

Height and Width of the Box

In this lesson, we will look at setting up the height and width of the box.
Let's begin!

WE'LL COVER THE FOLLOWING



- Listing 10-10: Explicit box size

Normally, the browser calculates the width of a box from the width of its container. From the width of the box, the content width is calculated by subtracting the size of these six elements: the left margin, the left border, the left padding, the right padding, the right border, and finally the right margin. For all direct children of the `<body>` tag, the width of the container is the browser window's width.

Let's make an example calculation. Assume, we have an `<h1>` tag nested directly into `<body>` and the style sheet defines this rule:

```
h1 {  
  margin: 20px;  
  border-left: 8px;  
  border-right: 16px;  
  padding: 12px;  
}
```



If the browser window's width is 1000 *pixels*, the width of the `<h1>` tag is $1000 - 20 - 8 - 12 - 12 - 16 - 20 = 912$ *pixels*.

The height of the box is calculated from the height of the content by adding to it the sizes of the following six elements: the top margin, the top border, the top padding, the bottom padding, the bottom border, and finally, the bottom margin.

Now, let's make a calculation for the same `<h1>` tag we used before, provided, the browser window's width is 1000 *pixels*, the height of the `<h1>` tag's

content is 80 pixels , and the style sheet defines this rule:

```
h1 {  
  margin: 5%;  
  border: 8px;  
  padding: 12px;  
}
```



Because the margin is defined in percentages, the real size is the 5% of the browser window's width, namely 50 pixels . The height of the `<h1>` box is $80 + 50 + 8 + 12 + 12 + 8 + 50 = 220\text{ pixels}$.

If you need to, you can specify the size of the box explicitly with the `height` and `width` properties, as shown in Listing 10-10.

 **NOTE:** You cannot control the size of inline elements.

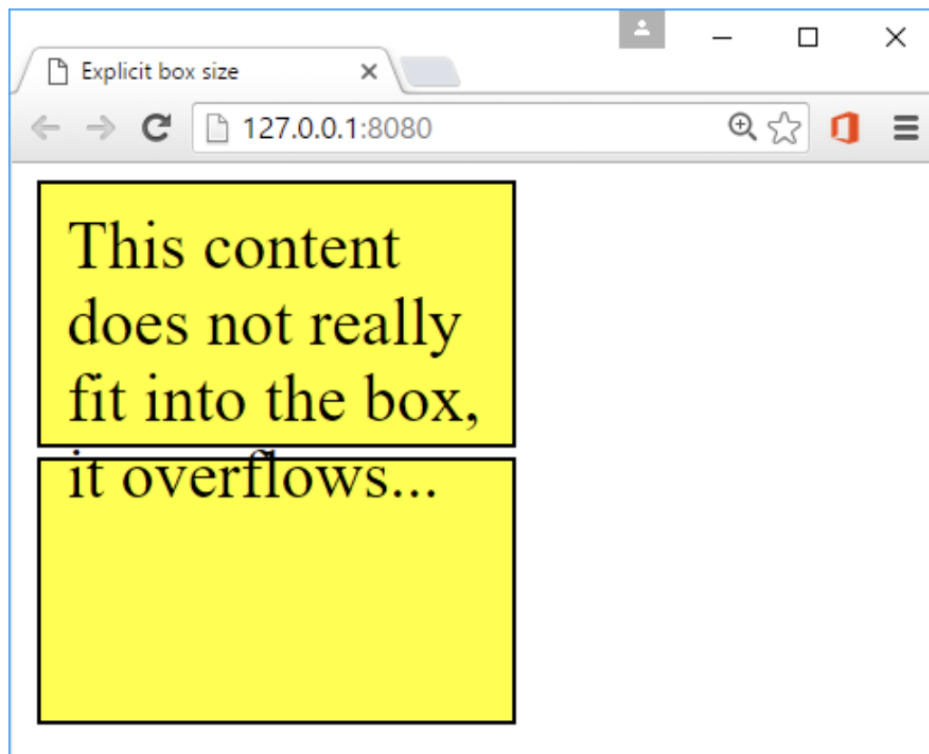
Listing 10-10: Explicit box size

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>Explicit box size</title>  
  <style>  
    .sizedbox {  
      width: 200px;  
      height: 100px;  
      background-color: yellow;  
      border: 2px solid black;  
      padding: 12px;  
      margin: 4px;  
      font-size: 2em;  
    }  
  </style>  
</head>  
<body>  
  <div class="sizedbox">  
    This content does not really  
    fit into the box, it overflows...  
  </div>  
  <div class="sizedbox"></div>  
</body>  
</html>
```

This simple markup uses two `<div>` tags and set their contents to 200 pixels wide 100 pixels tall. According to the margin, border, and padding settings, each box around its content is 236 pixels wide and 136 pixels tall.

The image below uncovers an issue related to the explicit size: the content of the first box is too big, so it overflows the box.

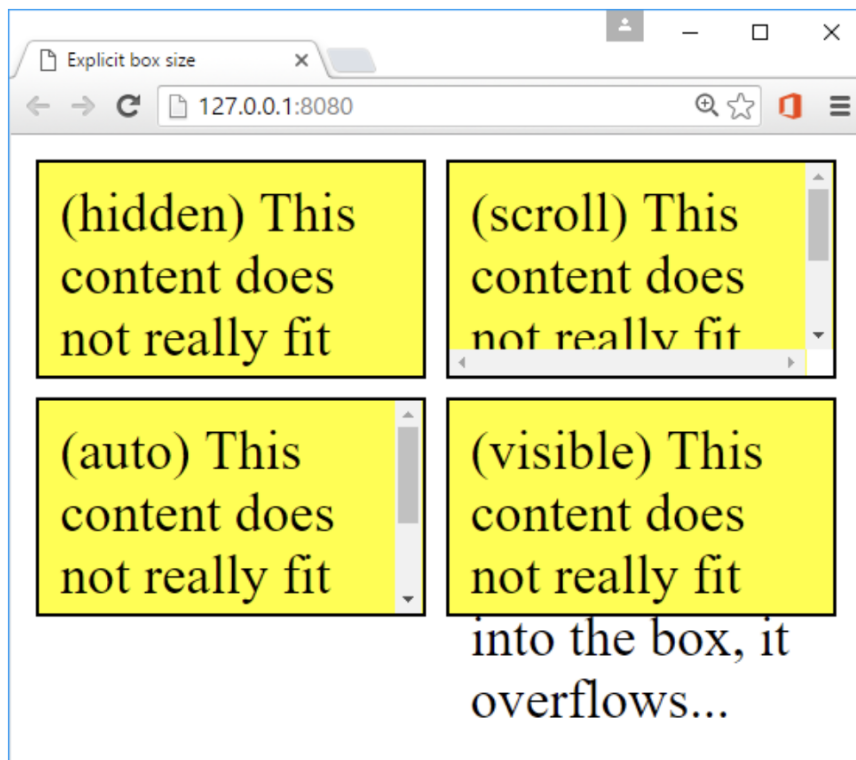
This is a common issue. Normally, the browser adjusts the size (generally the height) of the box so that the content can fit within the box. However, when you set the box size explicitly, you lose this convenient feature.



Explicit box sizing: the content overflows

Fortunately, with the `overflow` property, you can control what a browser should do in this situation. Specify the visible value to let the browser do its normal rendering. Set the property value to `hidden` to hide any content outside of the box. Using the `scroll` value, the box gets a scrollbar and you'll be able to scroll the content within the box. The `auto` value checks if the content fits into the box, and automatically uses the `scroll` value, if it overflows.

The image below demonstrates these settings.



Using the overflow property

NOTE: You can find the source code listing in the Exercise-10-14 folder.

In many cases, you need more flexibility between letting the browser size the box or set the box dimensions explicitly. You can add this flexibility by setting minimum and maximum heights and widths with the `min-height`, `max-height`, `min-width`, and `max-width` properties. These settings let the browser size the box between the size limits you specify.

In the *next lesson*, we will box our inline elements.

Stay tuned!