For just a normal tag directly tag name and you can go

For id #

For class .

Like

P {

Background-color : blue ;

}

#<the id> {

Background-color : blue ;

}

.<the class name>{

Background-color : blue ;

}

To cascade it

H1, h2 , h3 {

Background-color : blue ;

}

That means its applied to all the three

Body h1 {

Background-color : blue ;

}

It means that the h1 tag which is in the body tag

@ has been around since the days of @import in CSS1, although it's arguably becoming increasingly common in the recent @media (CSS2, CSS3) and @font-face (CSS3) constructs. The @ syntax itself, though, as I mentioned, is not new.

These are all known in CSS as **at-rules**. They're special instructions for the browser, not directly related to styling of (X)HTML/XML elements in Web documents using rules and properties, although they do play important roles in controlling how styles are applied.

Some code examples:

/\* Import another stylesheet from within a stylesheet \*/

@import url(style2.css);

/\* Apply this style only for printing \*/

@media print {

body {

color: #000;

background: #fff;

}

}

/\* Embed a custom web font \*/

@font-face {

font-family: 'DejaVu Sans';

src: local('DejaVu Sans Regular'), url(/fonts/DejaVuSans.ttf);

}

* [@font-face rules](http://www.w3.org/TR/css3-fonts/#font-face-rule) define custom fonts for use in your designs that aren't always available on all computers, so a browser downloads a font from the server and sets text in that custom font as if the user's computer had the font.
* [@media rules](http://www.w3.org/TR/css3-conditional/#at-media), in conjunction with [media queries](http://www.w3.org/TR/css3-mediaqueries) (formerly only [media types](http://www.w3.org/TR/CSS21/media.html)), control which styles are applied and which aren't based on what media the page is being displayed in. In my code example, only when printing a document should all text be set in black against a white (the paper) background. You can use media queries to filter out print media, mobile devices and so on, and style pages differently for those.

In css3, attribute selector can also be done, like:-

P[name]{color:blue;}// that paragraph where attribute name is used

Or even more specific can be used:-

P[name=”FirstName”]{color:red;}// here every paragraph which has attribute name as FIrstName would get this css

In that also more specific can be done…using ‘^’ , ‘$’ or ‘\*’ operator… ^for the matching string in starting of the actual one… $ for the ending and \* for anywhere in the string in the actual one, example:-

P[name^=”Firstname”]{color:blue;} // ehre every paragraph whose attribute name starts with FirstName… e.x. :-

<p name=”FirstNameTest”></p> // this will be selected…likewise

We can do css of same class, i.e. pseudo class concept it is…:-

Something like this can be used to have any particular style according to n, e.x.:-

P:nth-child(3){color:blue;}// so here 3rd one would be in this style, not pretty much useful though

P:nth-child(odd)(color:blue;) // every odd element would be applied this style

P:nth-child(3n+1){color:blue;} // every element acc. To this formula would be styled

Similarly any formula can be applied…also the thing is to be written like this only… as In

<element name>:nth-child(<condition>)(<style>)

For negation pseudo class:-

<element tag may be applied>:not(<class name like .bukky>){color:blue;}

Okay for specifying that the element Is preceded by the same element we can do is:-

p.bukky + p{color:blue;}// this applies on all the p of class bukky which are immediately after a p

p.bukky ~ p{color:blue;} // this applies on all the p of class bukky which are after a p, but not necessarily immediately after it

parent child relationship , as :-

div > p {clor : blue;}// applie to those p who are child of div…\*note\* : this does not get applied to p which is child of some section or some other tag , and which is child of div… since then it wont be parent child relationship… it would be grandparent child relationship

div p {color: red;} // this will be applied to any p which is anywhere inside a div… not neccesarrily immediate

in the main wrapper class always mention width…1000px is the best option…

use float like float : left and also give width likewise..

clear : both; // for clearing all the float used above the tag…and i.e. back to normal

\*{..} // for overall website things such as margin padding

You should be able to utilize the asterisk and !important elements within CSS.

html \*

{

font-size: 1em !important;

color: #000 !important;

font-family: Arial !important;

}

The asterisk matches everything (you could probably get away without the html too).

The !important ensures that **nothing can override what you've set in this style** (unless it is also important). (this is to help with your requirement that it should "ignore inner formatting of text" - which I took to mean that other styles could not overwrite these)