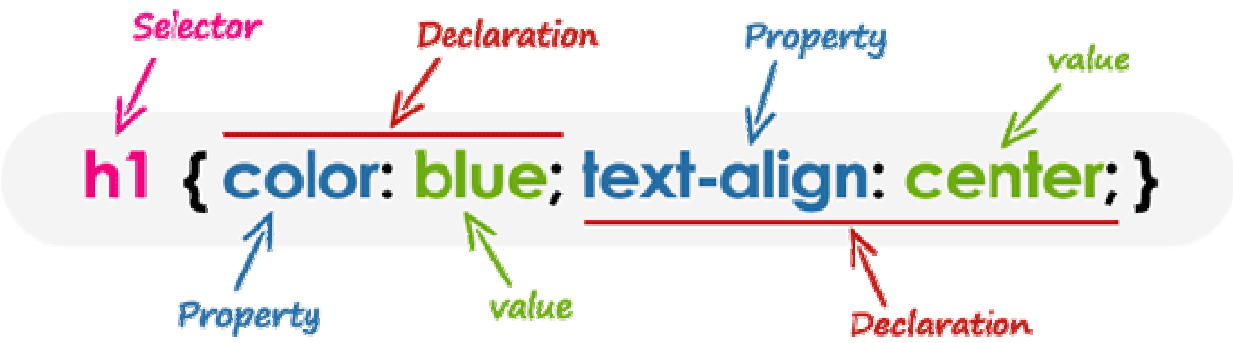
What is New in CSS3

- 2D Transforms
- Backgrounds & Borders
- Color
- Values and Units
- Selectors
- Web Fonts
- **Media Queries**
- Namespaces
- 3D Transforms
- Animations
- Gradient
- **CSS Exclusions (Floats)**
- **Flexible Box (“Flexbox”) Layout**
- **Grid Layout**
- **Multi-column Layout**
- Region
- SVG Filter Effects
- Text Shadow
- Transitions

Basics – Syntax

A CSS stylesheet consists of a set of rules that are interpreted by the web browser and then applied to the corresponding elements such as paragraphs, headings, etc. in the document.

A CSS rule have two main parts, a selector and one or more declarations:

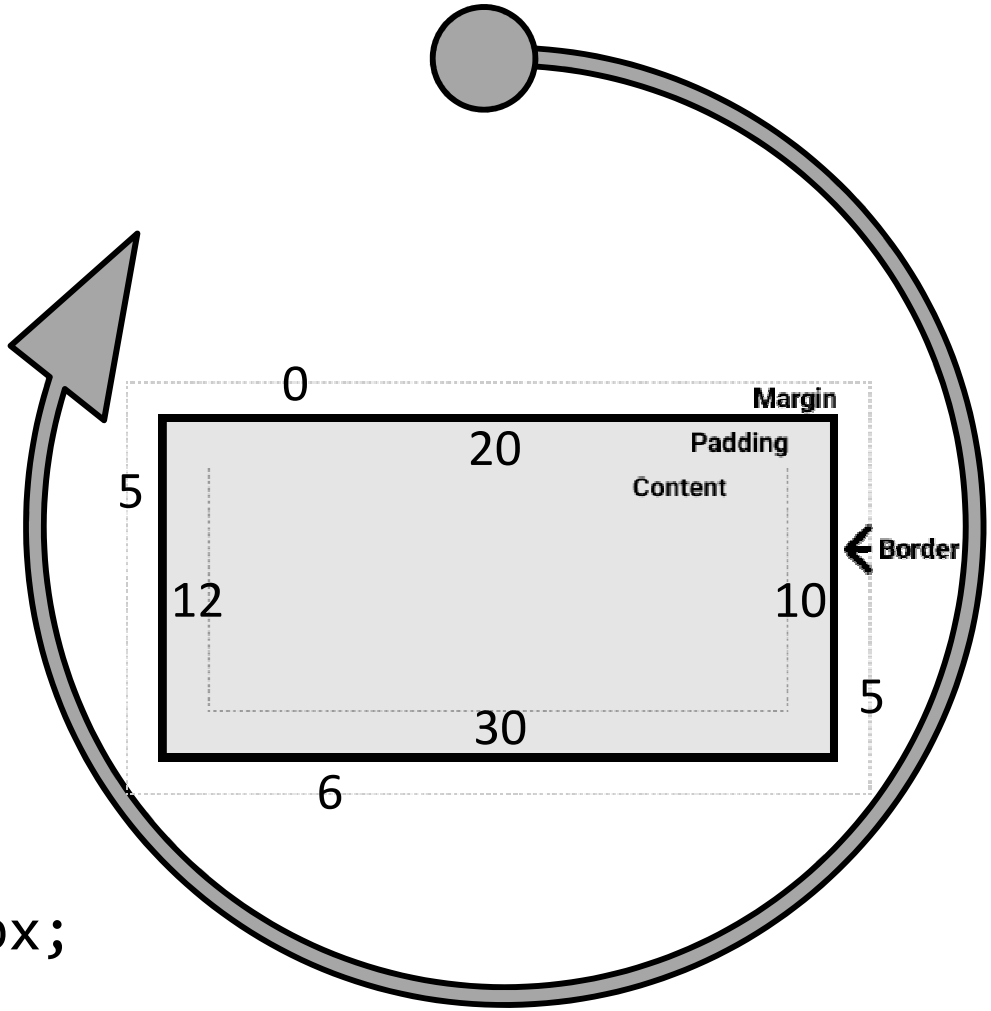


Basics – Shorthand

```
#container {
  margin-top: 0;
  margin-right: 5px;
  margin-bottom: 6px;
  margin-left: 5px;
  padding-top: 20px;
  padding-right: 10px;
  padding-bottom: 30px;
  padding-left: 12px;
}

vs.

#container {
  padding: 20px 10px 30px 12px;
  margin: 0px 5px 6px 5px;
}
```



Basics – Others

CSS property names and many values are **not case-sensitive**. Whereas, CSS selectors are usually case-sensitive, for instance, the class selector `.maincontent` is not the same as `.mainContent`.

A CSS **comment** begins with `/*`, and ends with `*/`

CSS **units** are:

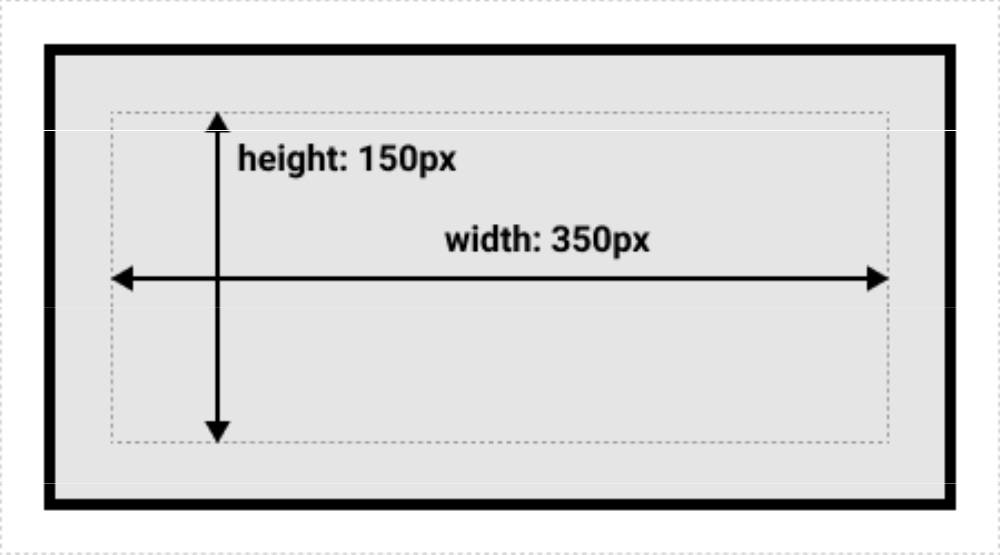
- cm, mm, Q, in, pc, pt, px (absolute)
- [r]em, ex, ch, re, lh, vw, vh, vmin, vmax (relative)

Basic – Box Model

The full CSS box model applies to **block boxes; inline boxes** only use some of the behavior defined in the box model. The model defines how the different parts of a box – margin, border, padding, and content – work together to create a box that you can see on the page.

Basic – Box Model

```
.box { width: 350px; height: 150px;
margin: 10px; padding: 25px; border: 5px
solid black; }
```



Final dimension: width = 410px, height = 210px

Basic – Box Model

```
.box { box-sizing: border-box; }
```

Final dimension: width = 350px, height = 150px


CSS
IS
AWESOME

Selectors

A CSS selector is a **pattern to match the elements** on a web page. The style rules associated with that selector will be applied to the elements that match the selector pattern.

Universal Selector

The universal selector, denoted by an asterisk (*), **matches every single element** on the page. It may be omitted if other conditions exist on the element.

```
* { margin: 0; padding: 0; }
```

Note: It is recommended *not to use* the universal selector too often in a production environment, since this selector matches every element on a web page that puts too much of unnecessary pressure on the browsers. Use element type or class selector instead.

Element Type Selectors

An element type selector **matches all instance of the element** in the document with the corresponding element type name. Let's try out an example to see how it actually works:

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

Id Selectors

The id selector is used to define style rules for a ***single* or *unique*** element.

The id selector is defined with a hash sign (#) immediately followed by the id value.

```
#error { color: red; }
```

Note: The value of an id attribute must be unique within a given document — meaning no two elements in your HTML document can share the same id value.

Class Selectors

The class selectors can be used to select any HTML **element that has a class attribute**. All the elements having that class will be formatted according to the defined rule.

The class selector is defined with a period sign (`.`) immediately followed by the class value.

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

Descendant Selectors

You can use these selectors when you need to select an element that is the **descendant of another element**, for example, if you want to target only those anchors that are contained within an unordered list, rather than targeting all anchor elements.

```
ul.menu li a { text-decoration: none; }
h1 em { color: green; }
```

Child Selectors

A child selector is used to select only those elements that are the direct children of some element.

A child selector is made up of two or more selectors separated by a greater than symbol (>). You can use this selector, for instance, to select the first level of list elements inside a nested list that has more than one level.

```
ul > li { list-style: square; }
ul > li ol { list-style: none; }
```

Adjacent Sibling Selectors

The adjacent sibling selectors can be used to **select sibling elements** (i.e. elements at the same level). This selector has the syntax like: `E1 + E2`, where E2 is the target of the selector.

```
h1 + p { color: blue; font-size: 18px; }
ul.task + p { color: #f0f; text-indent: 30px; }
```

General Sibling Selectors

The general sibling selector is similar to the adjacent sibling selector (E1 + E2), but it **is less strict**. A general sibling selector is made up of two simple selectors separated by the tilde (~) character. It can be written like: E1 ~ E2, where E2 is the target of the selector.

```
h1 ~ p { color: blue; font-size: 18px; }
ul.task ~ p { color: #f0f; text-indent: 30px; }
```

Attributes Selectors

The CSS attribute selectors provides an easy and powerful way to apply the styles on HTML elements based on the **presence of a particular attribute** or attribute value.

```
[title] { color: blue; }
abbr[title] { color: red; }
input[type="submit"] { border: 1px solid green; }

[class~="warning"],
[class*="error"]
[lang|=en],
a[href^="http://"],
a[href$=".pdf"] { color: green; }
```

Pseudo-classes Selectors

The pseudo-classes allow you to style the **dynamic states of an element** such as hover, active and focus state, as well as elements that are existing in the document tree but can't be targeted via the use of other selectors without adding any IDs or classes to them. A pseudo-class starts with a colon (`:`).

```
a:visited { text-decoration: none; }
ol li:first-child { border-top: none; }
table tr:nth-child(2n) td { background: #eee; }
q.red:lang(cs) { quotes: "~" "~"; }
```

List of Pseudo-classes

- :link
- :visited
- :hover
- :active
- :focus
- :enabled
- :disabled
- :checked
- :indeterminate
- :target
- :default
- :valid
- :invalid
- :in-range
- :out-of-range
- :required
- :optional
- :read-only
- :read-write

Pseudo-elements Selectors

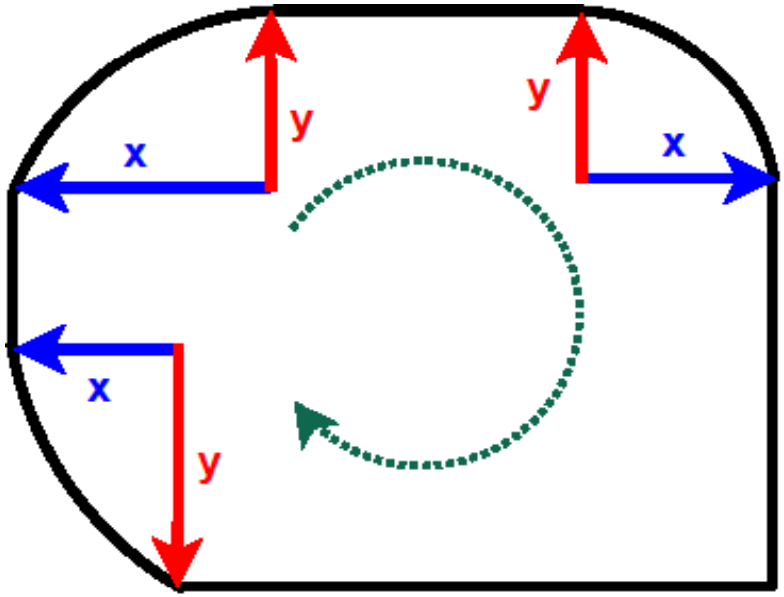
The pseudo-elements allow you to style the elements or parts of the elements without adding any IDs or classes to them. Such elements **aren't explicitly defined** by a position in the document tree. A pseudo-element starts with a double-colon (`::`).

```
h1::before { content: url("images/marker.gif"); }
p.article::first-letter { font-size: 2em; }
```

Examples

Border Radius

```
#container {
  border-radius: 5px 10px 10px 10px / 10px 10px 5px 5px;
  border-radius: 5px 10px / 10px;
  border-radius: 10px;
}
```



Opacity

The `opacity` property sets the opacity level for an element. The opacity-level describes the transparency-level, where 1 is not transparent at all, 0.5 is 50% see-through, and 0 is completely transparent.

```
div.first { opacity: 0.2; }
div.second { background: rgba(76, 175, 80, 0.5); }
```

opacity 0.2

opacity 0.5

opacity 1 (default)

Shadows

```
h1 {
  text-shadow: 2px 2px 8px #ff0000;
}
```

Text-shadow with blur effect

```
#container {
  box-shadow: 5px 5px 8px blue,
    10px 10px 8px red,
    15px 15px 8px green;
}
```

Define multiple shadows with blur effect.

Multiple Background Images

```
#example {
  width: 500px;
  height: 250px;
  background-image: url("decoration.png"), url("ribbon.png"),
    url("old_paper.jpg");
  background-repeat: no-repeat;
  background-position: left top, right bottom, left top;
}
```

Multi-column Layout

The `column-count` property specifies the number of columns an element should be divided into.

```
.newspaper {
  column-count: 3;
  column-gap: 40px;
  column-rule: 4px double #ff00ff;
}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Typi non habent claritatem insitam; est usus legentis in iis qui facit eorum claritatem. Investigationes demonstraverunt lectores legere me lius quod ii legunt saepius.

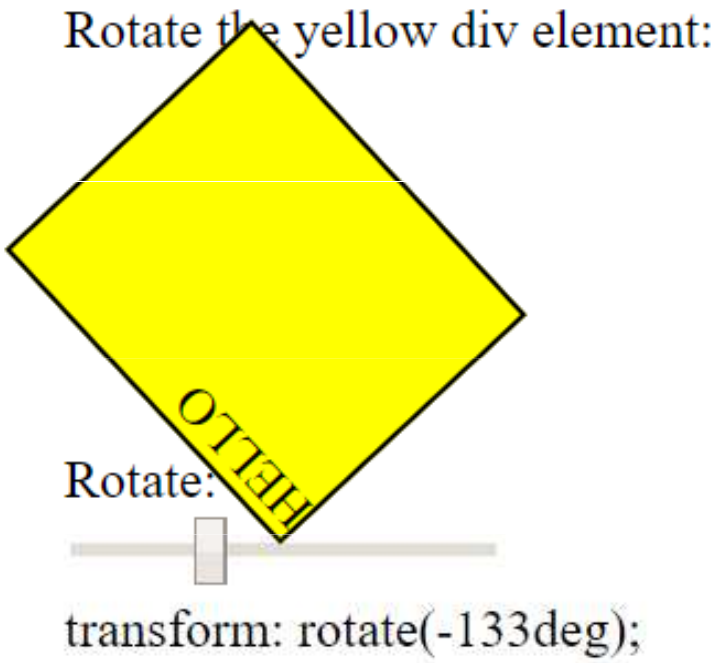
Transform

The `transform` property applies a 2D or 3D transformation to an element. This property allows you to rotate, scale, move, skew, etc., elements.

```
div.a {
  transform: rotate(20deg);
}

div.b {
  transform: skewY(20deg);
}

div.c {
  transform: scaleY(1.5);
}
```

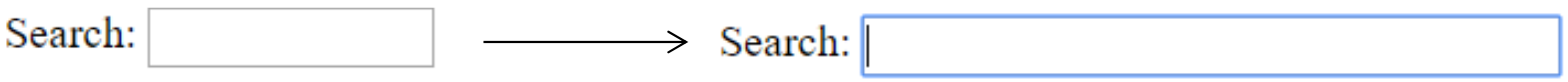


Transition

When an <input type="text"> gets focus, gradually change the width from 100px to 250px:

```
input[type=text] {
  width: 100px;
  transition: width .35s ease-in-out;
}

input[type=text]:focus {
  width: 250px;
}
```



Media Query

The Media Queries Module specifies methods to enable web developers to scope a style sheet to a set of **precise device capabilities**.

Media types:

- Desktop browser, screen, print, braille
- Mobile browser
- Tablet
- Television
- Game console

Examples: https://mediaqueri.es/

Media Query

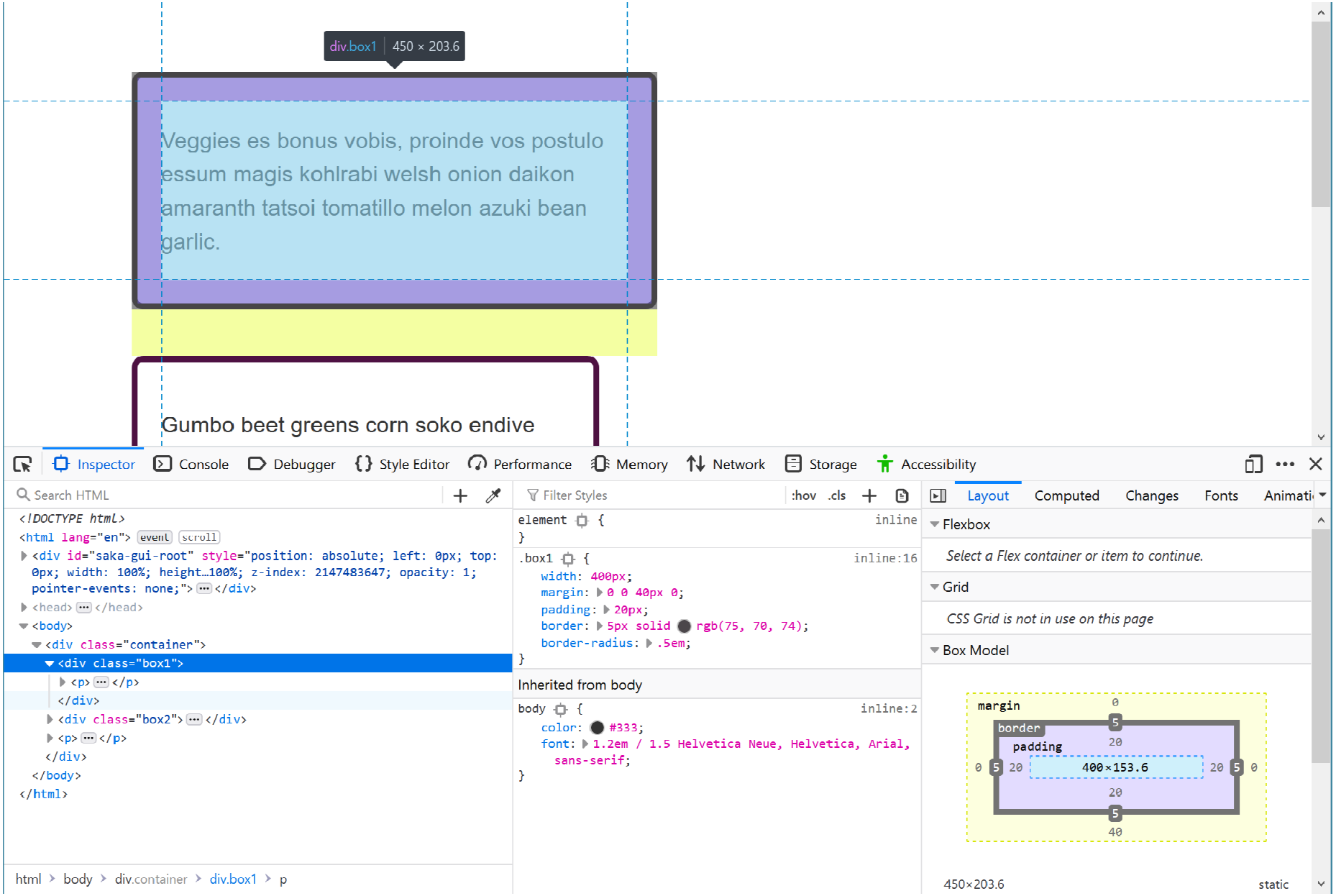
```
@media screen and (max-width: 600px) {
  body {
    font-size: 80%;
  }
}

@media screen and (min-width: 320px) and (max-width: 480px) {}
@media not print and (max-width: 600px) {}
```

Media properties are:

min/max-width, min/max-height, device-width, device-height, orientation, aspect-ratio, device-aspect-ratio, color, color-index, monochrome, resolution

Debugging CSS



References and Tools

- https://www.w3.org/Style/CSS/specs.en.html
- https://css3test.com/
- https://css3generator.com/
- http://css3pie.com/

- https://flukeout.github.io/
- https://flexboxfroggy.com/
- https://cssgridgarden.com/