Contents

[1 . INTRODUCTION 2](#_Toc505634876)

[2. PREREQUISITES 3](#_Toc505634877)

[3.GOALS 3](#_Toc505634878)

[4.WHAT 3](#_Toc505634879)

[5.HOW 4](#_Toc505634880)

[5.1HEAD 4](#_Toc505634881)

[5.2BODY 4](#_Toc505634882)

[6.WHY 5](#_Toc505634883)

[7.WHERE TO WRITE 5](#_Toc505634884)

[8.STRUCTURE 6](#_Toc505634885)

[9. Creating a website 7](#_Toc505634886)

[9.1 <h1> to <h6> 7](#_Toc505634887)

[9.2 <p> and <pre> tag 7](#_Toc505634888)

[9.3 <ul>, <ol> and <li> tags 8](#_Toc505634889)

[9.4 Media (Images, audio and video) 9](#_Toc505634890)

[10. Something's missing 11](#_Toc505634891)

[11. Getting started with CSS 11](#_Toc505634892)

[12. Cool CSS features 12](#_Toc505634893)

[12.1 Background 12](#_Toc505634894)

[12.2 Borders, Margins and Paddings 12](#_Toc505634895)

[12.3 Ids and Classes 13](#_Toc505634896)

[12.4 Miscellaneous 14](#_Toc505634897)

[12.4.1 Floats 14](#_Toc505634898)

[13 Identifying elements 15](#_Toc505634899)

[14 Keywords 16](#_Toc505634900)

[15 Good Practices 17](#_Toc505634901)

[960 Grid 18](#_Toc505634902)

[Choosing a container 19](#_Toc505634903)

[Grids 20](#_Toc505634904)

[Prefix and Suffix 20](#_Toc505634905)

# 1 . INTRODUCTION

# Hello Everyone

# Today, let's learn something new.

# You ever saw those amazing websites which make you go ahhhhhh ...wish i could do that. What if i said, you could. It's not that hard, not that easy. With focus and concentration we will do it. Before running, we learn to walk and before walking, we learn to crawl. So, let's start with crawling today.

<html>

<head>

<title>Hello</title>

</head>

<body>

Hi there beginners. This is not rocket science. Let’s set up your base so that you can make some beautiful websites.

</body>

</html>

# human-skeleton-3d-model-max-obj-3ds-fbx-mtl-mat.jpgNow, the gibberish which you think is written up here is not actually gibberish. It is the very foundation, the very core which you need in order to make your dream website. I like to compare it with this skeleton over here.­

# 2. PREREQUISITES

# -I want to say is there’s nothing in this world which is impossible to learn. All you need is motivation and an appetite to learn more.

# **-As such there are no requisites to learning HTML. Anyone who knows how to operate on a desktop can learn.**

# If someone somewhere thinks, it’s going too slow and you already know it, you are free to skip that topic.

# 3.GOALS

# -Knowledge in HTML,CSS and 960 Grid System.

# -Confidence to build your own website.

# The what, why and how is mandatory to learn anything. Let's take a look at HTML.

# 4.WHAT

# The first line we wrote thing you would have seen in the above snippet<html>. **HTML** means Hypertext Markup Language. A markup language is something which you use to process, define and present the text. You can arrange your text, define a structure to the page.

# 5.HOW

# This is how we will build "our skeleton." The <html> tag denotes the starting of your code. And also </html> denotes the end of your code. These keywords written in <> are called **tags**. So these are the ways to start and end the tags. You may also end a tag like <html/>. All tags in html are case insensitive, you may write them in upper or lower case. But follow just one of them for consistency.

## 5.1HEAD

Think of the head as your own. All important information goes there, same for html.

* The title of the page is defined in head tag.
* The information of the html(meta) is also stored in here.

## 5.2BODY

* Everything except the information in head is stored in here.
* Everything you see on a webpage is the body of it.

If you have any experience in any of the programming languages than you will know a compiler is required and without it we can’t move on. Now think about it, your website is going to be displayed on a browser, right? For markup languages you don’t need another compiler because you are not using it for any functionality.

# 6.WHY

# To make websites of course. As I said earlier, with this you create the structure, the “skeleton”. For it to look really good, we will learn another thing in this document itself.

# 7.WHERE TO WRITE

# While it is true that HTML can be written in any **text editor**(including notepad). But we should chose our text editor wisely, because there’s so much a text editor can do like - predicting text(like mobile phones predict text), indent code by itself.

# Moreover, most text editors these days will highlight parts of your code. When it is written properly, the highlighting will make it much easier to read. When you have errors, it won’t highlight properly. This makes the errors much easier to spot.

# My recommendation to the beginners is to go either Sublime Text or Brackets, for experienced users you may use any text editor of your choice. Now, I will be listing down the installation instructions on how to install either of these. Those who already have one may go to the next section.

# Based on your requirements you may choose any one of them. If you want something light which doesn’t take much space, go for Sublime.

# 1.Go to www.sublimetext.com/3 for Sublime Text Editor or brackets.io for Brackets text editor.

# 2. Download the software for your respective build.

# 3. You are now ready to code.

# 8.STRUCTURE

# We already discussed about the structure a bit informally. Let’s get see how it really is.

# 

# Let’s see what is what over here.

# The title you see there Google, it was written in the head tag.

* The logo, search bar, profile option and the footer is always written in the body.

But there's a lot you have to do to achieve things like these. Let's start by creating our very own website in the next section.

# 9. Creating a website

In this website, we will write an article on our favorite sport.

Let's get started.

* First, we have to write <html>. It is a good practice to close the tags before hand and then writing content inside. (Sublime text and brackets both do that passively)
* Now, we write the <head> tag and in it define the title.
* In the <body> tag you must be wondering what to do next. Let's look at some of the tags you will be using to complete your first website.

## 9.1 <h1> to <h6>

Any <hx> tag where x=1 to 6 is called a header tag. <h1> being the biggest one and <h6> is the smallest. This is the best way to give header to any article. So, let's do this.

In the body tag, write :

This will create a heading. Now, you would want to write something about it. Let's see.

<body>

<h1>My Sport</h1>

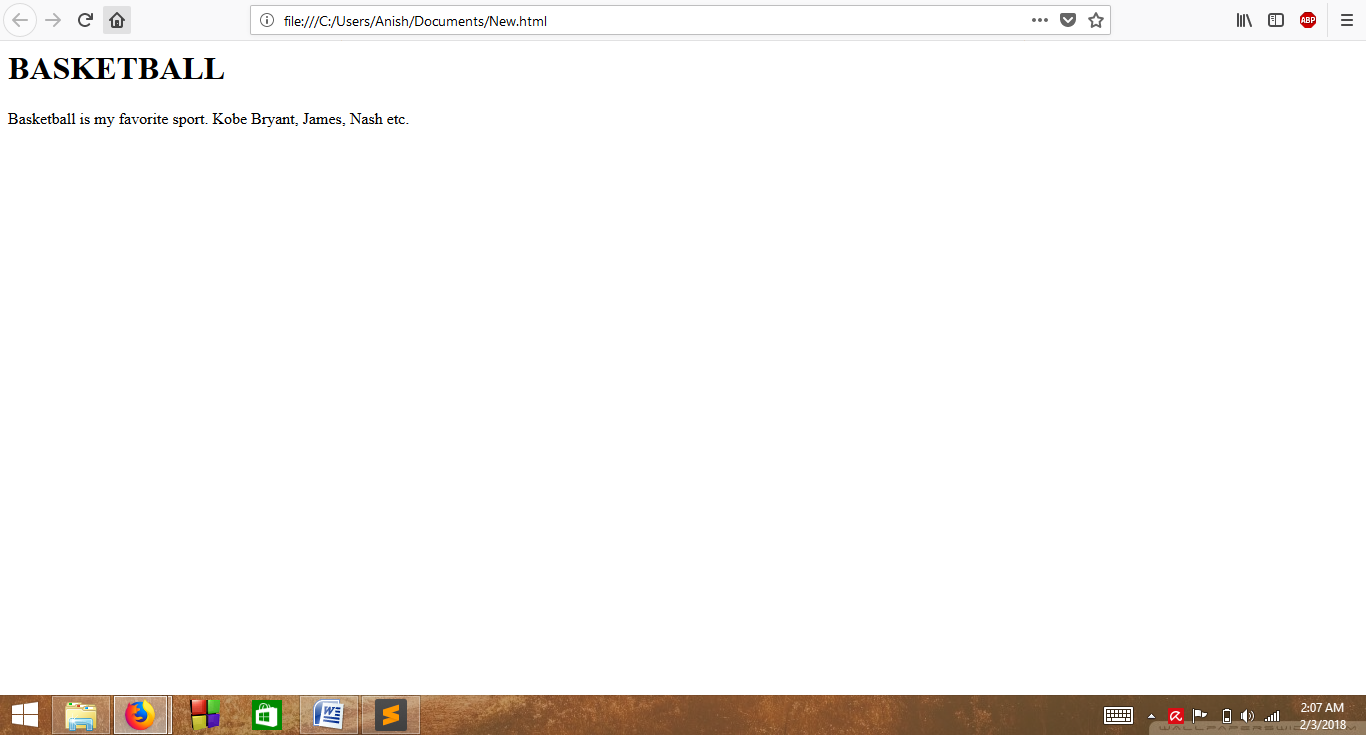
</body>

## 

## 9.2 <p> and <pre> tag

The < p> tag stands for paragraph, you can write anything about your sport in this tag. There's also another tag called as <pre> which stands for pre-formatted text. If you want the text to look exactly the same color, font or width wise, go for <pre> tag . It should be used in the rarest of instances and not anywhere you like. For eg in the footer of your website.

So, that's it if you wrote the tags properly and save the extension as .html , you shall have a site up and running and it should look something like this.



Also, you can bold any text or italise it using <b> or <i> tag.

So, good job people, you have successfully made your own website.

Now, what if I say list down the things you like about basketball. A proper bullet or numbered list. Let's see how to do it.

## 9.3 <ul>, <ol> and <li> tags

Here, ul stands for unordered list, ol stands for ordered list and li stands for list items.

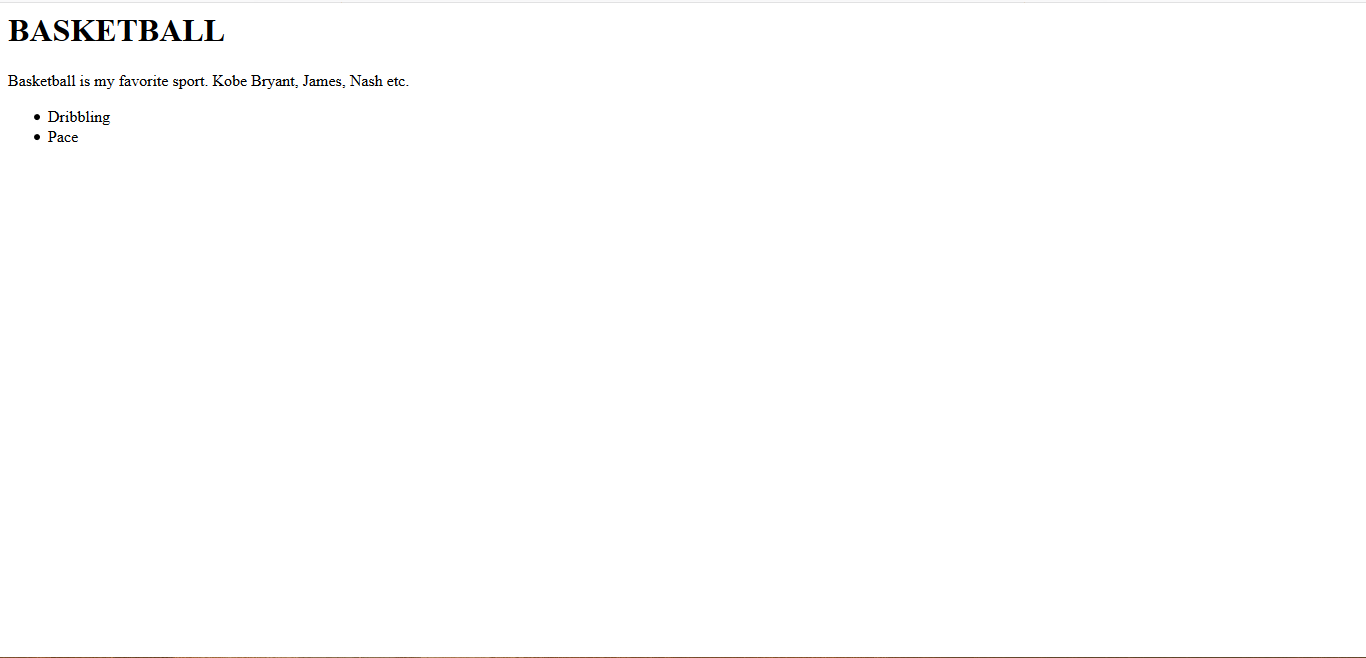
So, if you have to make a list of things you like about basketball on a webpage, it will go like this.

<ul>

<li>Pace</li>

<li>Well defined rules</li>

</ul>

This will create an unordered list which is bullets yes. If you want it to be numbered just replace the ul with ol.

After completing it shall look like this.

If you want to make an ordered list, just replace the ul with ol.

Good job guys! Its somewhat looking like a website.

But, remember any website you have seen without audio, images or video? No, right. Media plays a really important role to any website. Let's see how to add your own.

## 9.4 Media (Images, audio and video)

To insert an image, we use <img> tag but just an img tag would paste the photo and you wont be able to change anything. That's why we use **attributes**.

A normal image tag would look like this :

<img src = "images/image.png" alt = "Basketball" width = "100px">

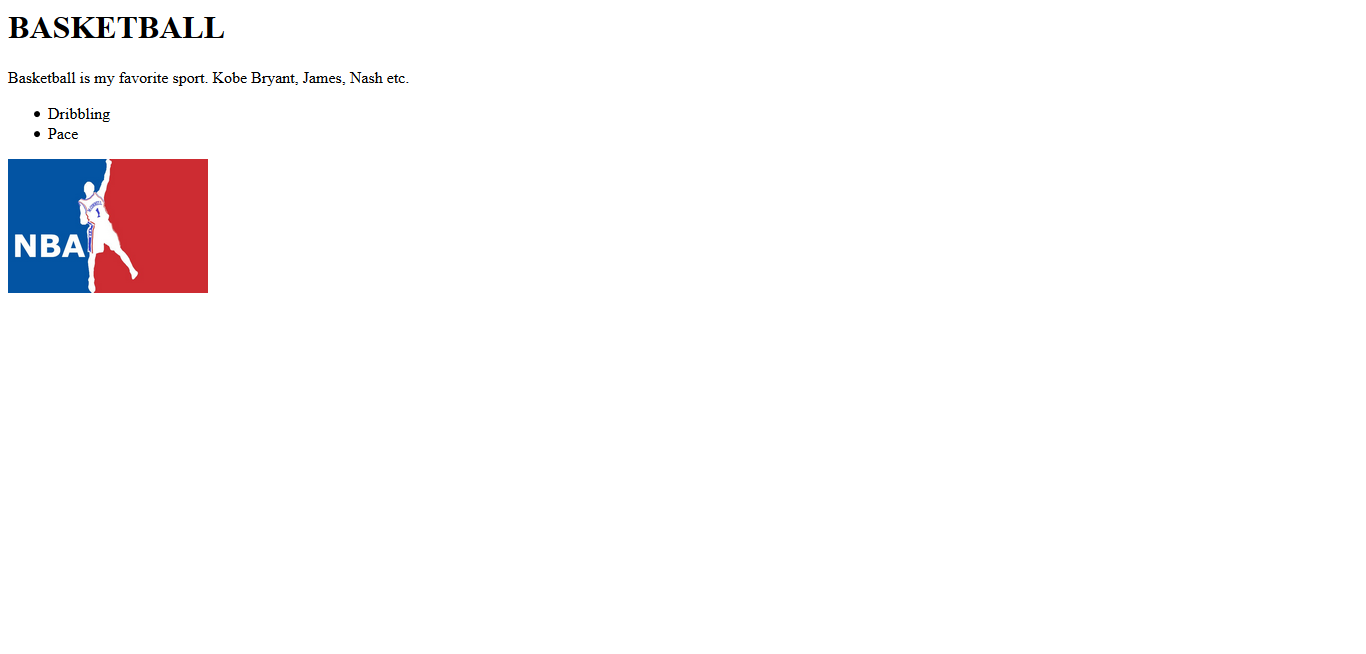
Now, before i confuse all of you, let's take it step by step

src : The source of the image file, where the file is being placed.

alt : Sometimes, the image fails to load or the browser is unable to render it .So, this attribute stands for "alternate text", if anything goes wrong. Pretty handy right?

width : Self explanatory

So, let's insert any image of your choice, first download it on your pc, then copy the source from the properties.



Good job. Now, let's get a bit technical here. You must have given the path like C:/Users/.. .... so and so right. Let's say when your site makes it to the server, then it won't find the this directory because it is present only in your pc. **We should never give absolute paths.** But always give paths relative to our html file. For eg. if it was stored in the same folder - the file and the image. I would have just given : src = "logo.jpg" , if it was in another folder in the same directory , I would have given src = "folder/file.jpg". Like this. We will learn more about it in the good practices section. But for now, just keep it in mind.

# 10. Something's missing

It's not looking like a website. Right? It shouldn't, you are just defining the structure. Remember, the skeleton? Is a skeleton really presentable? No. You need tissues, muscles, skin and finally some clothes to go on. Now, from where would you get it?

Cascading Style Sheets(CSS) is the answer.

CSS is the bread and butter of web developers. But without a structure there can't be any CSS, and nobody wants to look at a bad website. CSS and Html go hand in hand both are necessary. Let's see some of the magic that CSS can do.

# 11. Getting started with CSS

Let's see how to change the color of your header in your sports website.

It will be like this :

<h1 style = "color: red">Basketball</h1>

Now, this is the first step of customizing your content. Notice how, we placed the style attribute in there, this is called inline CSS. But, imagine when you will make a full fledged website.

So, it is not a **good practice** to write inline CSS. Instead what we do is make a different file with the extension of .css and link it with the html.

Let's say you want to make the header red.

Make a file basketball .css and in that write

h1 {

color : red;

}

And in your html file in the head section , you write

<link rel = "stylesheet" href = "css/basketball.css">

Step by step, let's see

* The link tag specifies that there's some linking that needs to be done.Quite simple.
* rel attribute stands for relationship
* And href means reference (to which file you are referring to)

Notice how I used the relative path again(You should make it a practice too).

Now, that your sheet is linked, you are free to do any changes.

# 12. Cool CSS features

## 12.1 Background

Bored with that white background?

Let's see how to change that... In your CSS file, make a section like this

body {

background : grey;

}

Note that you dont have to type the color, you can give the rgb value or hex code too which you will find easily on google.

## 12.2 Borders, Margins and Paddings

Let's say you want to apply border on a division or an image or anywhere.

div {

border : 1px solid white;

}

Here, 1px is the width , solid means there will be a constant line(you can go for dashed too) and last is the color.

Margins and Paddings are really necessary to position your elements. The following figure will tell you the difference



In any div if you want to set the padding and margins :

div {

padding-top : 10px;

}

You can do it like this. Similarly for left, right, bottom and margins too.

Notice one thing, that div means for all divisions. You will never practically use it because each element needs to be differently customized. So we can assign an id or a class accordingly. Let's see how to do that.

## 12.3 Ids and Classes

To assign an id or a class to any element, it is like adding an attribute. Have a look at the syntax

<p id = "first\_paragraph">Hello, people </p>

That's how you add an id to an element. Now, you can specifically customize this.

In your CSS file, you use it like this

#first\_paragraph {

color: grey;

}

Similarly, if there's a group of elements which needs to have the same customization. Use a class and a class is accessed like this :

.header{

text-font : "Times New Roman", serif;

}

## 12.4 Miscellaneous

CSS is very vast. So vast that you can't learn everything about it in one go. As you keep moving forward, keep striving for knowledge, you will get there. I will give you all a head start and rest looking for stuff is your job. After this I will provide you with a few links to refer to, but note that they are not the only pieces of information.

So, before wrapping up CSS, I will give a list of really handy things.

### 12.4.1 Floats

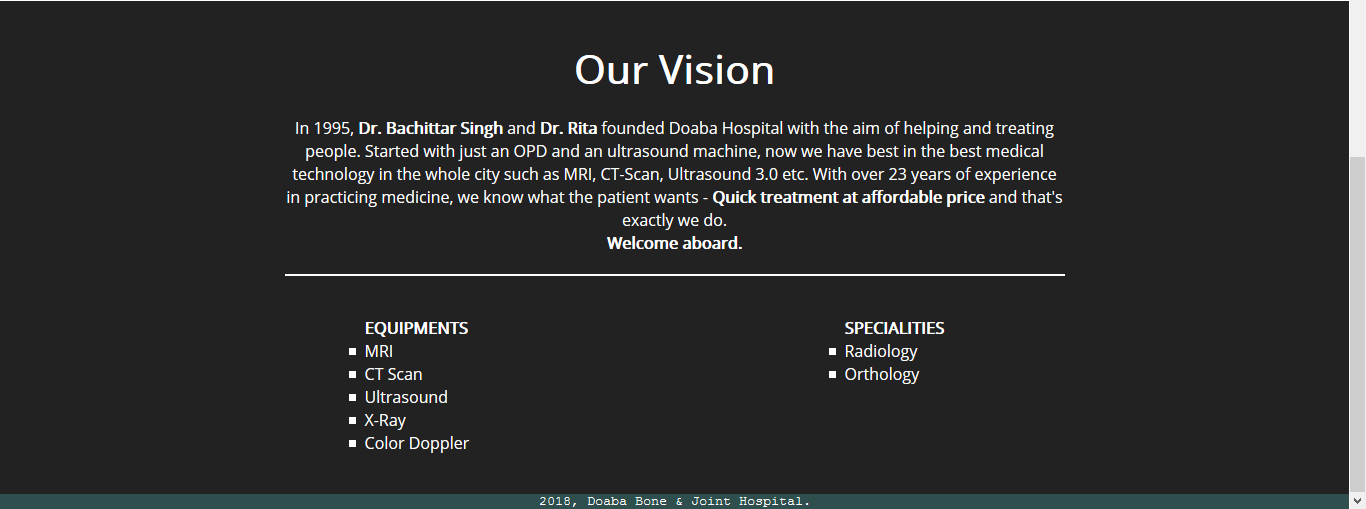
While using div, you can make it float to either left or right depending on layout you want for your website.

# 13 Identifying elements

Just like a musician listens to a song and identifies the chords, By looking at a web page, we should have some idea on what's going on there. It's not mandatory but helps.

There's another thing call inspect element in web browsers with which you can check the code of any website which is pretty handy. With that, you can have an idea what you want to make and how to.

Have a look at this :-

Here, take a few minutes and see what's going on.

See you on the next page.

* There's a division with color dark grey
* There's an h tag used which is placed in center using CSS
* The paragraph tags with a few words in bold
* Then a horizontal line (<hr> tag)
* Two list items (next topic will show you how they are placed)
* Footer tag with a different font

If you were able to make that out, good job!

Now, this much information is good for you to get started and explore. Self learning is one of the best experiences and helps you retain too. Now, I will refer you a few websites go check out.

1. Nothing teaches better than the official website documentation. So, be it html or css, check out their official website documentation.
2. https://developer.mozilla.org/en-US/docs/Web/HTML for HTML
3. https://developer.mozilla.org/en-US/docs/Web/CSS for CSS

# 14 Keywords

1. HTML
2. tag
3. attribute
4. div
5. text editor
6. CSS
7. Cascade
8. Mark up
9. Box object Model

You must have seen how i indent my code and all the spaces. There are some ethics or I should say some good practices which need to be followed when you code because you are going to be working in a team and a standard must be maintained. Just like English is the standard language because all can read and understand on a team, similarly it goes for good practices.

# 15 Good Practices

* Always indent your code properly. This enhances the readabiliity and makes it easier to keep track of your tags.
* Add comments for the same purpose. When you work on a project you know what you are doing no doubt, but your team should also know. Comments make it easier to understand
* Give relative paths. When the site makes it to the server it would start searching for C:/Users etc and it wont display. So always give relative paths.
* Make different folders for css and images files.
* Use meaningful ids and classes
* Write a meaning alt attribute for img tag

You can't always remember all the tags and their functionality. So, better keep a reference material (cheat sheet)with you at all times which will help you to perform better and swiftly.

Now, i want you to go practice on a few websites. And come back here.

So, now we meet after some practicing. I hope you are all doing good. I believe you found the resources useful and learned alot. Now, the problem which I think got the best of you was positioning the elements. It's really hard to position the elements at the level which you are solely using HTML, CSS of your own.

You remember till 5th grade, your maths notebook had rows and column - a grid for you to write in boxes. What if I could give you this ability too? What If I say some good folks made a grid so that we can use it and position our elements easily without too much worrying?

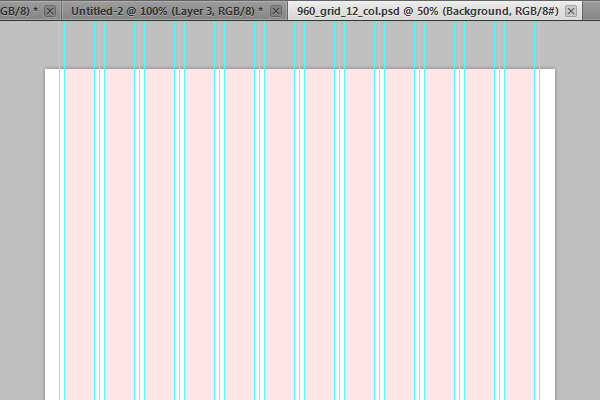
Yes, it has already been created. We from now on will use a grid system that is 960 Grid. Here, the 960 stands for 960 width. Which you means you shall be able to create a site with a width of 960px easily.

==============================================================================

# 960 Grid

960 Grid is a CSS framework (Pre defined classes ready to be used) which makes it easy to position your elements because all the structure of the page is pre defined. You just need to have the vision and you can place it instantly and easily.

Here's how the structure looks like.

Now, this is one of 12 column layouts that 960 grid framework provides.

The package comes with a lot of css files, you can choose any of your liking for eg

12 column grid, 16 column and 24 column.

We also recommend to other files to be linked :

1. reset.css : This css files reset all the margins, padding and positions with which it gets same for every browser be it firefox, safari, chrome . So that you don't have to worry about every browser.
2. text.css : It is the built in typography which has decent fonts already applied. You can override those with your own css sheet.

Never, change the css files of the framework. They are tried and tested. If you want to apply your own styles, override that particular element in your own stylesheet.

Let's have a look at what 960 Grid is capable of :

## Choosing a container

Now, it is up to you if you want 16 columns, 24 or 12.

So, to apply a 12 column grid, this is how we do it

<div class class = "container\_12">

This is how you apply 12 grid system.

</div>

Now, you have successfully made a 12 column grid.

Next, we will learn how to give appropriate size to our elements.

## Grids

That's the beauty of frameworks, it makes everything so easy. Let's say you are making a huge header and you want it to take the whole width.

All you have to do is :

<div class = "container">

<div class = "grid\_12">

<!-- insert header here-->

</div>

</div>

If you want 2 paragraphs taking half of page. Then use grid\_6 for both of them in the same container. This way, they will take up exactly half of the page. Now, if you for instance give grid\_6 to one and grid\_ 7 to the other, then it will automatically move to the next line. This is a very nice way to create even pages.

## Prefix and Suffix

Now you must be wondering what if you want to leave some space in between or start from the 3rd column or so. That is also possible in 960 Grid system.

If you use class = "prefix\_3" this will leave 3 columns in front and start from the 4th one.

And suffix\_3 will leave 3 columns behind.

As you know, an element can have any number of ids and classes. So, similarly you can club, grid\_x and prefix\_y and suffix\_z where x,y,z = 1 to 12 to get the desired result.

Let's show you an example of how 960 Grid really is :



Here, you can 960 grid has been used. Please note

* Its all in one container
* The header takes up the whole 12 columns
* The paragraph also takes up the whole 12 columns.
* The equipments are specialies are given exact size but prefix and suffix has been used.
* The footer takes up the whole 12 columns.

Even though the tutorial for 960 Grid was a small one but it has way too many facilities. Please note it is not the only grid system which you have to use but it's good for starting out. So, let's try to make our website a little better positioned and see you next time.

We will make things a little easier soon. ;)

Happy websiting