1) What is CSS?

CSS stands for Cascading Style Sheet. It is a popular styling language which is used with HTML to design websites. It can also be used with any XML documents including plain XML, SVG, and XUL.

2) What is the origin of CSS?

SGML (Standard Generalized Markup Language) is the origin of CSS. It is a language that defines markup languages.

3) What are the different variations of CSS?

Following are the different variations of CSS:

* CSS1
* CSS2
* CSS2.1
* CSS3
* CSS4

4) How can you integrate CSS on a web page?

There are three methods to integrate CSS on web pages.

1. Inline method - It is used to insert style sheets in HTML document
2. Embedded/Internal method - It is used to add a unique style to a single document
3. Linked/Imported/External method - It is used when you want to make changes on multiple pages.

5) What are the advantages of CSS?

* Bandwidth
* Site-wide consistency
* Page reformatting
* Accessibility
* Content separated from presentation

6) What are the limitations of CSS?

* Ascending by selectors is not possible
* Limitations of vertical control
* No expressions
* No column declaration
* Pseudo-class not controlled by dynamic behavior
* Rules, styles, targeting specific text not possible

7) What are the CSS frameworks?

CSS frameworks are the preplanned libraries which make easy and more standard compliant web page styling. The frequently used CSS frameworks are: -

* Bootstrap
* Foundation
* Semantic UI
* Gumby
* Ulkit

8) Why background and color are the separate properties if they should always be set together?

There are two reasons behind this:

* It enhances the legibility of style sheets. The background property is a complex property in CSS, and if it is combined with color, the complexity will further increase.
* Color is an inherited property while the background is not. So this can make confusion further.

9) What is Embedded Style Sheet?

An Embedded style sheet is a CSS style specification method used with HTML. You can embed the entire stylesheet in an HTML document by using the STYLE element.

1. **<style>**
2. body {
3. background-color: linen;
4. }
5. h1 {
6. color: red;
7. margin-left: 80px;
8. }
9. **</style>**

10) What are the advantages of Embedded Style Sheets?

* You can create classes for use on multiple tag types in the document.
* You can use selector and grouping methods to apply styles in complex situations.
* No extra download is required to import the information.

11) What is a CSS selector?

It is a string that identifies the elements to which a particular declaration apply. It is also referred as a link between the HTML document and the style sheet. It is equivalent of HTML elements. There are several different types of selectors in CSS: -

* CSS Element Selector
* CSS Id Selector
* CSS Class Selector
* CSS Universal Selector
* CSS Group Selector

12) Name some CSS style components.

Some CSS Style components are:

* Selector
* Property
* Value

13) What is the use of CSS Opacity?

The CSS opacity property is used to specify the transparency of an element. In simple word, you can say that it specifies the clarity of the image. In technical terms, Opacity is defined as the degree to which light is allowed to travel through an object. For example:

1. **<style>**
2. img.trans {
3. opacity: 0.4;
4. filter: alpha(opacity=40); /\* For IE8 and earlier \*/
5. }
6. **</style>**

14) Explain universal selector.

The universal selector matches the name of any of the element type instead of selecting elements of a specific type.

1. **<style>**
2. \* {
3. color: green;
4. font-size: 20px;
5. }
6. **</style>**

15) Which command is used for the selection of all the elements of a paragraph?

The p[lang] command is used for selecting all the elements of a paragraph.

16) What is the use of % unit?

It is used for defining percentage values.

17) Name the property used to specify the background color of an element.

The background-color property is used to specify the background color of the element. For example:

1. **<style>**
2. h2,p{
3. background-color: #b0d4de;
4. }
5. **</style>**

18) Name the property for controlling the image repetition of the background.

The background-repeat property repeats the background image horizontally and vertically. Some images are repeated only horizontally or vertically.

1. **<style>**
2. body {
3. background-image: url("paper1.gif");
4. margin-left:100px;
5. }
6. **</style>**

19) Name the property for controlling the image position in the background.

The background-position property is used to define the initial position of the background image. By default, the background image is placed on the top-left of the webpage.

You can set the following positions:

1. center
2. top
3. bottom
4. left
5. right
6. background: white url('good-morning.jpg');
7. background-repeat: no-repeat;
8. background-attachment: fixed;
9. background-position: center;

20) Name the property for controlling the image scroll in the background.

The background-attachment property is used to specify if the background image is fixed or scroll with the rest of the page in the browser window. If you set fixed the background image, then the image not move during scrolling in the browser. Let's take an example with the fixed background image.

1. background: white url('bbb.gif');
2. background-repeat: no-repeat;
3. background-attachment: fixed;

21) What is the use of ruleset?

The ruleset is used to identify that selectors can be attached with other selectors. It has two parts:

* Selector - Selector indicates the HTML element you want to style.
* Declaration Block - The declaration block can contain one or more declarations separated by a semicolon.

22) What is the difference between class selectors and id selectors?

An overall block is given to class selector while id selectors take only a single element differing from other elements.

CSS Class Selector

1. **<style>**
2. .center {
3. text-align: center;
4. color: blue;
5. }
6. **</style>**

CSS Id Selector

1. **<style>**
2. #para1 {
3. text-align: center;
4. color: blue;
5. }
6. **</style>**

23) What are the advantages of External Style Sheets?

* You can create classes for reusing it in many documents.
* By using it, you can control the styles of multiple documents from one file.
* In complex situations, you can use selectors and grouping methods to apply styles.

24) What is the difference between inline, embedded and external style sheets?

**Inline**: Inline Style Sheet is used to style only a small piece of code.

Syntax

1. **<htmltag** style="cssproperty1:value; cssproperty2:value;"**>** **</htmltag>**

**Embedded**: Embedded style sheets are put between the <head>...</head> tags.

Syntax

1. **<style>**
2. body {
3. background-color: linen;
4. }
5. h1 {
6. color: red;
7. margin-left: 80px;
8. }
9. **</style>**

**External**: This is used to apply the style to all the pages within your website by changing just one style sheet.

Syntax

1. **<head>**
2. **<link** rel="stylesheet" type="text/css" href="mystyle.css"**>**
3. **</head>**

25) What is RWD?

RWD stands for Responsive Web Design. This technique is used to display the designed page perfectly on every screen size and device, for example, mobile, tablet, desktop and laptop. You don't need to create a different page for each device.

26) What are the benefits of CSS sprites?

If a web page has a large number of images that take a longer time to load because each image separately sends out an HTTP request. The concept of CSS sprites is used to reduce the loading time for a web page because it combines the various small images into one image. It reduces the number of HTTP requests and hence the loading time.

27) What is the difference between logical tags and physical tags?

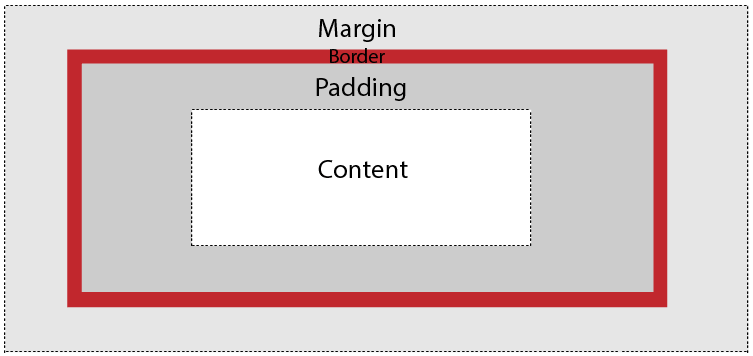
* Physical tags are referred to as presentational markup while logical tags are useless for appearances.
* Physical tags are newer versions, on the other hand, logical tags are old and concentrate on content.

28) What is the CSS Box model and what are its elements?

The CSS box model is used to define the design and layout of elements of CSS.

The elements are:

* Margin - It removes the area around the border. It is transparent.
* Border - It represents the area around the padding
* Padding - It removes the area around the content. It is transparent.
* Content - It represents the content like text, images, etc.



29) What is the float property of CSS?

The CSS float property is used to move the image to the right or left along with the texts to be wrapped around it. It doesn't change the property of the elements used before it.

To understand its purpose and origin, let's take a look at its print display. In the print display, an image is set into the page such that text wraps around it as needed.

Its web layout is also just similar to print layout.

30) How to restore the default property value using CSS?

In short, there is no easy way to restore to default values to whatever a browser uses.

The closest option is to use the 'initial' property value, which restores the default CSS values, rather than the browser's default styles.

31) What is the purpose of the z-index and how is it used?

The z-index helps to specify the stack order of positioned elements that may overlap one another. The z-index default value is zero and can take on either a positive or negative number.

An element with a higher z-index is always stacked above than a lower index.

Z-Index can take the following values:

* **Auto:** Sets the stack order equal to its parents.
* **Number:** Orders the stack order.
* **Initial:** Sets this property to its default value (0).
* **Inherit:** Inherits this property from its parent element.

32) Explain the difference between visibility: hidden and display: none?

**visibility: hidden** hides the element, but it occupies space and affects the layout of the document.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. h1.vis {
6. visibility: visible;
7. }
9. h1.hid {
10. visibility: hidden;
11. }
12. **</style>**
13. **</head>**
14. **<body>**
15. **<h1** class="vis"**>**It is visible**</h1>**
16. **<h1** class="hid"**>**It is hidden**</h1>**
18. **<p>**Note - Second heading is hidden, but it still occupy space.**</p>**
19. **</body>**
20. **</html>**

**display: none** also hides the element but not occupy space. It will not affect the layout of the document.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. h1.vis {
6. display: block;
7. }
9. h1.hid {
10. display: none;
11. }
12. **</style>**
13. **</head>**
14. **<body>**
15. **<h1** class="vis"**>**It is visible**</h1>**
16. **<h1** class="hid"**>**It is hidden**</h1>**
18. **<p>**Note - Second heading is hidden and not occupy space.**</p>**
19. **</body>**
20. **</html>**

33) What do you understand by W3C?

W3C stands for World Wide Web Consortium. Its purpose is to deliver the information of the World Wide Web. It also develops rules and guidelines for the Web.

34) What is tweening?

It is the process of generating intermediate frames between two images.

It gives the impression that the first image has smoothly evolved into the second one.

It is an important method used in all types of animations.

In CSS3, Transforms (matrix, translate, rotate, scale) module can be used to achieve tweening.

35) What is the difference between CSS2 and CSS3?

The main difference between CSS2 and CSS3 is that CSS3 is divided into different sections which are also known as modules. Unlike CSS2, CSS3 modules are supported by many browsers.

## 1: Display

Display takes on many different values, but only 4 are most commonly used.

div {

display: block;

display: inline-block;

display: inline;

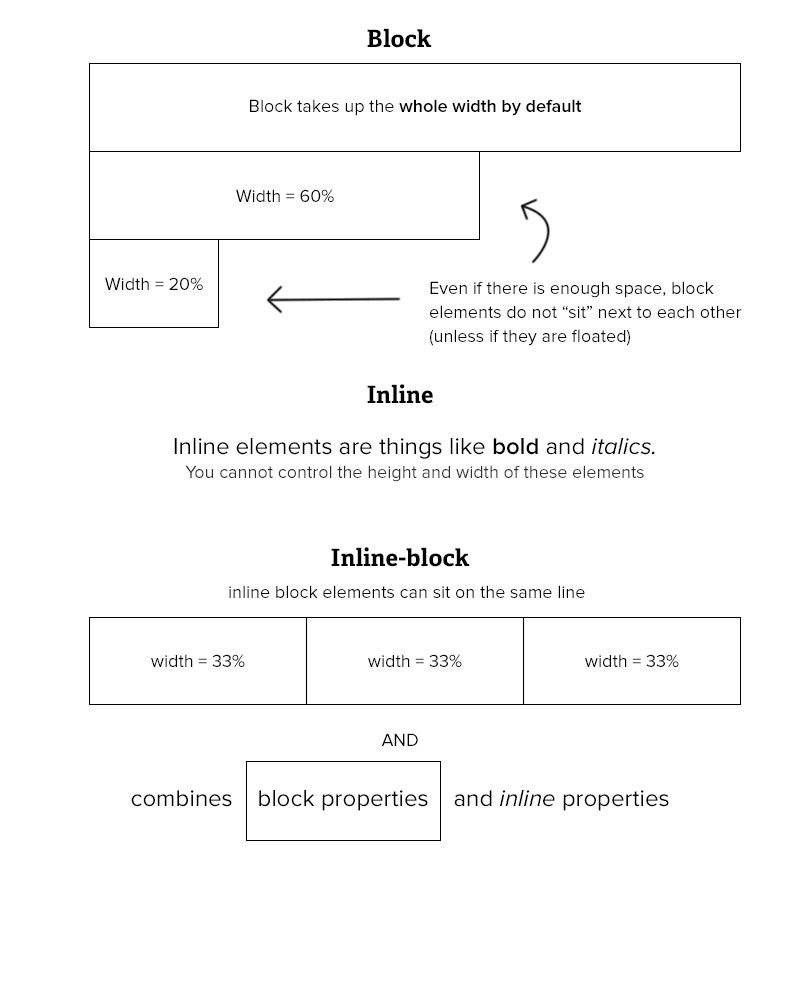
display: none;

}

* **block:** Many HTML elements are set to this mode of display by browsers’ stylesheets. They include <div>, <ul> and text blocks like <p>. Block level elements by default take up as much space as they can, and they cannot be placed on the same horizontal line with any other display modes, include other block elements. (Exception: unless they are floated)

With block elements, you gain the ability to alter the element’s width and height to your liking, which is why they are [used for layouts](http://zellwk.com/blog/layouts-of-a-website/)

* **inline:** The inline mode wraps many HTML elements tighty around them and is the defaults for all elements that are not specified with any other display values. Elements can be placed side by side on the same line as inline elements. Think about the <strong> tag that **bolds elements**, the <em> tag that creates italics and <a>tags that allow you to link to other web pages. These are all examples of inline elements. You will not be able to change an inline element’s width and height.
* **inline-block**: This is one display value that combines the properties of both block elements and inline elements. You get the ability to set a height and width, and the element can appear on the same horizontal line as other elements.
* **none**: Display none hides the element from the website and it will not be shown visually. This is very useful for CSS Dropdown menus where additional options appear when you hover on navigation menus. The rationale is that elements are set to a display value of none initially, and the display value is changed to block on hover.



Other display values can be found on the [W3 Schools website](http://www.w3schools.com/cssref/pr_class_display.asp)

## 2: Width and Height

Width and height properties are used closely with display:block and display:inline to set the width and height of HTML elements while creating a website. Common units units for Width and Height are:

* **px** - Pixels.
* **em** - A unit of measurement, where 1 em = current font size.
* **rem** - Root em. Same measurement as em, but makes life much easier without the inheritance problem
* **%** - Percentages.
* **auto** - highly useful unit, will explain below.

Other units of measurement can be found at the [W3 Schools website](http://www.w3schools.com/cssref/css_units.asp). If you’re wondering about the difference between px, em and rem, check out this great article on [font sizing with rem by Jonathan Snook](http://snook.ca/archives/html_and_css/font-size-with-rem)

Extremely useful properties like max-width, min-width, max-height and min-height come into play as well when you’re making responsive websites. Here’s one example of how auto and max-width can be used to make sure images fit tightly and snugly into available space:

img {

max-width: 100%;

height: auto;

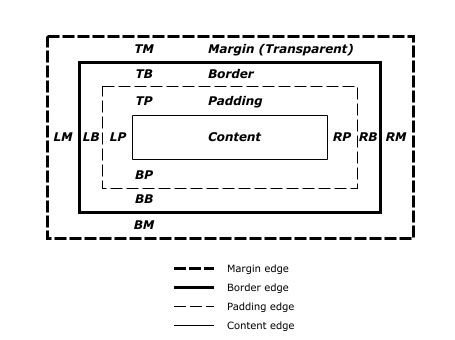
}

## 3: Margin and Padding

Margins are paddings are things that will definitely appear. Knowing how these things work will be extremely beneficial when writing CSS.

Margins and Paddings dictate the spaces between elements on your website. They are very similar and have the same units as Width and Height mentioned above.

The only difference between margins and paddings is the area the exert control over. Margins affect the area outside of borders whereas paddings affect areas inside the border. It is useful to refer to the box model below:



Ordinarily, margins are written in this manner:

div{

margin-top: 20px;

margin-bottom: 20px;

margin-right: 10px;

margin-left: 10px

}

They can be written in shorthand to simplify the lines of codes and make it easier to read. In fact, shorthands are the standard practice and you should know them. Here’s a quick explanation:

div{

margin: 20px 10px 20px 10px;

/\* This shorthand refers to TOP, Right, Bottom, Left. Its easier to picture a clock at 12, 3, 6 and 9 respectively \*/

margin: 20px 10px 20px;

/\* This refers to Top, Left and Right, Bottom \*/

margin: 20px 10px;

/\* This refers to Top and Bottom, Left and Right \*/

margin: 20px;

/\* This refers to 20px worth of margin on all 4 sides \*/

}

**Extra tip:** margins with auto on the left and right are used to center an element with a display value of block. Its written simply as:

div {

margin: 0 auto;

}

## 4: Border

Borders are… borders. I’m pretty sure you don’t need an explanation of what borders are.

Borders have 3 different properties that you have take care of:

* **border-width** – width of the border. Same units as width and height
* **border-style** – style of the border. Usual values are solid and dashed. For a complete list, take a look at [W3 Schools Website](http://www.w3schools.com/cssref/pr_border-style.asp)
* **border-color** – color of the border. Hex, and rgb values can be used.

Instead of writing the longer version, you could declare the border shorthand in this way:

div{

border: 1px solid black;

/\* border width, style and color \*/

}

Likewise to margins and paddings, borders refer to all 4 sides. If you are only interested in applying borders to 1 or 2 sides, I generally prefer to stick by border-top, border-bottom, border-left or border-right.

## 5: Floats

Floats are one of the core elements in today’s website. If you see two columns of text side by side, a sidebar / content configuration like what you see on my blog if your browser window is above 800px, you have noticed floats in action. Another commonly used area for floats are navigation items.

In the nutshell, what float simply does it to position the edge of the targeted HTML content at the edge of one side of the parent container. Subsequent floats will then be placed at the edge of your first floated content. (depending on whether you float it left or right)

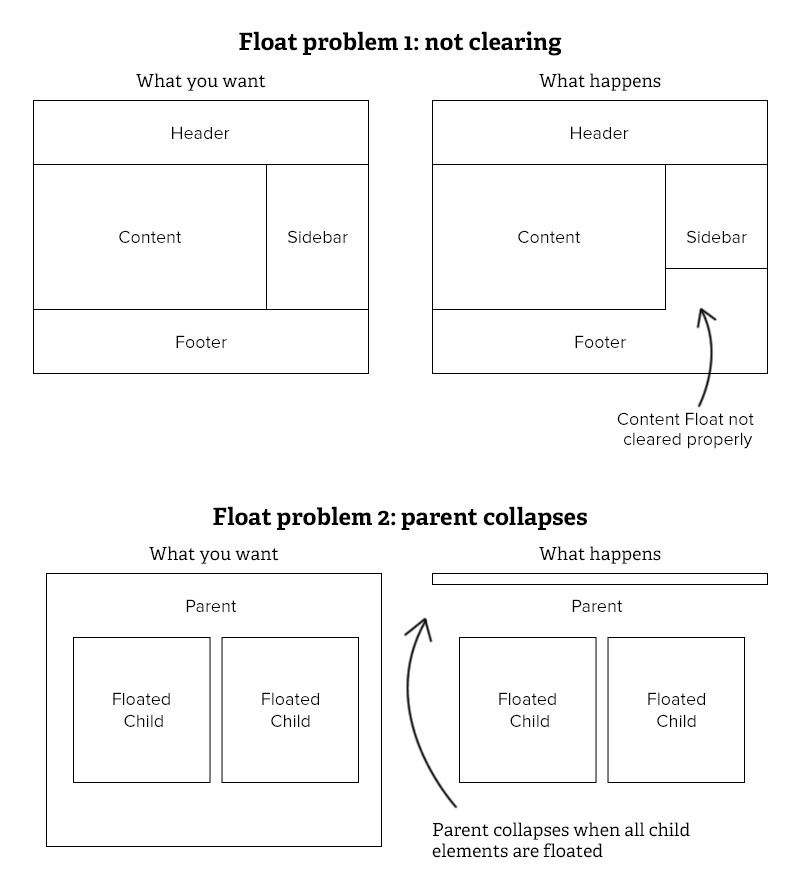
Floats have 3 basic properties that you might use often:

* **left**
* **right**
* **none** - removes the float

You may want to check out the this post on [float theory](http://coding.smashingmagazine.com/2007/05/01/css-float-theory-things-you-should-know/) as well

## 6: Clearing Floats

Even though floats are immensely useful, they are sometimes a headache if they are not cleared properly. In general, 2 kinds of problems may occur:



The are 3 main methods for clearing floats:

* **clear**: The Clear property takes in 3 values. left, right or both. As you might have suspected, clear:left; simply clears any floats on the left side, clear:right; clears floats on the right side while clear:both; ensures that all floats elements are cleared.
* **overflow:hidden**: This method is great for ensuring the parent element does not collapse like in problem 2. overflow:hidden; is set to the parent element to combat the problem.
* **clearfix**: No doubt you will have heard of or seen clearfix in tutorials. It is also set to the parent element to ensure float issues do not arise. The rationale behind the clearfix hack is to insert some content (a period) after the parent element to force the parent element to self clear since there is content after the floats.

This [post by Chris Coyier](http://css-tricks.com/snippets/css/clear-fix/) has a great snippets on the clearfix hack. Read more about the clearfix hack on the pages he link to, or just head right down to the bottom section to grab the latest version.

## 7: Color

**Color** here refers to text color. It takes on a #hex value or a rgb value as with border colors.

## 8: Background

**Background** refers to the background of the HTML element. Like many CSS properties, background has a shorthand to it as well.

body {

background:transparent image-url('image.png') left top no-repeat;

/\* All background definitions are option, but at least one must be stated. The above are default values given to background if you have left anything undefined \*/

}

Here are the explanations for the background properties in order from left to right:

* **background-color:** color of the background. Takes #hex value or an rgb value
* **background-image:** url(URI). Takes on the path to your image. Use the example above if the image is in the same folder.

To go down one folder, simple type the file name before the image.png. Example: css/image.png.

To go up one folder, type the file name with “…/”. Example: …/css/image.png

* **background-repeat:** whether you would like the background to repeat if the width exceeds the background size. Other values are repeat, repeat-x and repeat-y.
* **background-position:** position of the background relative to the HTML element. Two values are needed here, X and Y, where X is the amount of offset from the left and Y is the amount of offset from the top. Takes on either unit values (as with width and height) or left,center,right and top,center,bottom for left and right respectively.

## 9: Font

Fonts in general refer to the appearance of text in your website. There are a few things to look out for. Like other properties, font has a shorthand. Do note that it is common to see the font shorthand declaration only once in the whole CSS file. It is common to use the different properties at other times:

body {

font: italic small-caps bold 20px/1.5 "Proxima Nova", helvetica, arial, sans-serif;

/\* font shorthand \*/

}

Here are the explanations for the font properties in order from left to right:

* **font-style**: Style of the font. valid values are either italic or normal. Defaults to normal. Optional property in font shorthand
* **font-variant:** variant of the font. valid values are normal and small caps. Defaults to normal. Optional property in font shorthand and is not often used
* **font-weight:** weight of font. determines if text is bold. valid values are normal, bold, bolder, or 100 - 900. Optional property in font shorthand
* **font-size:** size of font. Takes a value thats the same as width and height
* **line-height:** determines the amount of space above and below the text. Very important to ensure good readability. Takes on the same values as font, and also a unitless value. If a unitless value is used, it means the line height is a multiple of the value provided.
* **font-family:** area to declare typefaces and fontstack that you would like to use.