Hareesh N Nampoothiri
Research Scholar, University of Kerala, Thiruvananthapuram

Distance Units in CSS3

CSS3 is the new standard for cascading style sheets recommended by W3C. CSS3 comes with a lot of exciting features which enable the designers to extend the possibilities of web design to further levels. Websites today are delivered across a variety of devices which supports all possible resolutions. The newly introduced distance units in CSS3 are a blessing for the designers which allows them to design adaptive websites without much effort.

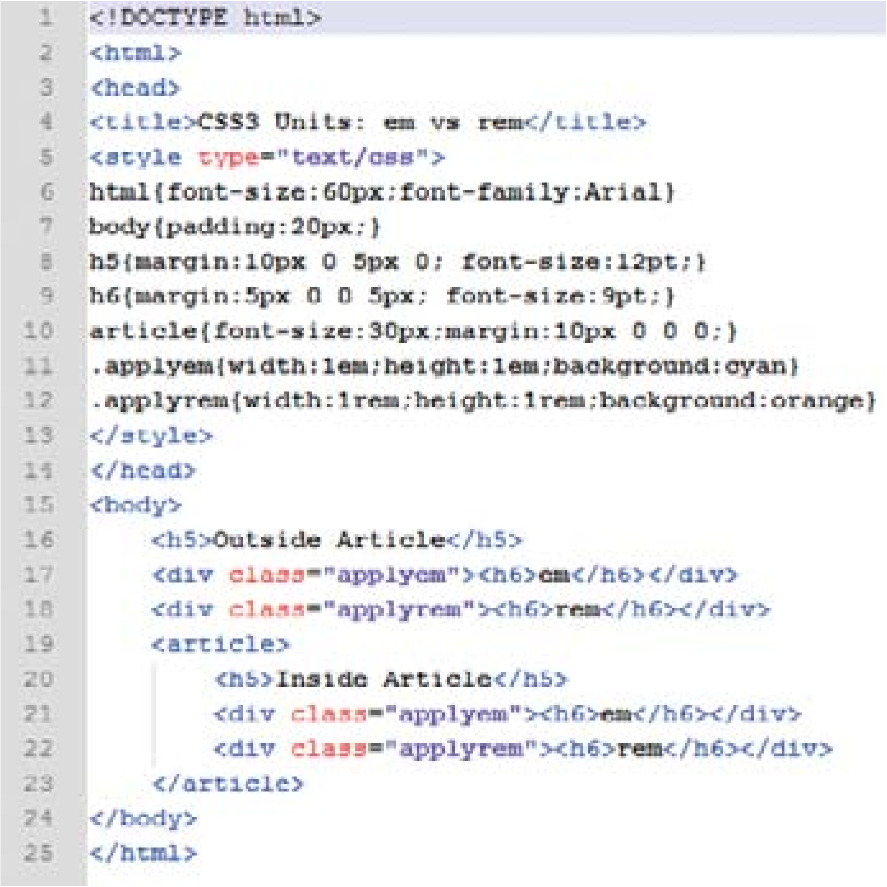
Distance units in CSS are used to define the lengths (*in the form of width, height, size, thickness etc.*) of various elements. In CSS, the length value is defined by a number (*with or without a decimal point*) followed by a unit identifier such as `px`, `em` etc. Two types of length units are used in CSS; absolute lengths and relative lengths.

The absolute length units are fixed in relation to each other and are anchored to some physical measurement. These units are widely used in websites using fixed width layouts. Units such as `mm`, `cm`, `in`, `pt`, `pc` and `px` fall under this category.

Unit	Definition
mm	millimeters
cm	centimeters
in	inches (1in = 2.54cm)
px	pixels; (1px = 1/96th of 1in)
pt	points; (1pt = 1/72nd of 1 in)
pc	picas; (1pc = 12pt)

Relative length units specify a length relative to another length property. Style sheets that use relative units can more easily scale for different resolutions and they are often used in variable width (*fluid*) layouts.

Until CSS3, only the first two relative length units (`em` *and* `ex`) were available. The last five relative length units (`ch`, `rem`, `vw`, `vh`, *and* `vmin`) are the new units that are included in CSS3. The units `ch` and `rem` are font relative length units, where as `vw`, `vh`, and `vmin` are viewport relative length units. It is also possible to define lengths in percentage values.



```
<!DOCTYPE html>
<html>
<head>
<title>CSS3 Units: em vs rem</title>
<style type="text/css">
html{font-size:60px;font-family:Arial}
body{padding:20px;}
h5{margin:10px 0 5px 0; font-size:12pt;}
h6{margin:5px 0 0 5px; font-size:9pt;}
article{font-size:30px;margin:10px 0 0 0;}
.applyem{width:1em;height:1em;background:cyan}
.applyrem{width:1rem;height:1rem;background:orange}
</style>
</head>
<body>
    <h5>Outside Article</h5>
    <div class="applyem"><h6>em</h6></div>
    <div class="applyrem"><h6>rem</h6></div>
    <article>
        <h5>Inside Article</h5>
        <div class="applyem"><h6>em</h6></div>
        <div class="applyrem"><h6>rem</h6></div>
    </article>
</body>
</html>
```

Fig. 1: HTML page showing the difference between em and rem

Unit	Definition
em	Relative to the font size of the element.
ex	Relative to the x-height of the element's font.
rem	Relative to the font size of the root element.
ch	Relative to the width of the "0" glyph in the element's font.
vw	Relative to the viewport's width.
vh	Relative to the viewport's height.
vmin	Relative to the minimum of the viewport's height and width.

Percentage values are always related to another value, a length for example. In CSS recommendations, percentage values are considered as a number type. Hence, we are not including it in the list of relative length units.

Font relative length units

The `rem` unit

The `rem` unit can be related to the `em` unit which is already available in CSS2. The `em` unit is equal to the computed value of the `font-size` property of the element in which it is used. The `rem` uses the computed value of the `font-size` of the root element. It can be defined in the style

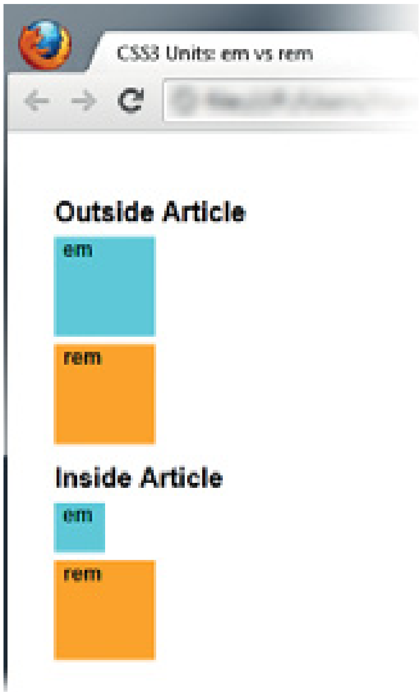


Fig. 2: The page coded in Fig. 1 as rendered in a browser

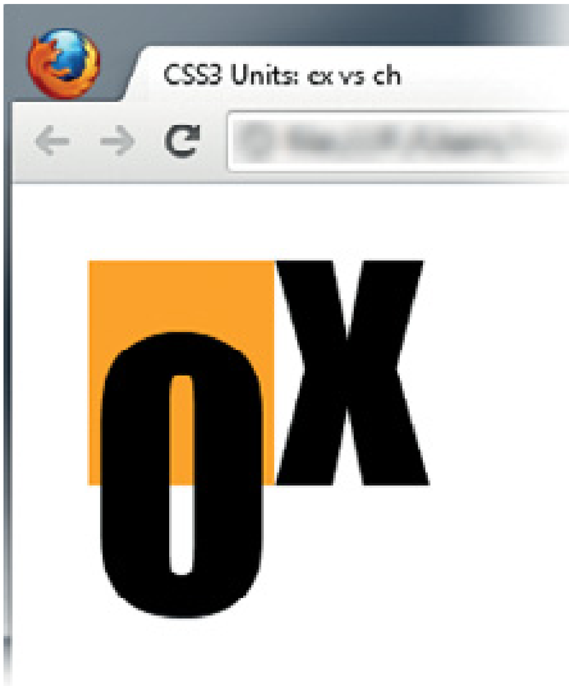
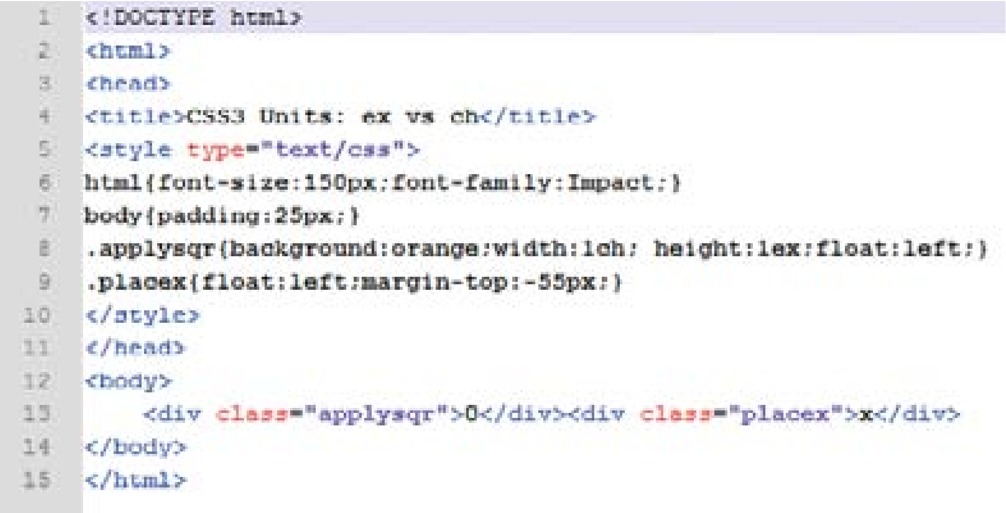


Fig. 4: The page coded in Fig. 3 as rendered in a browser

definition for the `html` tag or using the :root selector.

Fig. 1 shows a sample HTML code which shows the difference in usage of `em` and `rem`. The `font-size` of the root element is defined as `60px`. The width and height of the `<div>` elements is defined by two classes `.applyem` and `.applyrem`. The former sets the width and height of the `<div>` element to `1em` and the latter sets these properties to `1rem`. The width and height of both the `<div>` elements outside the `<article>` element is the same as both takes the computed value of the `font-size` property of the root element. But when both are applied inside `<article>` element it produces a different result. The `font-size` property of the `<article>` element is defined to `30px`. The width and height of the `<div>` element which takes the `.applyem` class definition will be `30px`. But the width and height of the `<div>` element which takes the `.applyrem` class definition remains the same, ie. `60px`. Thus the advantage of using the `rem` instead of `em` is that, when we use nested elements, the value of the length units do not change unintentionally.



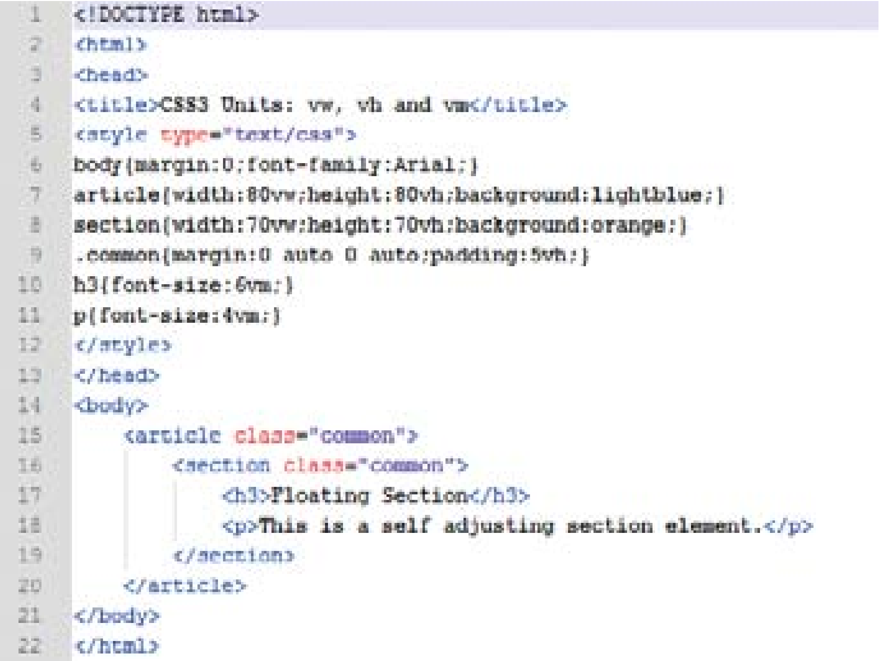
```
<!DOCTYPE html>
<html>
<head>
<title>CSS3 Units: ex vs ch</title>
<style type="text/css">
html{font-size:150px;font-family:Impact;}
body{padding:25px;}
.applysqr{background:orange;width:1ch; height:1ex;float:left;}
.placex{float:left;margin-top:-55px;}
</style>
</head>
<body>
    <div class="applysqr">0</div><div class="placex">x</div>
</body>
</html>
```

Fig. 3: HTML page showing the difference between ex and ch

The `ch` unit

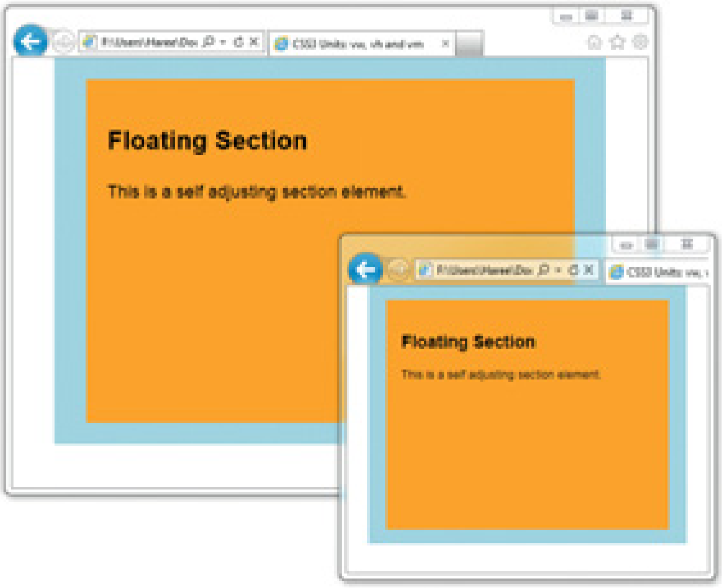
The `ch` unit can be related to the `ex` unit which is also available in CSS2. The `ex` unit takes the value of the height of character 'x' of the current font. The `ch` unit takes the value of the width of character '0' (zero) of the current font. In the example given in Fig. 3 we use Impact as the page font.

The `.applysqr` class defines the height and width of the `<div>` element as `1ch` and `1ex` respectively. Thus the height of the `<div>` section will be the height of the 'x' character and the width will be the width of the '0' character of the font. The browser output is given in Fig. 4.



```
<!DOCTYPE html>
<html>
<head>
<title>CSS3 Units: vw, vh and vm</title>
<style type="text/css">
body{margin:0;font-family:Arial;}
article{width:80vw;height:80vh;background:lightblue;}
section{width:70vw;height:70vh;background:orange;}
.common{margin:0 auto 0 auto;padding:5vh;}
h3{font-size:6vm;}
p{font-size:4vm;}
</style>
</head>
<body>
    <article class="common">
        <section class="common">
            <h3>Floating Section</h3>
            <p>This is a self adjusting section element.</p>
        </section>
    </article>
</body>
</html>
```

Fig. 5: HTML page showing the usage of vw, vh, and vm



Floating Section

This is a self adjusting section element.

Floating Section

This is a self adjusting section element.

Fig. 6: The page coded in Fig. 5, opened in two browser windows of different sizes

Viewport relative length units

The root viewport size is the width and height of the viewable area within the browser where the page is displayed. The units `vw`, `vh`, and `vmin` come under this category. All the three units are similar in nature. The vw unit stands for the viewport width and `1vw` is equal to 1% of the current viewport size. Similarly, `vh` represents the viewport height.

The `vmin` unit takes the value of `vw` or `vh`, whichever is smaller. When the height or width of the viewport is changed (by changing the browser screen size), the elements using viewport relative length units are scaled accordingly. (Fig. 6) Even though in W3C working draft the minimum value unit is given as `vmin`, we may use `vm` instead, as the modern browsers support `vm` instead of `vmin`.

***Note:** The new length units included in CSS3 working draft is not supported by all browsers. The `rem` unit is supported by latest versions of Webkit, Gecko, Trident, and Presto based browsers whereas `ch` is supported by only Gecko and Trident based browsers. The `vw`, `vh`, and `vm` are only supported in latest versions of Trident based browsers. Recent versions of Webkit based browsers support `vw` and `vh` but not `vm`.*

References

[1] *CSS Values and Units Module Level 3* (dated 08 March 2012). W3C Working Draft (CSS3). Retrieved 2012 June 26, from [http://www.w3.org/TR/css3-values/#lengths].

[2] *Syntax and basic data types* (n.d). W3C Recommendation (CSS2). Retrieved 2012 June 26, from [http://www.w3.org/TR/CSS2/syndata.html#values].

About the Author

Hareesh N Nampoothiri is a visual design consultant with an experience of more than a decade and worked with government organizations like C-DIT, C-DAC, University of Kerala and other private organizations. Currently, he is doing interdisciplinary research in ethnic elements in visual design in computer media. He is an author of two books on graphic design and a regular columnist in leading technology magazines including CSI Communications. Kathakli, blogging, and photography are his passions. He has directed a documentary feature on Kathakali and also directed an educational video production for IGNOU, New Delhi.