CSS Advances

<Training Topic /Lesson Name>

Lesson Objectives

- <Mandatory slide>
- <Brief the objectives of the course: What trainees learn after the course>
- <Remember to update the course version in Notes part>

Session 1

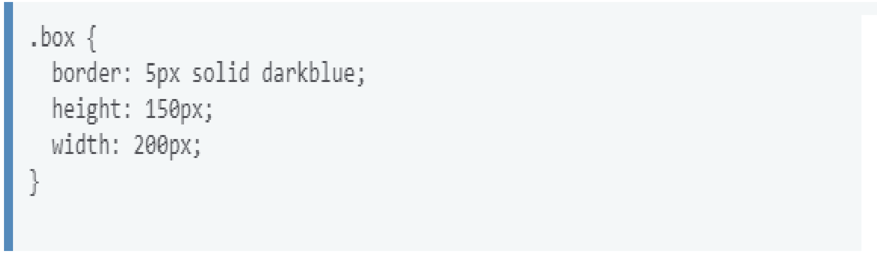
SIZING ITEM

Overview

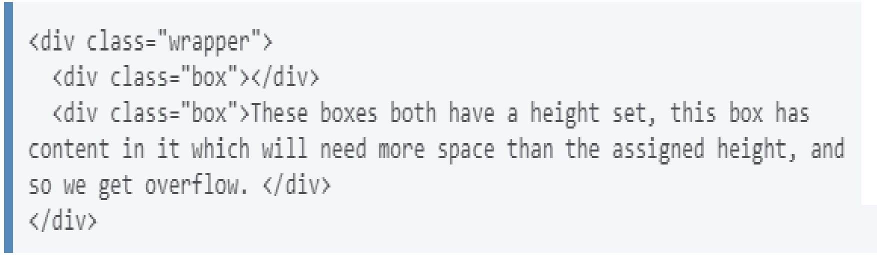
1. Setting a specific size
2. Using percentage
3. Percentage margin and padding
4. Min- and max- size
5. Viewport unit

1. Setting a specific size

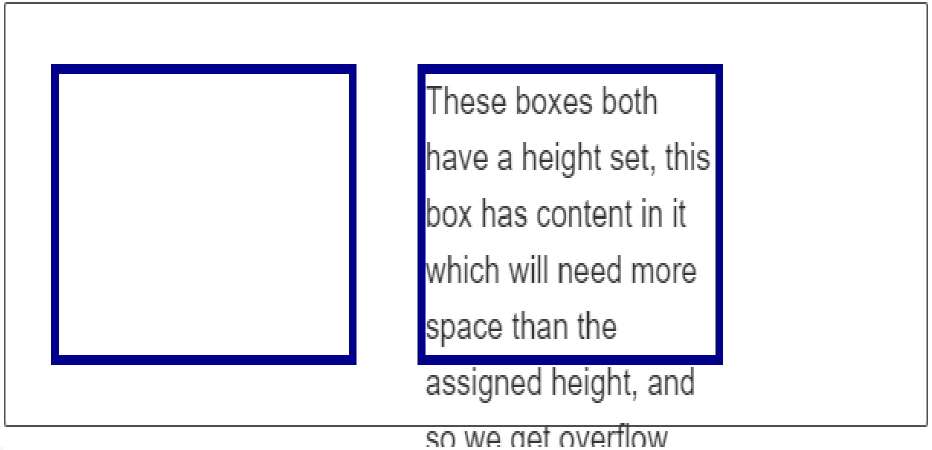
- The **height** and **width** properties are used to set the height and width ***(size)*** of an element.
- Setting height can cause content to ***overflow*** if there is more content than the element has space to fit inside it



```css
.box {
  border: 5px solid darkblue;
  height: 150px;
  width: 200px;
}
```



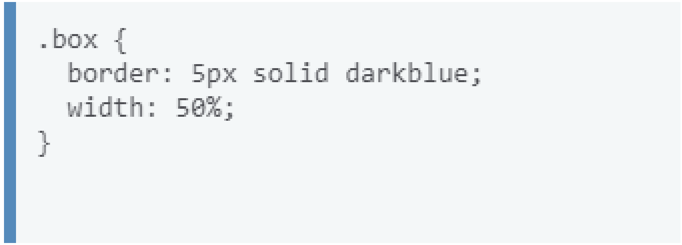
```html
<div class="wrapper">
  <div class="box"></div>
  <div class="box">These boxes both have a height set, this box has
content in it which will need more space than the assigned height, and
so we get overflow. </div>
</div>
```



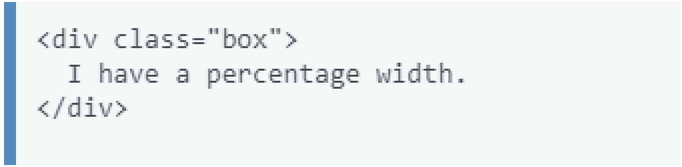
These boxes both
have a height set, this
box has content in it
which will need more
space than the
assigned height, and
so we get overflow

2. Using percentage

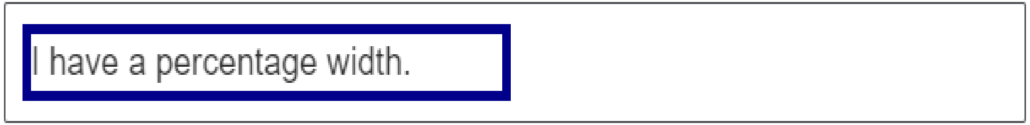
- Due to the problem of overflow, fixing the height of elements with lengths or **percentages** is a solution
- Using a ***percentage*** you need to be aware what it is a percentage of.



```
.box {
  border: 5px solid darkblue;
  width: 50%;
}
```



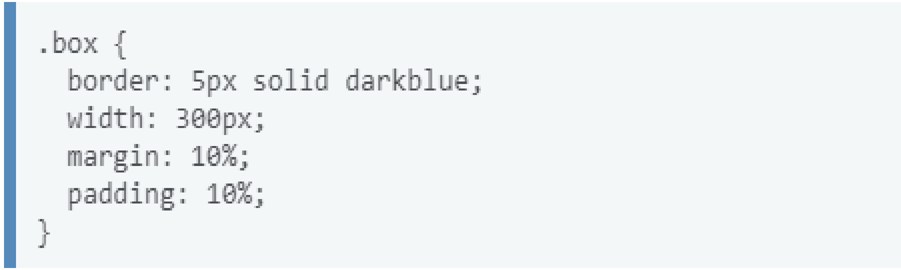
```
<div class="box">
  I have a percentage width.
</div>
```



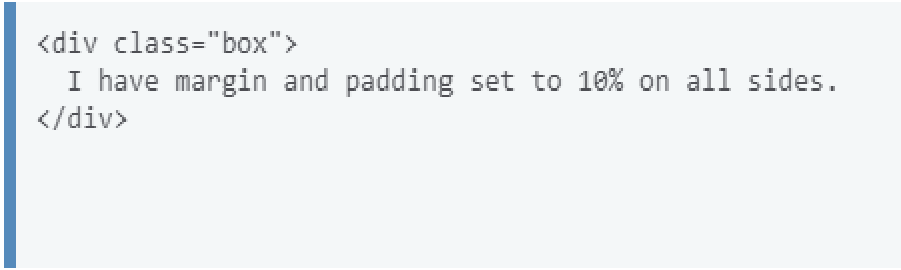
I have a percentage width.

3. Percentage margin and padding

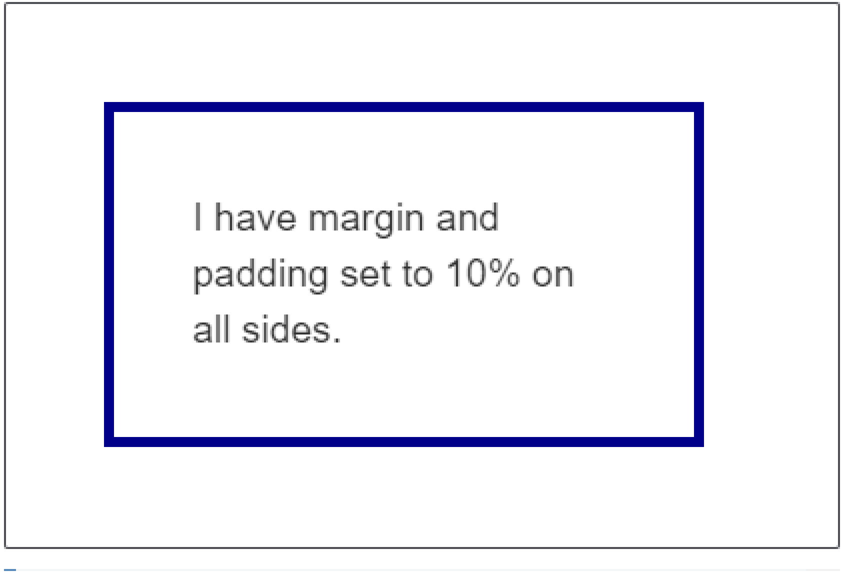
- The ***margin*** and ***padding*** of a child element in percentage is the percentage of the width of its parent element



```
.box {
  border: 5px solid darkblue;
  width: 300px;
  margin: 10%;
  padding: 10%;
}
```



```
<div class="box">
  I have margin and padding set to 10% on all sides.
</div>
```

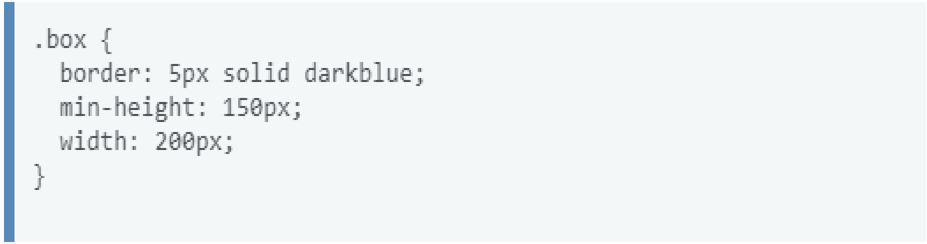


I have margin and padding set to 10% on all sides.

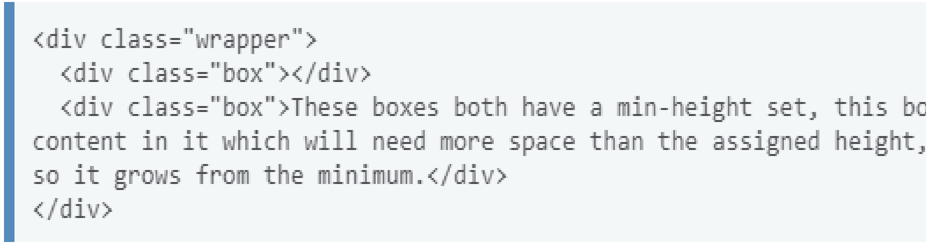
4. Min- and max- size

- We can give an element a **minimum** or a **maximum** size

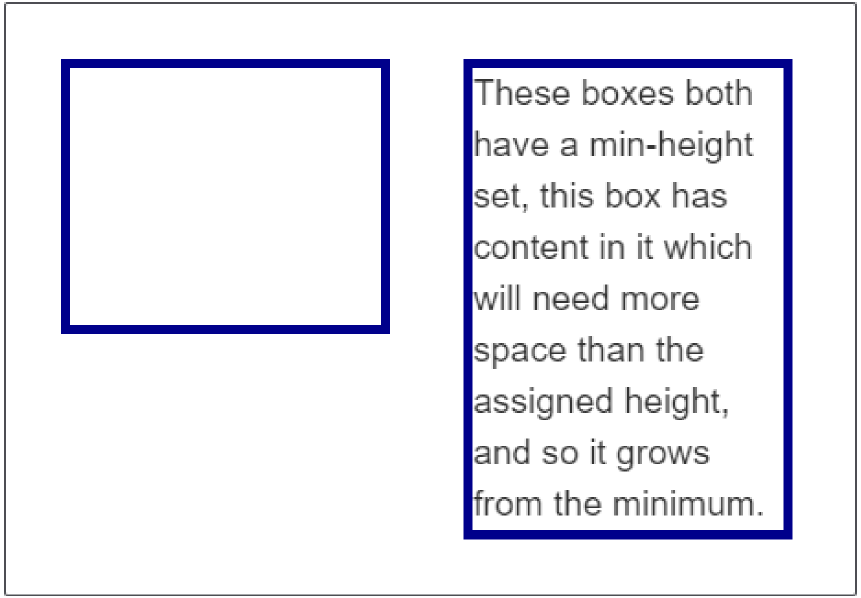
=> This is very useful for dealing with variable amounts of content while avoiding overflow.



```
.box {
  border: 5px solid darkblue;
  min-height: 150px;
  width: 200px;
}
```



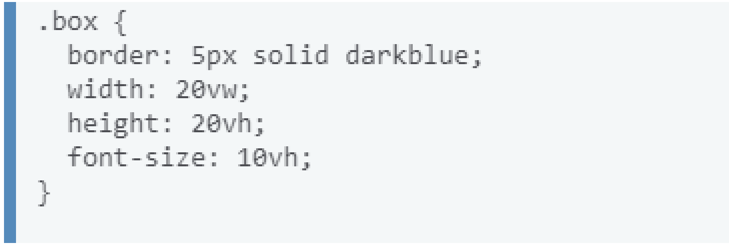
```
<div class="wrapper">
  <div class="box"></div>
  <div class="box">These boxes both have a min-height set, this bo
content in it which will need more space than the assigned height,
so it grows from the minimum.</div>
</div>
```



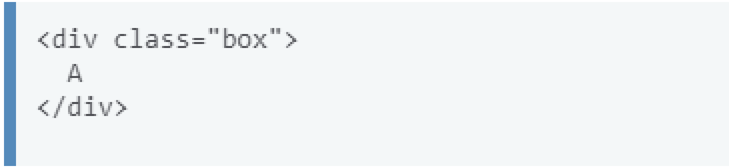
These boxes both have a min-height set, this box has content in it which will need more space than the assigned height, and so it grows from the minimum.

5. Viewport unit

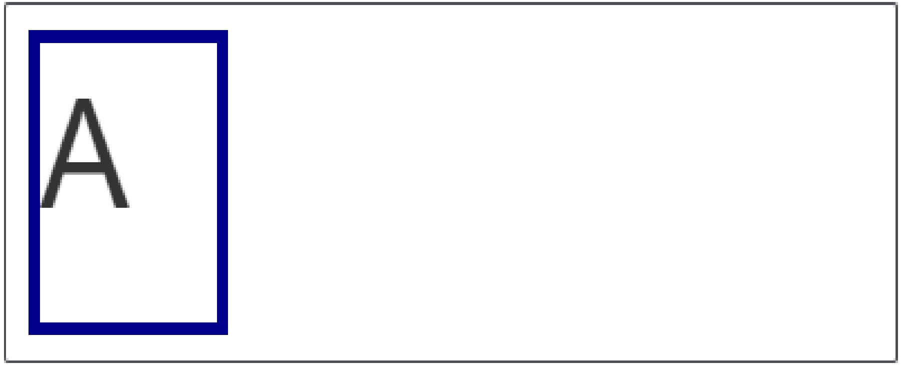
- The **viewport** is the user's visible area of a web page.
- In CSS we have units which relate to the size of the **viewport** — the ***vw*** unit for viewport width, and ***vh*** for viewport height.
- **1vh** is equal to 1% of the viewport height
- **1vw** is equal to 1% of the viewport width



```
.box {
  border: 5px solid darkblue;
  width: 20vw;
  height: 20vh;
  font-size: 10vh;
}
```



```
<div class="box">
  A
</div>
```



Session 2

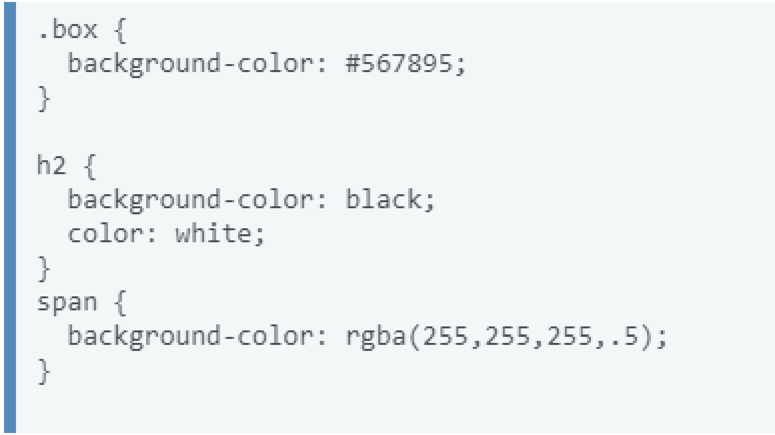
BACKGROUND

Overview

1. **Background color**
2. **Background images**
3. **Sizing/positioning background images**
4. **Gradient backgrounds**
5. **Multiple background images**
6. **Background shorthand**

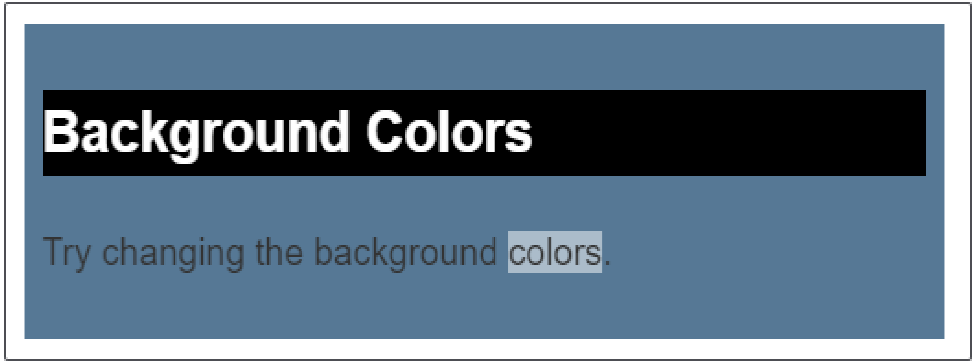
1. Background color

- The ***background-color*** property specifies the background color of an element.
- The property accepts any valid color: color name, RGB, HEX,..



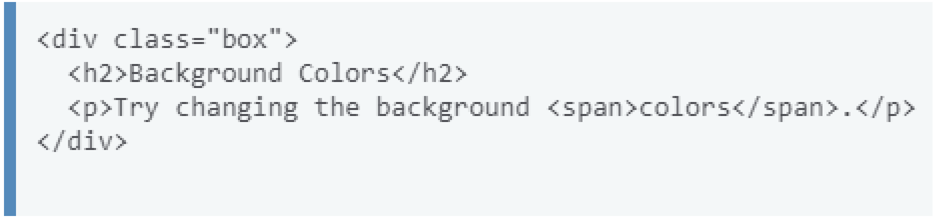
```
.box {
  background-color: #567895;
}

h2 {
  background-color: black;
  color: white;
}
span {
  background-color: rgba(255,255,255,.5);
}
```



Background Colors

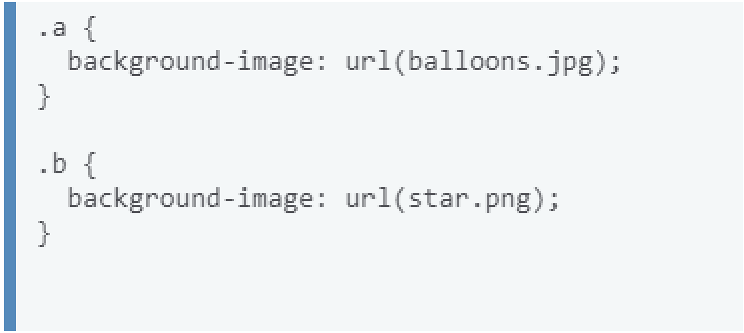
Try changing the background colors.



```
<div class="box">
  <h2>Background Colors</h2>
  <p>Try changing the background <span>colors</span>.</p>
</div>
```

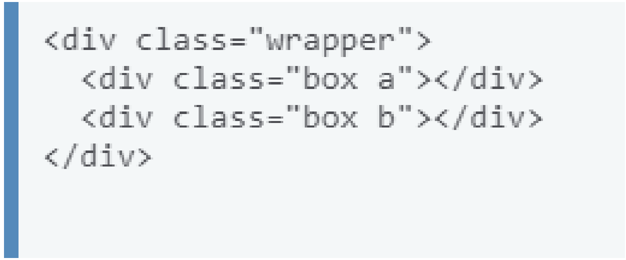
2. Background images

- The ***background-image*** property specifies an image to use as the background of an element.
- By default, the image is ***repeated*** so it covers the entire element.

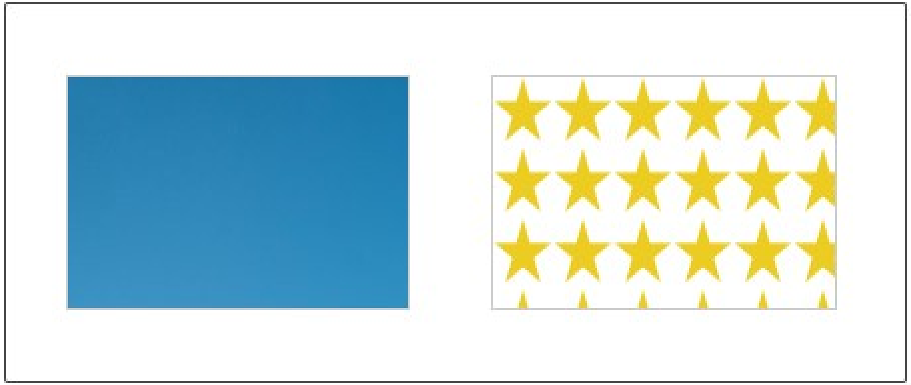


```
.a {
  background-image: url(balloons.jpg);
}

.b {
  background-image: url(star.png);
}
```



```
<div class="wrapper">
  <div class="box a"></div>
  <div class="box b"></div>
</div>
```



2. Background images

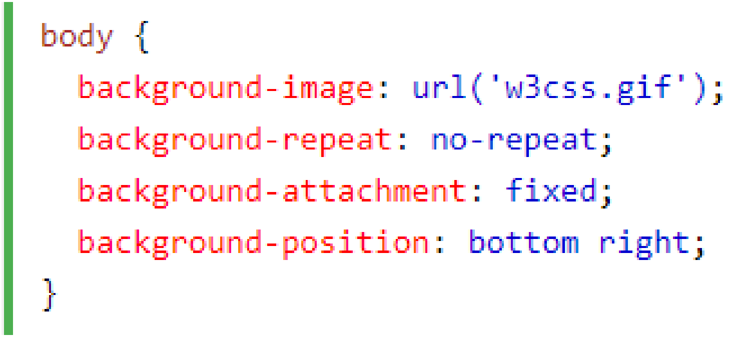
- **Controlling background-repeat:** The background-repeat property is used to control the tiling behavior of images. The available values are
  - *no-repeat* — stop the background from repeating altogether.
  - *repeat-x* — repeat horizontally.
  - *repeat-y* — repeat vertically.
  - *repeat* — the default; repeat in both directions.

3. Sizing/positioning background images

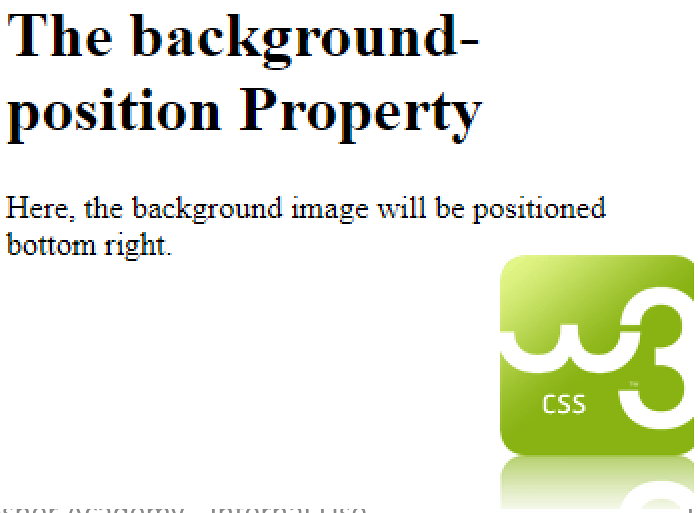
- **Sizing the background image:** We can use the ***background-size*** property, which can take length or percentage values, to size the image to fit inside the background or you can use keyword ***cover*** and ***contain***
  - ***cover****: the browser will make the image just large enough so that it completely covers the box area while still retaining its aspect ratio*
  - ***contain****: the browser will make the image the right size to fit inside the box.*

3. Sizing/positioning background images

- **Positioning the background image:** The ***background-position*** property allows you to choose the position in which the background image appears on the box it is applied to. You can use keywords such as ***top***, ***right, bottom, left, center*** and ***length*** or ***percentage***



```
body {
  background-image: url('w3css.gif');
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-position: bottom right;
}
```

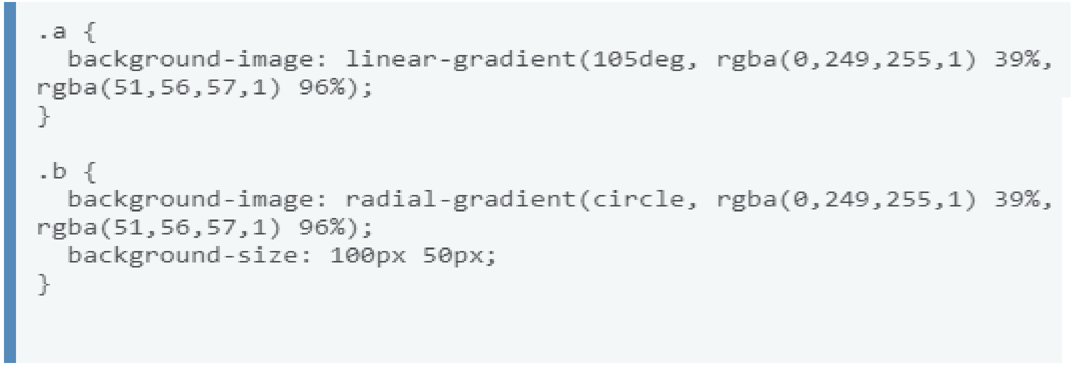


The background-position Property

Here, the background image will be positioned bottom right.

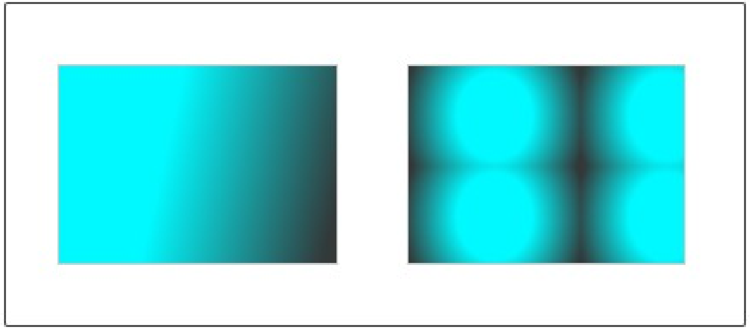
4. Gradient backgrounds

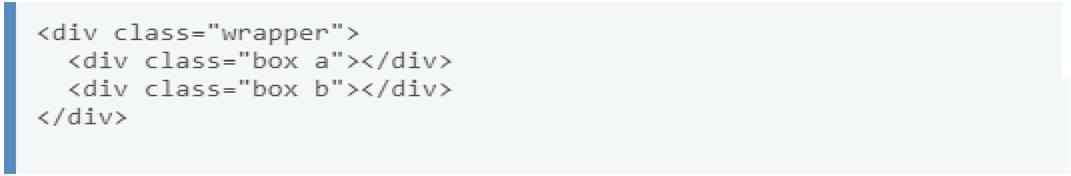
- ***A gradient*** — when used for a background — acts just like an image and is also set by using the ***background-image*** property.
- ***CSS gradients*** let you display smooth transitions between two or more specified colors.



```
.a {
  background-image: linear-gradient(105deg, rgba(0,249,255,1) 39%,
rgba(51,56,57,1) 96%);
}

.b {
  background-image: radial-gradient(circle, rgba(0,249,255,1) 39%,
rgba(51,56,57,1) 96%);
  background-size: 100px 50px;
}
```

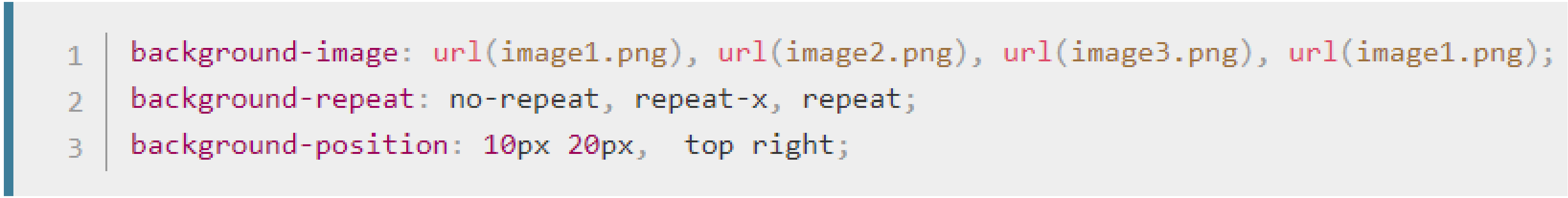




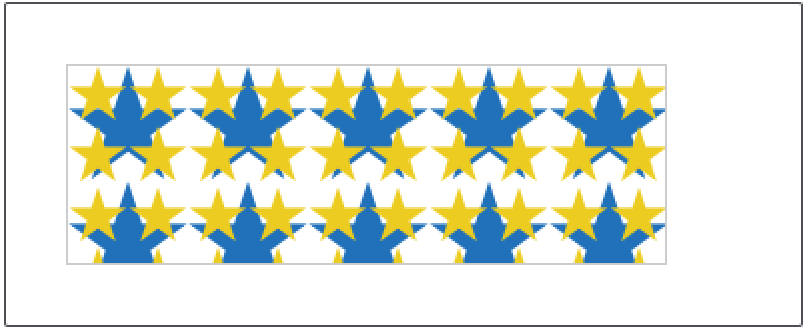
```
<div class="wrapper">
  <div class="box a"></div>
  <div class="box b"></div>
</div>
```

5. Multiple background images

- You can specify ***multiple background-image*** values in a single property value, separating each one with a comma.
- The other **background-*** properties can also have values comma-separated in the same way as background-image

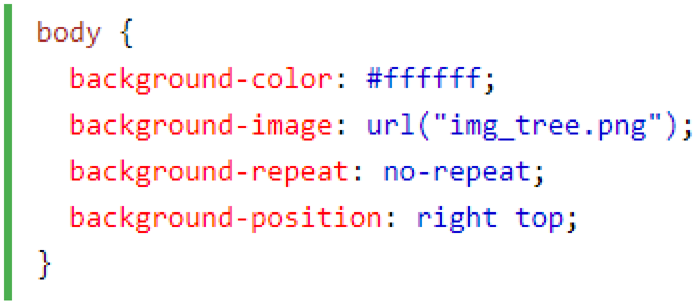


```css
background-image: url(image1.png), url(image2.png), url(image3.png), url(image1.png);
background-repeat: no-repeat, repeat-x, repeat;
background-position: 10px 20px,  top right;
```



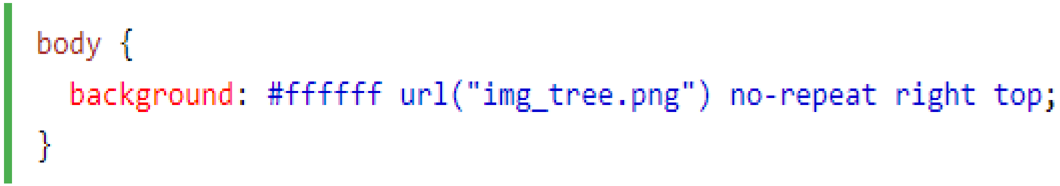
6. Background shorthand

- To shorten the code, it is also possible to specify all the background properties in one single property. This is called a **shorthand** property.
- Instead of writing:



```
body {
  background-color: #ffffff;
  background-image: url("img_tree.png");
  background-repeat: no-repeat;
  background-position: right top;
}
```

- You can use the shorthand property background



```
body {
  background: #ffffff url("img_tree.png") no-repeat right top;
}
```

6. Background shorthand

- When using the **shorthand** property the order of the property values is:
  - *background-color*
  - *background-image*
  - *background-repeat*
  - *background-attachment*
  - *background-position*

Session 3

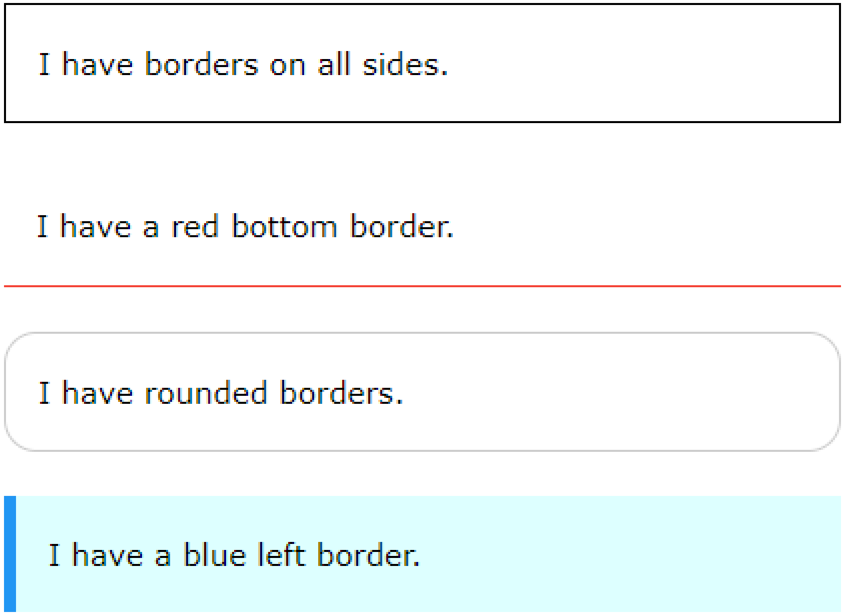
BORDER

Overview

1. **Borders**
2. **Rounded borders**

1. Borders

- The **CSS border** properties allow you to specify the style, width, and color of an element's border.



I have borders on all sides.

I have a red bottom border.

I have rounded borders.

I have a blue left border.

1. Borders

- The ***border-style*** property specifies what kind of border to display.
  - ***dotted*** - Defines a dotted border
  - ***dashed*** - Defines a dashed border
  - ***solid*** - Defines a solid border
  - ***double*** - Defines a double border
  - ***groove*** - Defines a 3D grooved border. The effect depends on the border-color value
  - ***inset*** - Defines a 3D inset border. The effect depends on the border-color value
  - ***outset*** - Defines a 3D outset border. The effect depends on the border-color value
  - ***outset*** - Defines a 3D outset border. The effect depends on the border-color value
  - ***hidden*** - Defines a hidden border

1. Borders

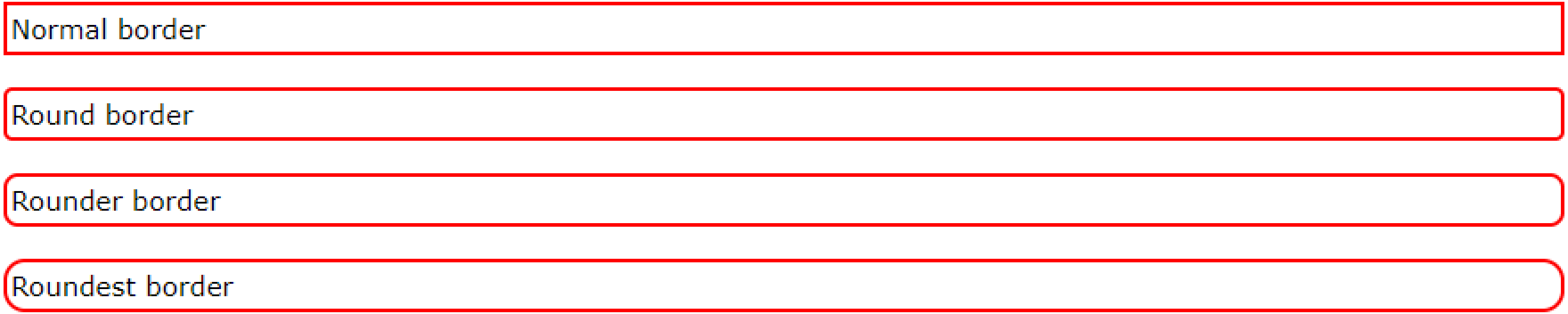
- The ***border-width*** property specifies the width of the four borders.
- The width can be set as a specific size (in px, pt, cm, em, etc) or by using one of the three pre-defined values: thin, medium, or thick.
- The ***border-color*** property is used to set the color of the four borders. The color can be set by: name color, HEX, RGB, HSL,..
- It is possible to specify a different border for each side. In CSS, there are also properties for specifying each of the borders (top, right, bottom, and left)

1. Borders

- ***Border shorthand***: To shorten the code, it is also possible to specify all the individual border properties in one property.
- The ***border*** property is ***a shorthand property*** for the following individual border properties:
  - *border-width*
  - *border-style* (**required**)
  - *border-color*

2. Rounded borders

- The ***border-radius*** property is used to add ***rounded borders*** to an element
- The ***border-radius*** property is actually ***a shorthand property*** for the *border-top-left-radius*, *border-top-right-radius*, *border-bottom-right-radius* and *border-bottom-left-radius* properties



Normal border

Round border

Rounder border

Roundest border

Session 4

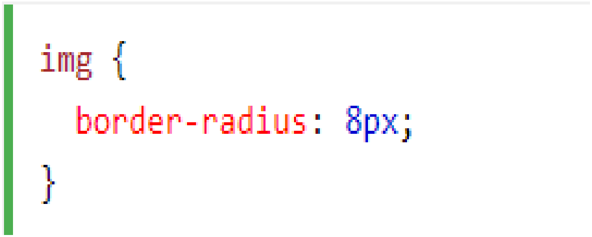
STYLING IMAGES AND FORMS

Overview

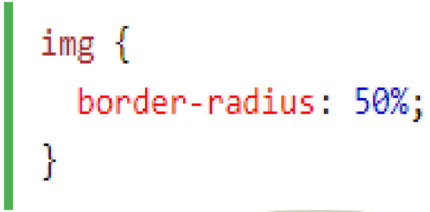
1. **Styling images**
2. **Styling forms**

1. Styling images

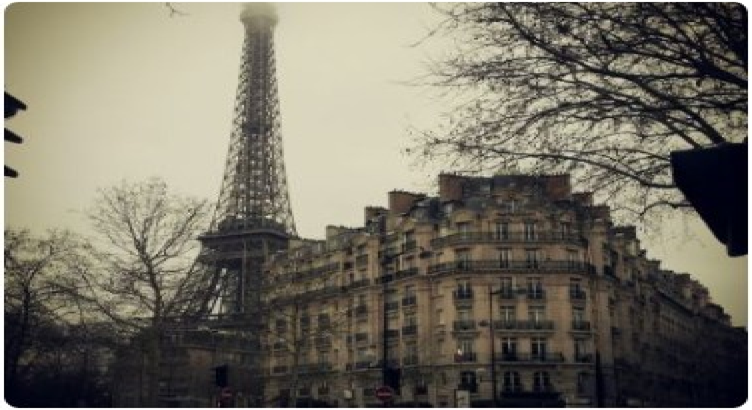
- Use the **border-radius** property to create rounded images

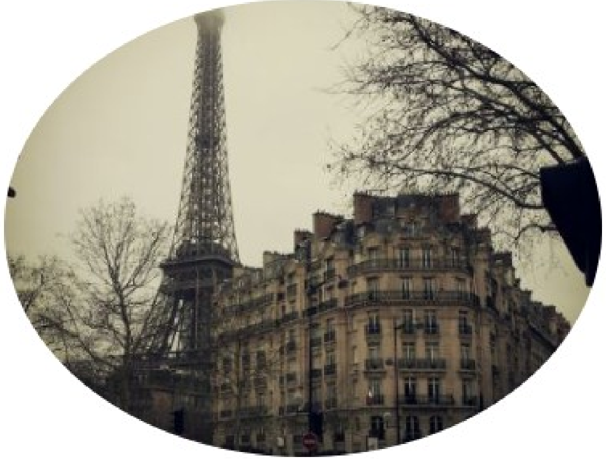


```
img {
  border-radius: 8px;
}
```



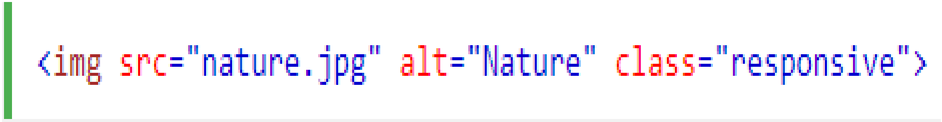
```
img {
  border-radius: 50%;
}
```





1. Styling images

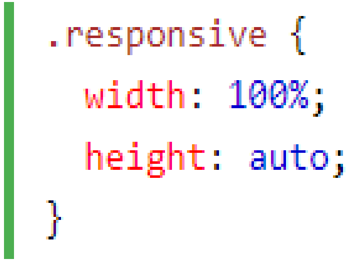
- **Responsive images** will automatically adjust to fit the size of the screen
  - Step1: Add HTML



```html
<img src="nature.jpg" alt="Nature" class="responsive">
```

  - Step 2: Add CSS

If you want the image to scale both up and down on responsiveness, set the CSS width property to 100% and height to auto

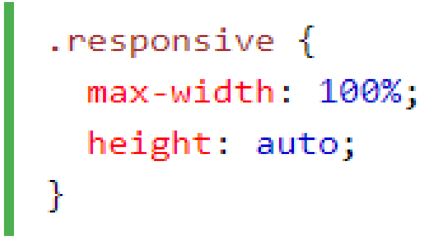


```css
.responsive {
  width: 100%;
  height: auto;
}
```

1. Styling images

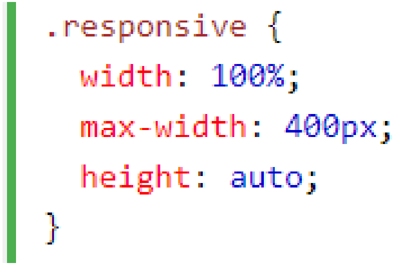
- **Responsive images**

If you want an image to scale down if it has to, but never scale up to be larger than its original size, use max-width: 100%



```
.responsive {
  max-width: 100%;
  height: auto;
}
```

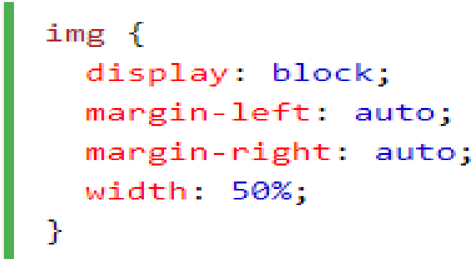
If you want to restrict a responsive image to a maximum size, use the max-width property, with a pixel value of your choice



```
.responsive {
  width: 100%;
  max-width: 400px;
  height: auto;
}
```

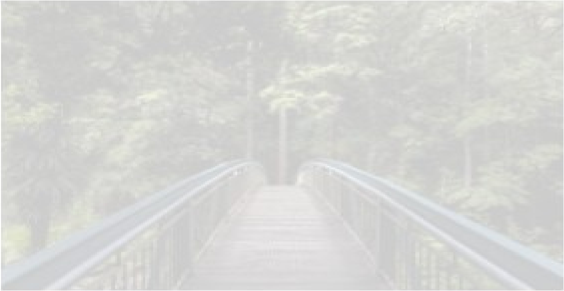
1. Styling images

- **Center an image:** To center an image, set left and right margin to auto and make it into a block element

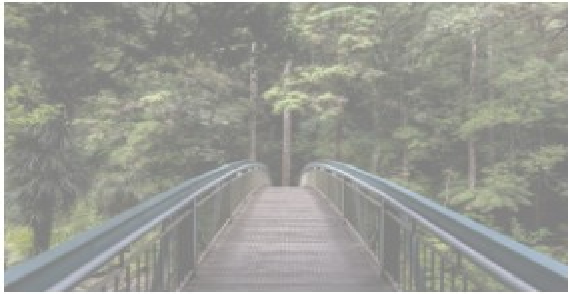


```
img {
  display: block;
  margin-left: auto;
  margin-right: auto;
  width: 50%;
}
```

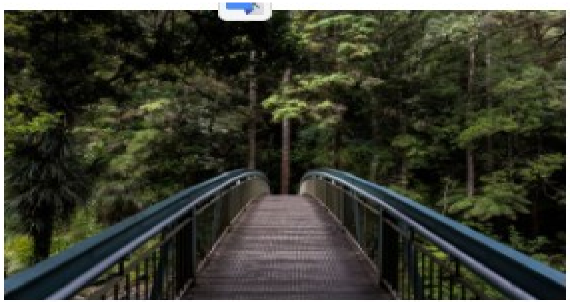
- **Transparent image:** The ***opacity*** property can take a value from 0.0 - 1.0. The lower value, the more transparent



opacity 0.2



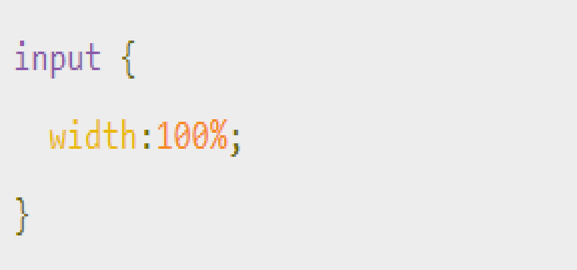
opacity 0.5



opacity 1
(default)

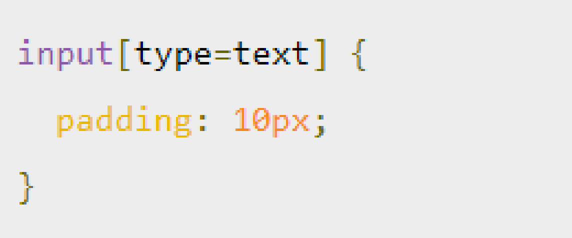
2. Styling forms

- **Styling Input Fields:**
  - Use the ***width*** property to determine the width of the input field



```
input {
  width:100%;
}
```

  - Use the ***padding*** property to add space inside the text field

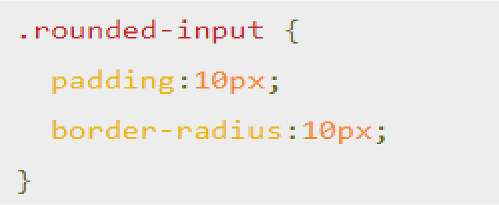


```
input[type=text] {
  padding: 10px;
}
```

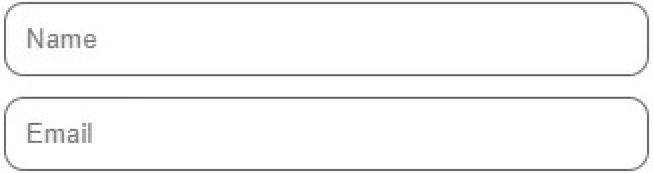
2. Styling forms

➢ Styling Input Fields:

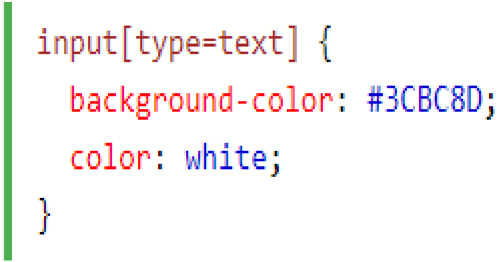
- Use the ***border*** property to change the border size and color, and use the ***border-radius*** property to add rounded corners



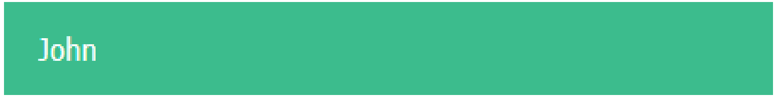
```
.rounded-input {
  padding:10px;
  border-radius:10px;
}
```



- Use the ***background-color*** property to add a background color to the input, and the ***color*** property to change the text color



```
input[type=text] {
  background-color: #3CBC8D;
  color: white;
}
```



Session 5

DEBUGGING

Overview

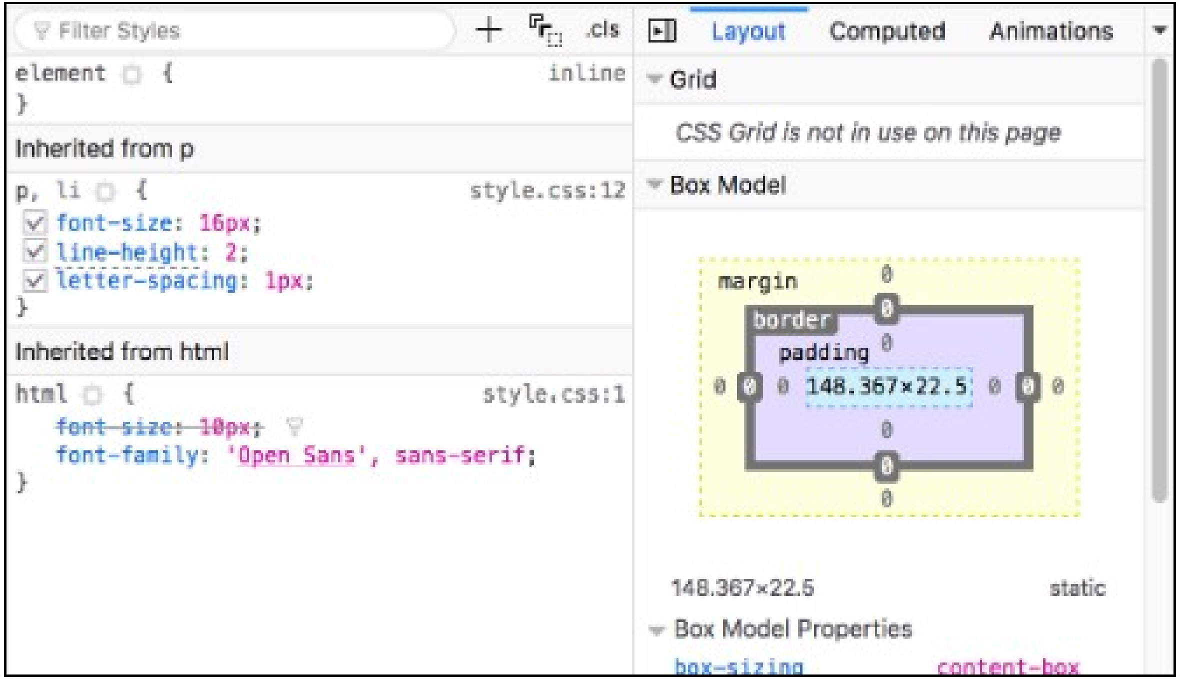
1. DevTools
2. Inspecting the applied CSS
3. Edit value
4. Adding property
5. Understand box model
6. Solving specificity problem

1. DevTools

- Every modern web browser includes a powerful suite of **developer tools.** These tools do a range of things, from inspecting currently-loaded HTML, CSS and JavaScript to showing which assets the page has requested and how long they took to load
- The **devtools** live inside your browser in a subwindow that looks roughly like this, depending on what browser you are using

2. Inspecting the applied CSS

- **Exploring the CSS editor:** By default, the CSS editor displays the CSS rules applied to the currently selected element



```
element {                                 inline
}
```

Inherited from p

```
p, li {                               style.css:12
  font-size: 16px;
  line-height: 2;
  letter-spacing: 1px;
}
```

Inherited from html

```
html {                                 style.css:1
  ~~font-size: 10px;~~
  font-family: 'Open Sans', sans-serif;
}
```

Grid

CSS Grid is not in use on this page

Box Model

148.367×22.5 static

Box Model Properties

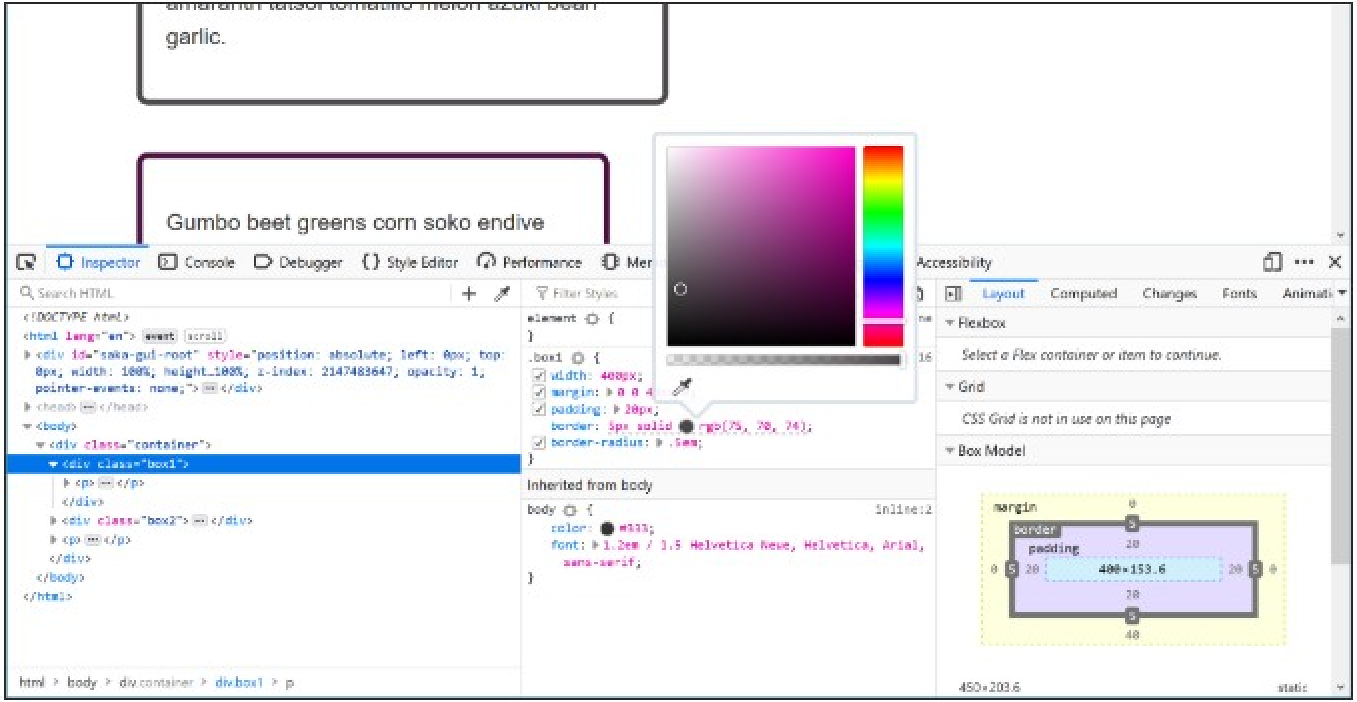
box-sizing content-box

2. Inspecting the applied CSS

- **Exploring the CSS editor**
  - Next to each rule is the file name and line number the rule is defined in. Clicking that rule causes the dev tools to jump to show it in its own view, where it can generally be edited and saved.
  - You can also click the closing curly brace of any rule to bring up a text box on a new line, where you can write a completely new declaration for your page

3. Editing value

- In addition to turning properties on and off, you can ***edit*** their values, DevTools can save you a lot of time editing a stylesheet and reloading the page.



garlic.

Gumbo beet greens corn soko endive

```
<!DOCTYPE html>
<html lang="en"> event scroll
▸ <div id="saka-gui-root" style="position: absolute; left: 0px; top:
  0px; width: 100%; height 100%; z-index: 2147483647; opacity: 1;
  pointer-events: none;"> … </div>
▸ <head> … </head>
▾ <body>
  ▾ <div class="container">
    ▾ <div class="box1">
      ▸ <p> … </p>
      </div>
    ▸ <div class="box2"> … </div>
    ▸ <p> … </p>
    </div>
  </body>
</html>
```

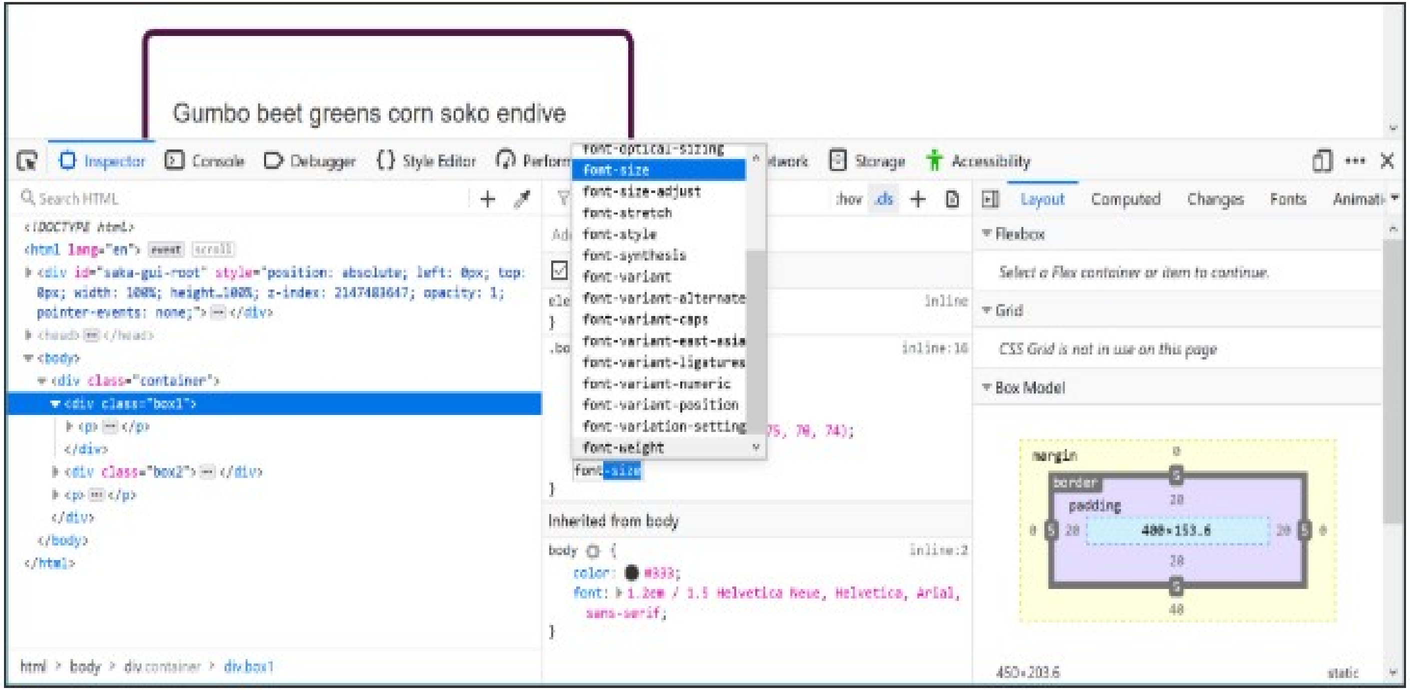
```
element {
}
.box1 {
☑ width: 400px;
☑ margin: ▸ 0 0 4
☑ padding: ▸ 20px;
  border: 5px solid ● rgb(75, 70, 74);
☑ border-radius: ▸ .5em;
}
Inherited from body
body {                                            inline:2
  color: ● #333;
  font: ▸ 1.2em / 1.5 Helvetica Neue, Helvetica, Arial,
    sans-serif;
}
```

Select a Flex container or item to continue.

CSS Grid is not in use on this page

4. Adding a new property

- You can add properties using the DevTools. You can try this out in DevTools before adding it to your CSS file.

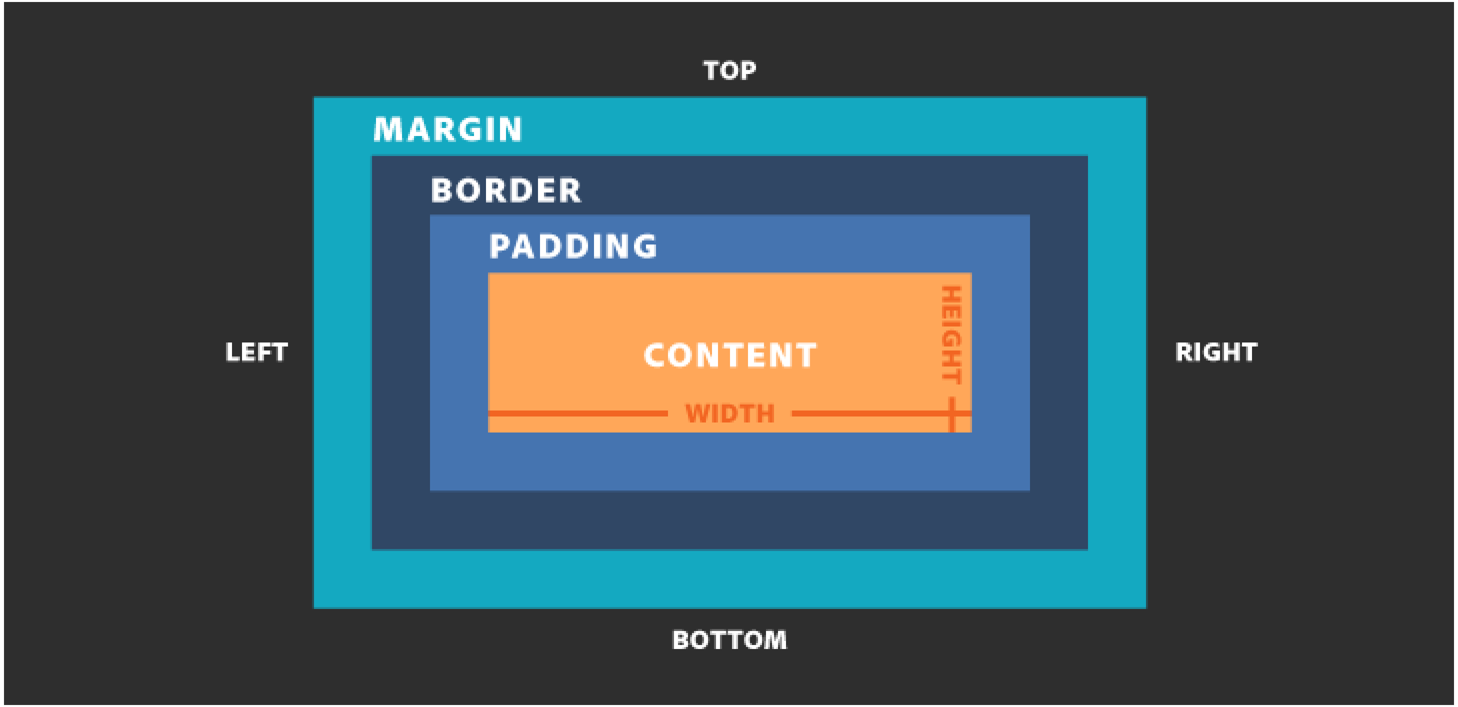


```
<!DOCTYPE html>
<html lang="en"> event scroll
  <div id="saka-gui-root" style="position: absolute; left: 0px; top: 0px; width: 100%; height…100%; z-index: 2147483647; opacity: 1; pointer-events: none;"> … </div>
  <head> … </head>
  <body>
    <div class="container">
      <div class="box1">
        <p> … </p>
      </div>
      <div class="box2"> … </div>
      <p> … </p>
    </div>
  </body>
</html>
```

```
body {                                              inline:2
  color: ● #333;
  font: 1.2em / 1.5 Helvetica Neue, Helvetica, Arial,
    sans-serif;
}
```

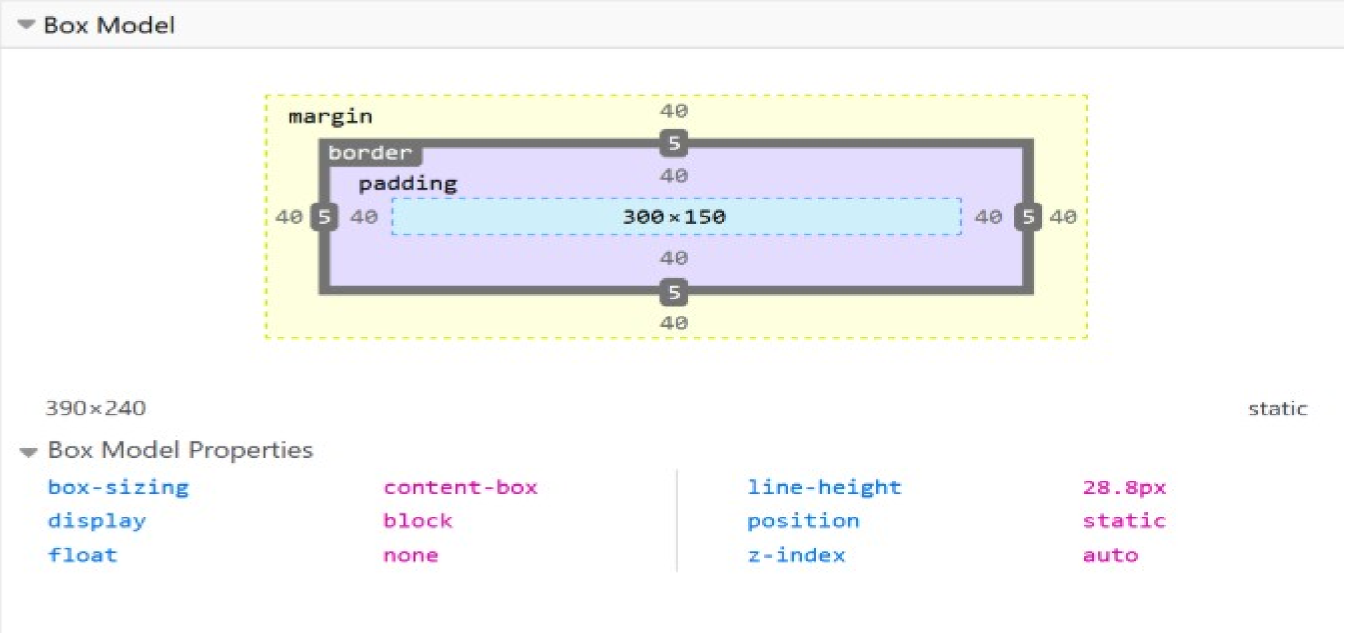
5. Understanding box model

- **Box model**



5. Understanding box model

- If you inspect an element in browser **DevTools**, you can see the size of the element plus its margin, padding, and border. Inspecting an element in this way is a great way to find out if your box is really the size you think it is



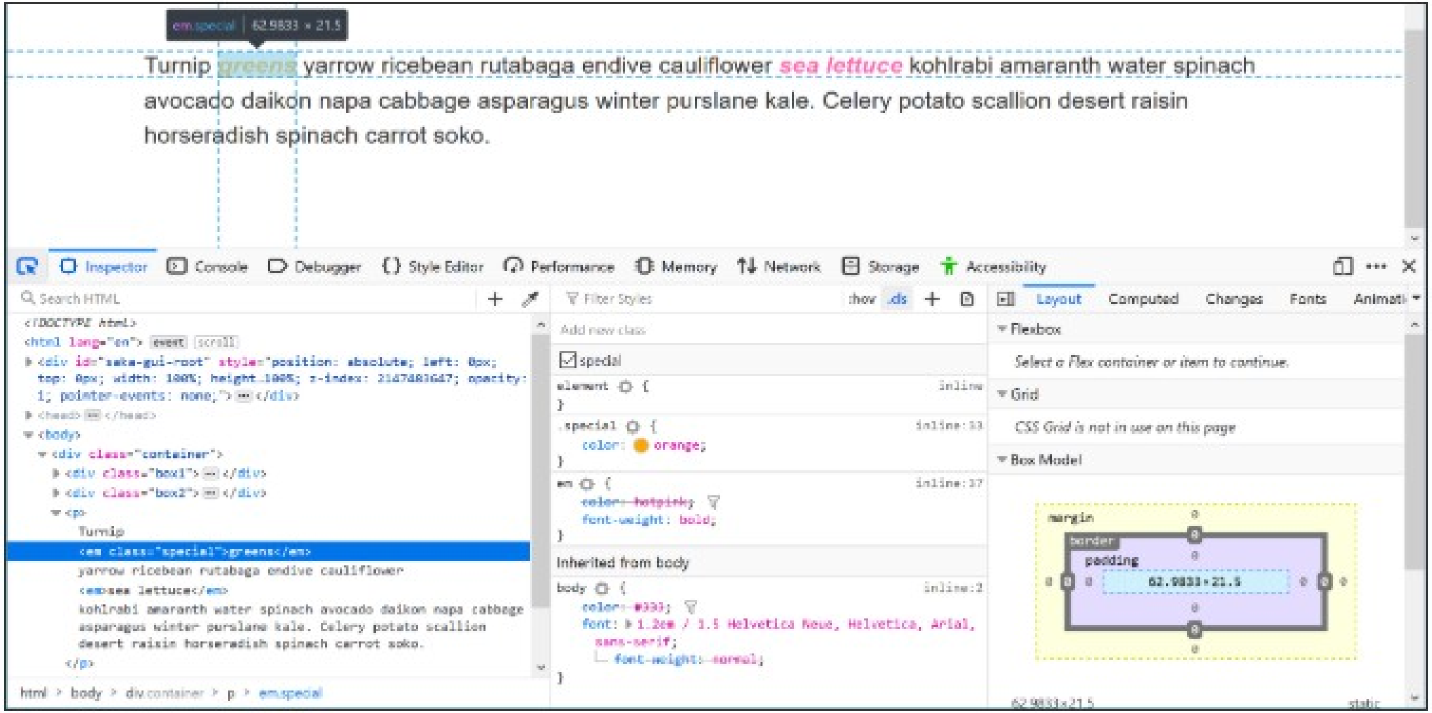
390×240 static

Box Model Properties

box-sizing	content-box	line-height	28.8px
display	block	position	static
float	none	z-index	auto

6. Solving specificity problem

- ➢ We can solve specificity problem with DevTool by inspecting element to see which style is applied to element.



Turnip greens yarrow ricebean rutabaga endive cauliflower sea lettuce kohlrabi amaranth water spinach avocado daikon napa cabbage asparagus winter purslane kale. Celery potato scallion desert raisin horseradish spinach carrot soko.

```
<!DOCTYPE html>
<html lang="en"> event scroll
 <div id="saka-gui-root" style="position: absolute; left: 0px;
  top: 0px; width: 100%; height 100%; z-index: 2147483647; opacity:
  1; pointer-events: none;"> … </div>
 <head> … </head>
 <body>
  <div class="container">
   <div class="box1"> … </div>
   <div class="box2"> … </div>
   <p>
     Turnip
     <em class="special">greens</em>
     yarrow ricebean rutabaga endive cauliflower
     <em>sea lettuce</em>
     kohlrabi amaranth water spinach avocado daikon napa cabbage
     asparagus winter purslane kale. Celery potato scallion
     desert raisin horseradish spinach carrot soko.
   </p>
html > body > div.container > p > em.special
```

```
☑ special
element {                                   inline
}
.special {                                  inline:13
  color: orange;
}
em {                                        inline:17
  ~~color: hotpink;~~
  font-weight: bold;
}
Inherited from body
body {                                      inline:2
  ~~color: #333;~~
  font: 1.2em / 1.5 Helvetica Neue, Helvetica, Arial,
    sans-serif;
    └ ~~font-weight: normal;~~
}
```

7. Debugging problems in CSS

- **Some tips for debugging problems in CSS**
  - Take a step back from the problem
  - Do you have valid HTML and CSS?
  - Is the property and value supported by the browser you are testing in?
  - Is something else overriding your CSS?
  - Make a reduced test case of the problem

Session 6

STYLING TEXT

Overview

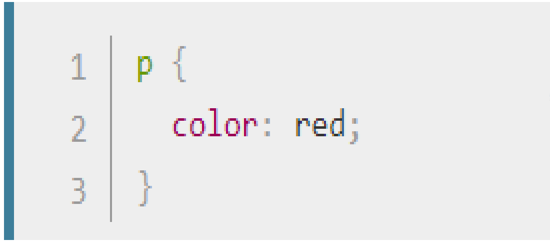
1. **Overview of styling text**
2. **Text and font styling**
3. **Styling lists**
4. **Web fonts**

1. Overview of styling text

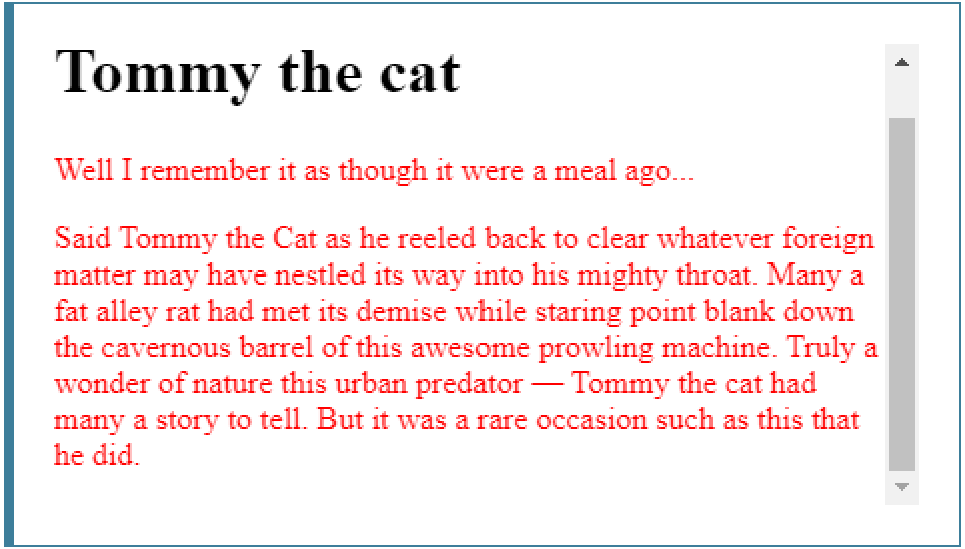
- The CSS properties used to **style text** generally fall into two categories:
  - ***Font styles:*** Properties that affect the font that is applied to the text, affecting what font is applied, how big it is, whether it is bold, italic, etc
  - ***Text layout styles:*** Properties that affect the spacing and other layout features of the text, allowing manipulation of, for example, the space between lines and letters, and how the text is aligned within the content box.

2. Text and font styling

- **Fonts**
  - **Color:** The color property sets the color of the foreground content of the selected elements.



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



Tommy the cat

Well I remember it as though it were a meal ago...

Said Tommy the Cat as he reeled back to clear whatever foreign matter may have nestled its way into his mighty throat. Many a fat alley rat had met its demise while staring point blank down the cavernous barrel of this awesome prowling machine. Truly a wonder of nature this urban predator — Tommy the cat had many a story to tell. But it was a rare occasion such as this that he did.

2. Text and font styling

- **Fonts**
  - **Font families:** The font-family property specifies the font for an element.
    - The **font-family** property can hold several font names as a "fallback" system. If the browser does not support the first font, it tries the next font.
    - There are two types of font family names:
      - ***family-name*** - The name of a font-family, like "times", "courier", "arial", etc.
      - ***generic-family*** - The name of a generic-family, like "serif", "sans-serif", "cursive", "fantasy", "monospace".

2. Text and font styling

- **Fonts**
  - **font-size:** The font-size property sets the size of a font.
  - **font-style**: The font-style property specifies the font style for a text
  - **font-weight**: The font-weight property sets how thick or thin characters in text should be displayed.
  - **text-transform**: The text-transform property controls the capitalization of text.
  - **text-decoration**: The text-decoration property specifies the decoration added to text.

2. Text and font styling

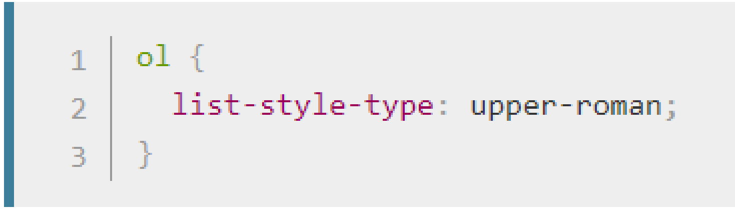
- **Fonts**
  - **Font size:** The font-size property sets the size of a font.
    - **Font size** can take values measured in most of these units (and others, such as percentages), however the most common units you'll use to size text are **px**, **rem**, **em**
    - There are two types of font family names:
      - ***family-name*** - The name of a font-family, like "times", "courier", "arial", etc.
      - ***generic-family*** - The name of a generic-family, like "serif", "sans-serif", "cursive", "fantasy", "monospace".

2. Text and font styling

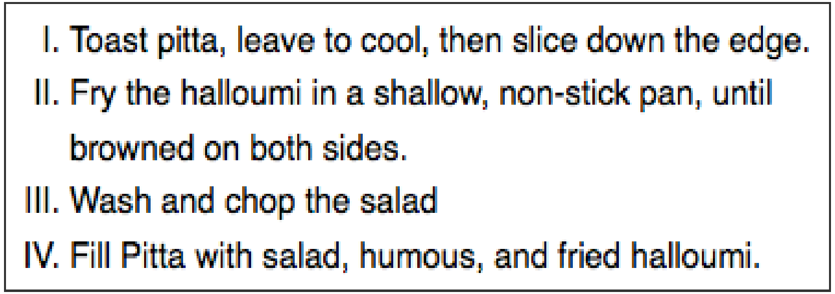
- **Text layout**
  - **Text alignment:** The text-align property is used to control how text is aligned within its containing content box *(left, right, center, justify).*
  - **Light height:** The line-height property specifies the height of a line.
  - **Letter and word spacing:** The letter-spacing and word-spacing properties allow you to set the spacing between letters and words in your text.

3. Styling list

- **Bullet styles:** The list-style-type property specifies the type of list item marker.



```
ol {
  list-style-type: upper-roman;
}
```



I. Toast pitta, leave to cool, then slice down the edge.
II. Fry the halloumi in a shallow, non-stick pan, until browned on both sides.
III. Wash and chop the salad
IV. Fill Pitta with salad, humous, and fried halloumi.

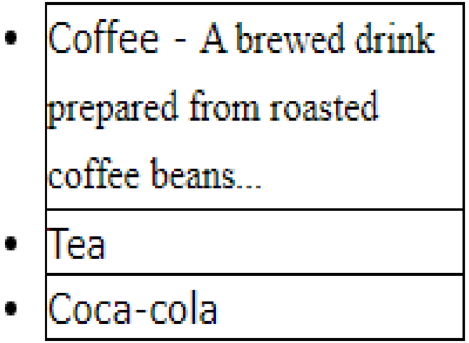
3. Styling list

- **Bullet styles:** some value of list-style-type property
  - disc
  - circle
  - square
  - decimal
  - decimal-leading-zero
  - lower-roman
  - lower-greek
  - ...

3. Styling list

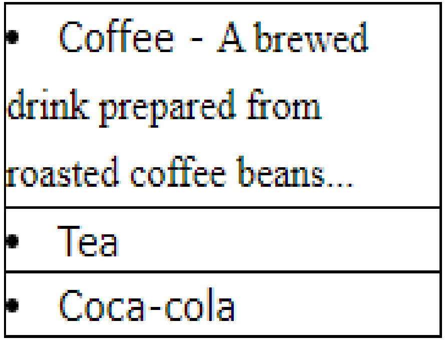
- **Bullet position:** The list-style-position property specifies the position of the list-item markers.

  - list-style-position: out



- Coffee - A brewed drink prepared from roasted coffee beans...
- Tea
- Coca-cola

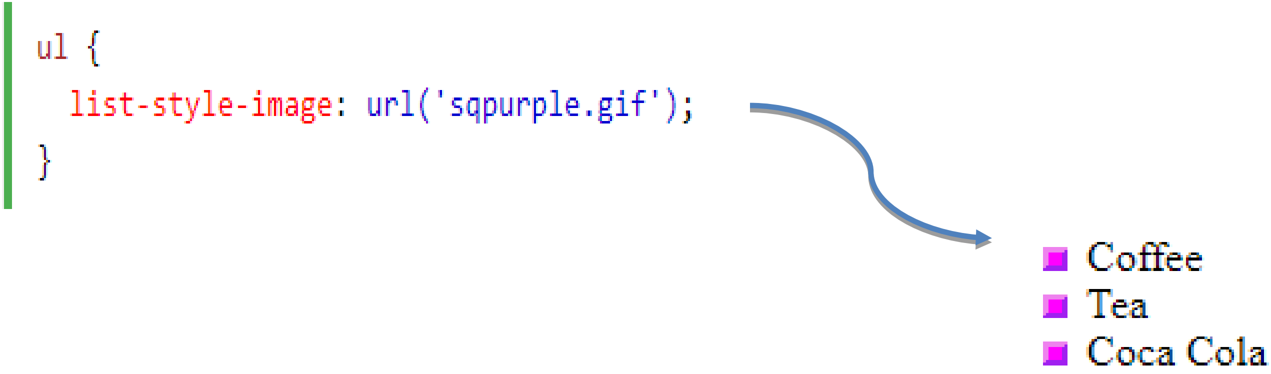
  - list-style-position: inside



- Coffee - A brewed drink prepared from roasted coffee beans...
- Tea
- Coca-cola

3. Styling list

- **Using a custom bullet image:** The list-style-image property specifies an image as the list item marker

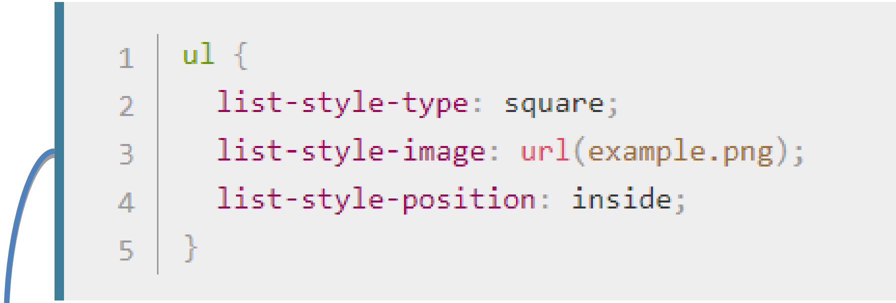


```
ul {
  list-style-image: url('sqpurple.gif');
}
```

- Coffee
- Tea
- Coca Cola

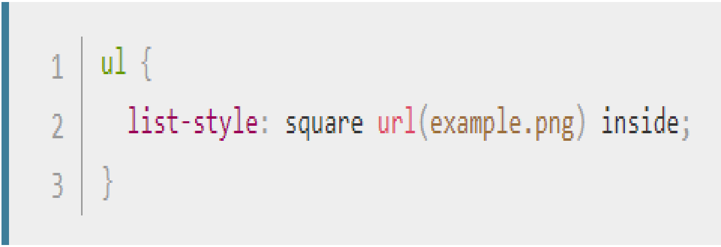
3. Styling list

- **list-style shorthand:** The three properties mentioned above can all be set using a single shorthand property, list-style



```css
ul {
  list-style-type: square;
  list-style-image: url(example.png);
  list-style-position: inside;
}
```

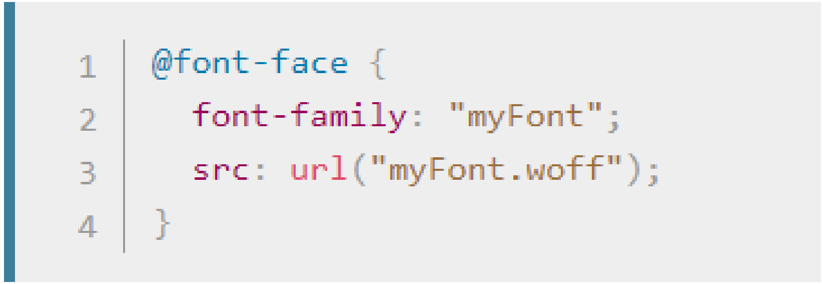
Could be replaced by this



```css
ul {
  list-style: square url(example.png) inside;
}
```

4. Web font

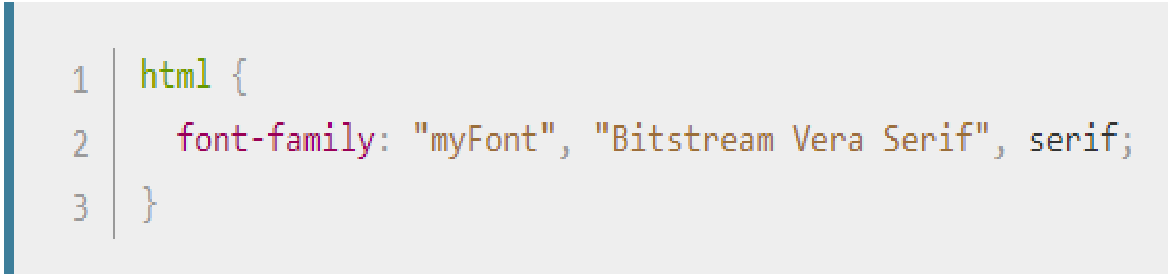
- **Web fonts** are a CSS feature that allows you to specify font files to be downloaded along with your website as it is accessed.
- **@font-face:** First of all, you have a @font-face block at the start of the CSS, which specifies the font file(s) to download:



```css
@font-face {
  font-family: "myFont";
  src: url("myFont.woff");
}
```

4. Web font

- **@font-face:** To use the font for an HTML element, refer to the name of the font (myFirstFont) through the font-family property



```
html {
  font-family: "myFont", "Bitstream Vera Serif", serif;
}
```

4. Web font

- **Steps to use a web font**
    1. Finding fonts
    2. Generating the required code
    3. Implementing the code in your demo

Thank you