**Muhammad Abdullah**

**SE(6A) | 19F-0916**

Web Engineering Couse

Assignment 2: Difference between absolute and relative units

**Difference Between Absolute and Relative Units in HTML**

Introduction:

Before proceeding to the difference between absolute and relative units, let us try to understand what units are and what is their role in HTML.

HTML/CSS has different units that are used to determine the length between different elements. These units allow us to play with the length between elements, changing those lengths, putting spaces in-between elements, and many more. It can be of many types like margins, paddings, width, font sizes, box sizes, etc. These different units are expressed in terms of units like px for Pixels, cm for centimeters, etc. In general, there are 2 types of units in HTML/CSS.

* Relative Units.
* Absolute Units.

**Absolute Units:**

As the name depicts, absolute units are of fixed length and they are rendered the same as the amount given to them, irrespective of any other unit of length. As they are fixed, they are not welcomed to be used to screens as HTML sites vary from device to device. Although, they can be used for a fixed size of output mediums like printing pages.

The following table shows different Absolute Units and how to use them for lengths:

|  |  |  |
| --- | --- | --- |
| Units | Description | Use Examples |
| cm | Used for centimeters | font-size: 2.5cm |
| mm | Used for millimeters | font-size: 7.5mm |
| in | Used for inches | font-size: 1in |
| pt | Used for points | font-size: 30pt |
| pc | Used for picas | font-size: 15pc |
| Px\* | Used for pixels | font-size: 50px |

**Relative Units:**

Relative units specify a length according to another length property. These units are based upon the length units of other properties. These units scale better in-between rendering elements. They are easy to use and manage because they handle themselves according to the screen size of the device on which it is being shown.

The following table shows different Relative Units and how to use them for lengths:

|  |  |  |
| --- | --- | --- |
| Units | Description | Use Examples |
| em | Relative to Font Size | font-size: 13em |
| ex | Relative to Height of the Font | font-size: 15ex |
| ch | Relative to Width of 0 (Zero) | font-size: 17ch |
| rem | Relative to Font-Size of Root Element | font-size: 27rem |
| vm | Relative to 1% of Width of viewport | font-size: 25vm |
| vh | Relative to 1% of Height of viewport | font-size: 54vh |
| vmin | Relative to 1% of viewport’s smaller dimension. | font-size: 34vmin |
| vmax | Relative to 1% of viewport’s larger dimension. | font-size: 53vmax |
| % | Relative to Parent Element | font-size: 36% |

**Conclusion:**

If you want to use length units for output resources or similar types of screens, then you can go with **absolute measures**. But if you want to have a responsive website for different devices having different screen ratios, then you have to choose **relative units** otherwise absolute units will result in a damaged website, full of indentation mistakes.