\*\*\*\*\*\*HTML5\*\*\*\*\*\*

BASIC EXAMPLE

<!DOCTYPE html>

<html> //HTML TAG

<head> //HEAD TAG

<title>gvk garu </title> //TITLE TAG

hahaha

</head>

<body> //BODY TAG

iga start chedham

</body>

</html>

TO USE A PARAGRAPH

<p>Hello there!</p>

HEADING ELEMENTS

The following is the list of heading elements available in HTML. They are ordered from largest to smallest in size.

<h1> - used for main headings, all other smaller headings are used for subheadings.

<h2>

<h3>

<h4>

<h5>

<h6>

UNORDERED LISTS

<ul> tag </ul>

INDIVIDUAL LIST ITEMS

<li>list tag</li>

ORDERED LISTS

<ol> tag </ol>

LINK TO OTHER WEB PAGES

EXAMPLE(without href attribute):<a>This Is A Link To Wikipedia</a>

EXAMPLE(with href attribute): <a href="https://www.wikipedia.org/">This Is A Link To Wikipedia</a>

For a link to open in a new window, the target attribute requires a value of \_blank. The target attribute can be added directly to the opening tag of the anchor element, just like the href attribute.

EXAMPLE:<a href="https://en.wikipedia.org/wiki/Brown\_bear" target="\_blank">The Brown Bear</a>

ADDING IMAGE

The <img> element lets you add images to a web page. This element is special because it does not have a closing tag, it only has an opening tag. This is because the <img> element is a self-closing element.

EXAMPLE

<img src="https://www.example.com/picture.jpg" />

The alt attributes also serves the following purposes:

If an image fails to load on a web page, a user can mouse over the area originally intended for the image and read a brief description of the image. This is made possible by the description you provide in the alt attribute.

Visually impaired users often browse the web with the aid of of screen reading software. When you include the alt attribute, the screen reading software can read the image's description outloud to the visually impaired user.

EXAMPLE:<img src="#" alt="A field of yellow sunflowers" />

IMAGES INTO LINKS

it's possible to turn images into links by simply wrapping the <img> element with an <a> element.

EXAMPLE:<a href="https://en.wikipedia.org/wiki/Opuntia" target="\_blank"><img src="#" alt="A red prickly pear fruit"/></a>

LINE BREAKS:

you can use HTML's line break element: <br />.

EXAMPLE:Shall I compare thee to a summer's day?<br />Thou art more lovely and more temperate

ADDING COMMENTS TO HTML FILE

Comments begin with <!-- and end with -->. Any characters in between will be treated as a comment.

EXAMPLE:<!-- This is a comment that the browser will not display. -->

TABLE TAGS:

<table> </table>

To add rows using the table row element:

<tr> </tr> //each tag for each row and in between table tags

 Add data using the table data element:

<td> value </td> //in between row tags

 A border could be added to a table using the border attribute and setting it equal to an integer. This integer would represent the thickness of the border.

Example: <table border="1">

<tr>

<td>73</td>

<td>81</td>

</tr>

</table>

Data can span columns using the colspan attribute. The attributes accepts an integer (greater than or equal to 1) to denote the number of columns it spans across.

Example: <td colspan="2">Out of Town</td>

The rowspan attribute is used for data that spans multiple rows (perhaps an event goes on for multiple hours on a certain day). It accepts an integer (greater than or equal to 1) to denote the number of rows it spans across.

Example: <td rowspan="2">Work</td>

For headings of the tables we use <thead> </thead> tags

Long tables can be sectioned off using the table body element: <tbody> </tbody>. //same as body in html file

The bottom part of a long table can also be sectioned off using the <tfoot> </tfoot> element.

Example:

<tfoot>

<td>Total</td>

<td>28</td>

</tfoot>

You can use CSS to style tables just like you have done in the past. Specifically, you can change style the various aspects

Example:

th, td {

font-family: 'Lato', sans-serif;

font-weight: 400;

padding: 20px;

text-align: left;

width: 33.3333%;

}

\*\*\*\*\*\*\*CSS\*\*\*\*\*\*

LINKING CSS FILE TO CSS

<link href="style.css" type="text/css" rel="stylesheet">

**using <style> </style>**

tags in between <head></head>tags we can directly write a css code in html file.

Example for heading :

<head> <style> h2 { font-family: Arial; } </style> </head>

Example for img ,a and \*(all modules)

\* { font-family: 'Georgia', 'Times', serif; }

a { color: SeaGreen; text-decoration: none; }

img { border-radius: 100%; }

COLORING TEXT:

 color - this property styles an element's foreground color.

 background-color - this property styles an element's background color

157 COLORS: <http://www.colors.commutercreative.com/grid/>

RGB (Red, Green, Blue) colors offer the option of 16,777,216 possible colors.

USING RGB COLORS COLORING A TEXT:

ECAMPLE: h1 {

color: rgb(123, 20, 233);

background-color: rgb(99, 21, 127);

}

Hex color codes also offer 16,777,216 color options, but they follow a different syntax.

EXAMPLE:

h1 {

color: #09AA34;

}

COLORING USING HSL:

HSL stands for **H**ue, **S**aturation, and **L**ightness

Hue - the technical term that describes what we understand as "color." In HSL, hue is represented on a color wheel. It can take on values between 0 and 360.

Saturation - the amount of gray in a given color. In HSL, saturation is specified using a percentage between 0% and 100%. The percentage 0% represents a shade of gray, whereas 100% represents full saturation.

Lightness - the amount of white in a given color. Similar to saturation, lightness is specified using a percentage between 0% and 100%. The percentage 0% represents black, whereas 100% represents white. 50% is normal.

EXAMPLE:

h1 {

color: hsl(182, 20%, 50%);

}

 To modify opacity in RGB colors, CSS offers the rgba() value. Note the slight difference in rgb() and rgba().a is alpha value. It represents the opacity of a color. The alpha value can be a number between 0 or 1,

h1 {

color: rgba(123, 88, 9, 0.5);

}

 The alpha value can also be used for HSL colors, using hsla()

h1 {

color: hsla(239, 45%, 22%, 0.4);

}

FONT

To change the typeface of text on your web page, you can use the font-family property

EXAMPLE: h1 {

font-family: Garamond;

}

When the name of a typeface consists of more than one word, it must be enclosed in double quotes

EXAMPLE: h1 {

font-family: "Courier New";

}

two types of FONTS: serif fonts and sans-serif fonts.

Serif - the letters in these fonts have extra details on the ends of each letter. Examples include fonts like Times New Roman or Georgia,

s-Serif - the letters in these fonts do not have extra details on the ends of each letter. Instead, letters have straight, flat edges. Some examples include Arial or Helvetica.

When the stylesheet specifies a font not installed on a user's computer, the pre-installed fonts can be used as fallback fonts for users.

EXAMPLE: h1 {

font-family: Garamond, Times, serif;

}

FONT SITE: <https://fonts.google.com/>

To use a Google Font, you can use a <link> element, just like you did for a CSS stylesheet:

EXAMPLE: <head>

<link href="https://fonts.googleapis.com/css?family=Raleway" type="text/css" rel="stylesheet" >

</head>

To change the size of text on your web page, you can use the font-size property.

p {

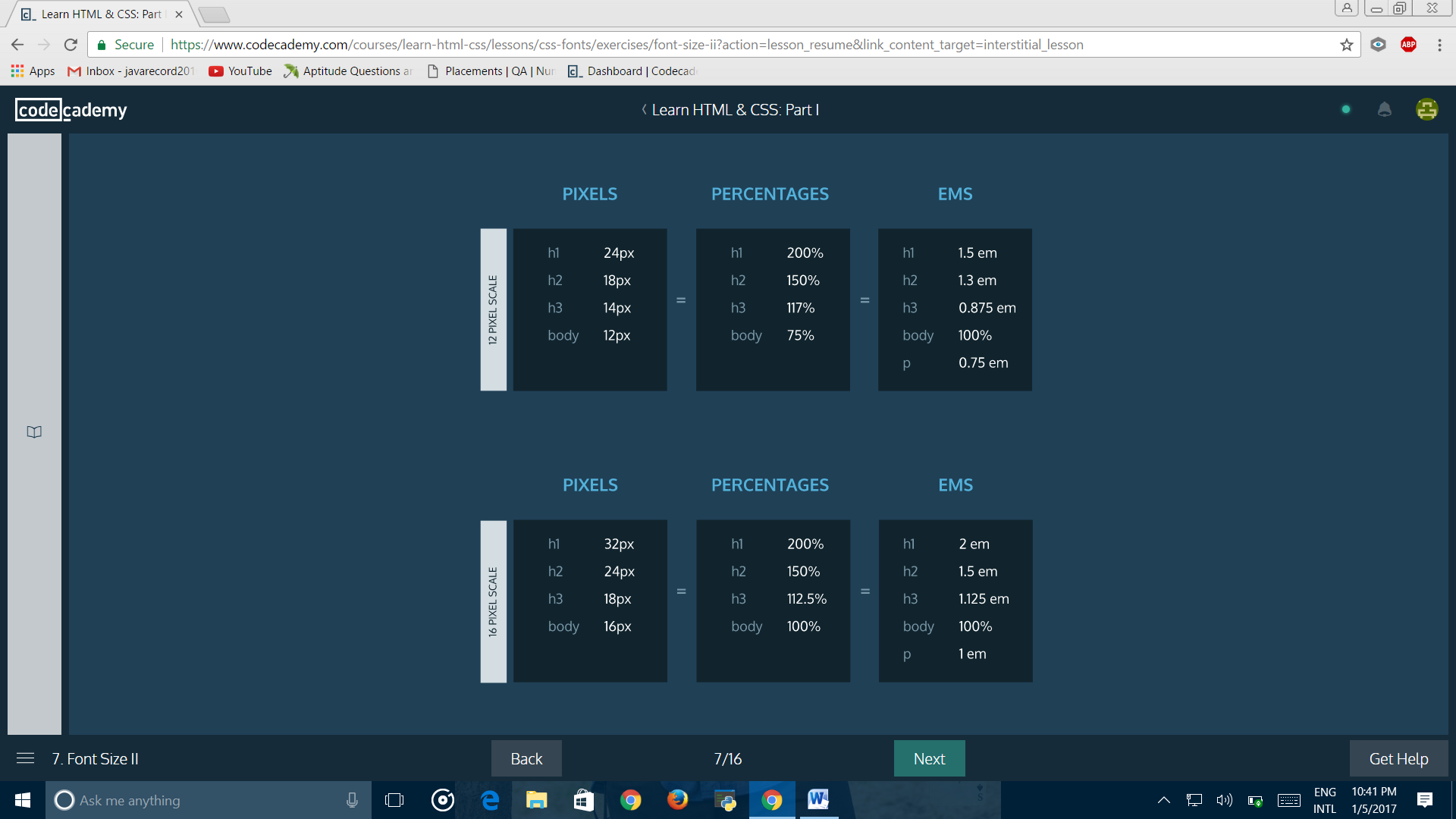
font-size: 18px;

}

There are three units of measurement for font size: PIXELS(px)

PERCENTAGE(%)

EMS(em)



EXAMPLES:

PIXEL:

p {

font-size: 18px;

}

PERCENTAGE:

p {

font-size: 150%;

}

Ems:

p {

font-size: 1.3em;

}

LINE HEIGHT

 you can modify the spacing between lines of text with the line-height property.

p {

line-height: 1.5em;

}

When the line height is increased, the spacing between lines of text increases, which can make text easier to read.

The line height can be modified using pixels or ems, but the unit of ems is preferred, since ems offer a spacing relative to the size of the text on the page.

You can also increase the spacing between words in a body of text using word-spacing.

The default amount of space between words is usually 0.25em.

h1 {

word-spacing: 0.3em;

}

T

he technical term for the spacing between letters is the "kerning". The kerning can be adjusted with the letter-spacing property in CSS.

h1 {

letter-spacing: 0.3em;

}

the font-weight property turns bold on or off.

p {

font-weight: bold;

}

 the font weight of that particular element to normal, essentially "shutting off" bold for that element.

You can also *italicize* words with the font-style property.

h3 { font-style: italic; }

Text can also be styled to appear in either all uppercase or lowercase with the text-transform property.

h1 { text-transform: uppercase; }

To move, or align, text, we can use the text-alignproperty.

h1 { text-align: right; }

The text-align property can be set to one of the following three values:

1. left - aligns text to the left hand side of the browser.
2. center - centers text.
3. right - aligns text to the right hand side of the browser.

With the proper labels, we can style individual HTML elements! Specifically, we can label HTML elements with a unique identifier, or *ID*. We can then style that specific element in the stylesheet.

To label an element with an ID, we can use the id attribute on an HTML element.

<h1 id="botswana">Botswana</h1>

To style a specific element labeled with an ID, you can use an *ID selector* in the stylesheet.

#botswana {

background-color: #56ABFF;

}

To style elements of the same class, you can use a class selector in the stylesheet.

Class selectors begin with a period (.) and are immediately followed by the name of the class

.science {

font-family: Georgia, Times, serif;

color: #A3B4C5;

text-transform: uppercase;

}

.breaking {

font-family: Georgia, Times, serif;

}

p.breaking {

line-height: 1.3em;

} The example above uses a new selector: p.breaking. What's the difference between the .breaking and p.breaking selectors?

The .breaking selector targets all elements with a class of breaking. The p.breaking selector targets only <p>elements with a class of breaking. This type of selector allows you to be even more specific about a particular element's styling, even when that element must share some styling with other elements.

you learned how to use a multiple element selector to style multiple elements at once.

h1, p {

font-family: Garamond, serif;

}

The same syntax can be used to style multiple classes at once.

.first, .last {

font-size: 20px;

}

It's also possible to label HTML elements with more than one class.

<h1 class="book domestic">The Way of the Deep</h1>

<h1 class="book foreign">A Night in the Sky</h1>

.book {

font-family: Georgia, serif;

}

.domestic {

font-color: #0902CC;

}

.foreign {

font-color: #B097DD;

}

Classes and IDs are useful for labeling elements, but a disorganized, overuse of classes and IDs can result in an HTML file that is difficult to read. We need a way of organizing the contents of the HTML file. This will help:

1. Keep HTML code easy to read.
2. Group elements that belong together.

HTML offers an element that is the backbone of code organization: the div, represented by <div> in HTML.

You can think of the div as a box, or container, that groups elements that belong together.

For example, a <div> can be used to group together all of the elements that make up the navigation for a web page, or any other section of a page.

<div>

<h1>Alice In Wonderland</h1>

<p> ... </p>

</div>

<div> with class

<div class="container">

<h1 class="title">Alice In Wonderland</h1>

<p> ... </p>

</div>

div.container {

background-color: rgb(252, 255, 205);

font-family: Roboto, Helvetica, sans-serif;

}

h1.title {

color: #0D1A2F;

}

**Box model**:

1. The dimensions of an element's box
2. The borders of an element's box
3. The content within an element's box
4. The area around all four sides of an element's box

The default position of an element can be changed by setting its position property. The position pr-widoperty can take one of four values:

1. static - the default value (it does not need to be specified)
2. relative-This value allows you to position an element relative to its default position the web page.
3. Absolute-all other elements on the page will ignore the element and act like it is not present on the page.
4. Fixed-  
   We can fix an element to a specific position on the page (regardless of user scrolling.

 the div has tobe positioned using two of the four *offset properties*. The valid offset properties are:

1. top - moves the element down.
2. bottom - moves the element up.
3. left - moves the element right.
4. right - moves the element left.

EXAMPLE:

.question {

margin: 120px auto;

text-align: center;

position: relative;

top: 240px;

}

When boxes on a web page have a combination of different positions, the boxes (and therefore, their content) can overlap with each other, making the content difficult to read or consume.

The z-index property controls how far "back" or how far "forward" an element should appear on the web page.

The z-index property accepts integer values. Depending on their values, the integers instruct the browser on the order in which elements should be displayed on the web page.

EXAMPLE.

div.navigation {

background-color: #4BF5FF;

width: 100%;

height: 100px;

position: fixed;

z-index: 1;

}

div.description {

background-color: #1D09AF;

width: 50px;

height: 50px;

position: relative;

top: 200px;

}

If you're simply interested in moving an element as far left or as far right as possible on the page, you can use the float property.

The float property can be set to one of two values:

1. left - this value will move, or float, elements as far left as possible.
2. right - this value will move elements as far right as possible.

div.block {

width: 100px;

float: left;

}

The clear property specifies how elements should behave when they bump into each other on the page. It can take on one of the following values:

1. left - the left side of the element will not touch any other element within the same containing element.
2. right - the right side of the element will nottouch any other element within the same containing element.
3. both - neither side of the element will touch any other element within the same containing element.
4. none - the element can touch either side.

div {

width: 200px;

float: left;

}

div.special {

clear: left;

}

 CSS offers two properties that can limit how narrow or how wide an element's box can be sized to.

1. min-width - this property ensures a minimum width for an element's box.
2. max-width - this property ensures a maximum width for an element's box.

p {

min-width: 300px;

max-width: 600px;

}

You can also limit the minimum and maximum *height* of an element.

1. min-height - this property ensures a minimum height for an element's box.
2. max-height - this property ensures a maximum height for an element's box.

p {

min-height: 150px;

max-height: 300px;

}

The overflow property controls what happens to content when it spills, or *overflows*, outside of its box. It can be set to one of the following values:

1. hidden - when set to this value, any content that overflows be hidden from view.
2. scroll - when set to this value, a scrollbar will be added to the element's box so that the rest of the content can be viewed by scrolling.

p {

min-width: 300px;

max-width: 600px;

min-height: 150px;

max-height: 300px;

overflow: scroll;

}

**BOX BORDERS**

It's not possible to view a box's border if the border's *style* has not been set. A border's style can be set with the border-style property. This property can take on one of the following values:

1. solid - border is a solid line.
2. dashed - border is a series of lines or dashes.
3. dotted - border is a series of square dots.
4. double - border is two solid black lines.
5. groove - border is a groove (or carving).
6. inset - border appears to cut into the screen.
7. outset - border appears to pop out of the screen.
8. ridge - border appears as a picture frame.
9. hidden or none - no border.

div {

border-style: solid;

}

You can control the thickness, or width, of borders with the border-width property. The value of border-width is given in pixels.

p {

border-style: solid;

border-width: 5px;

}

It's also possible to also set the border-width property to one of the following named thicknesses:

1. thin
2. medium
3. thick

Another version of the border-width property allows you to specify the width for each side of the border.

p {

border-style: solid;

border-width: 3px 1px 2px 1px;

}

The values in the example above refer to the border width in clockwise order (top: 3 pixels, right: 1 pixel, bottom: 2 pixels, left: 1 pixel).

If you'd like to be even more specific about the width of different sides of the border, you can use the following properties:

1. border-top-width
2. border-right-width
3. border-bottom-width
4. border-left-width

Each property affects the width of only one side of the border, giving you more flexibility in customization.

p {

border-style: solid;

border-left-width: 4px;

}

The color of a border can also be customized with the border-color property.

div.container {

border-style: solid;

border-width: 3px;

border-color: rgb(22, 77, 100);

}

div.container {

border-style: solid;

border-width: 3px;

border-color: rgb(22, 77, 100);

}

The code in the example above can be shortened using the border property:

div.container {

border: 3px solid rgb(22, 77, 100);

}

The corners of an element's border box can be modified with the border-radius property.

div.container {

border: 3px solid rgb(22, 77, 100);

border-radius: 5px;<!—100%--> }

**IMAGES**

we can modify the dimensions of an image using the width and height properties.

<img src="#" alt="" class="" />

img.X {

width: 350px;

height: 200px;

}

 By default, images are inline elements. For images to center properly, they must behave as block-level elements.

img.leaf {

width: 300px;

height: 200px;

display: block;

}

In the example below, the image is aligned using the margin property. The top and bottom margins of the image's box are set to 0 margin. The left and right margins are set to auto, which automatically sets the exact amount of margin needed on the left and right sides in order to center the image.

img.leaf {

width: 300px;

height: 200px;

display: block;

margin: 0px auto;

}

Images can also be added to the backgrounds of HTML elements with the background-image property.

body {

background-image: url("https://www.example.com/leaf.jpg");

}

You can control how a background image *repeats* with the background-repeat property. This property can take one of four values:

1. repeat - the default value — the image will repeat horizontally and vertically.
2. repeat-x - the background image will be repeated only along the x-axis (horizontally).
3. repeat-y - the background image will be repeated only along the y-axis (vertically).
4. no-repeat - the background image will not be repeated at all and will appear only once.

p {

background-image: url("#");

background-repeat: repeat-x;

}

When a background image is not repeated, its position can be modified with the background-position property.

1. left top - top left corner of the element's box.
2. center top - top center of the element's box.
3. right top - top right corner of the element's box.
4. left center - left column, center row.
5. center center - the center of the element's box.
6. right center - right column, center row.
7. left bottom - bottom left corner of the element's box.
8. center bottom - bottom center of the element's box.
9. right bottom - bottom right corner of the element's box.

p {

background-image: url("#");

background-repeat: no-repeat;

background-position: right center;

}

The code in the example above can be shortened using the background property.

p {

background: url("#") no-repeat right center;

}

To modify a background image's size, you can use the background-size property.

This exercise will focus on two of the most common values of the background-size property:

1. cover - expands the image as large as possible to cover the full width or height of a container. If the dimensions of the container (say, a div) are larger than the dimensions of the image, the image will become distorted. This value is best for images that don't communicate important content to the user, like background images.
2. contain - expands the image as large as possible, but the image will be [letterboxed](https://www.codecademy.com/courses/learn-html-css/lessons/adding-images/exercises/background-size?action=lesson_resume&link_content_target=interstitial_lesson), which means it won't cover the full width  or height of a container.

div.header {

height: 400px;

width: 100%;

background: url("#") no-repeat right center;

background-size: cover;

}

With the background-attachment property, you can specify whether or not a background image should remain at one position on the web page or whether it should move up and down as the user scrolls through a web page.

The background-attachment property can take one of two values:

1. scroll - this value allows the image to move up and down as a user scrolls on the web page (this is the default value).
2. fixed - this value pins the image's position on the page.

p {

background: url("#") no-repeat right center;

background-attachment: fixed;

}

The background-image property can be set to the following value:

1. -webkit-linear-gradient() - this value accepts two arguments: the two colors the linear gradient will transition to and from. The colors are usually specified as hex color codes.

div.header {

height: 400px;

width: 400px;

background-image: -webkit-linear-gradient(#666CCC, #BC1324);

}

The space between the contents of a box and the borders of a box is known as *padding*. In CSS, you can modify this space with the padding property.

p {

border: 3px solid #A2D3F4;

padding: 10px;

}

Another version of the padding property lets you specify exactly how much padding there should be on each side of the content.

p {

border: 3px solid #XXXXXX;

padding: 5px 10px 5px 10px; <!—top,left,bottom,right-->

}

If you want to be even more specific about the amount of padding on each side of a box's content, you can use the following properties:

1. padding-top
2. padding-right
3. padding-bottom
4. padding-left

Each property affects the padding on only one side of the box's content, giving you more flexibility in customization.

p {

border: 3px solid #2D3FA3;

padding-bottom: 10px;

}

The margin refers to the space directly outside of the box. You can adjust this spacing with the margin property.

p {

border: 1px solid #23AD44;

margin: 20px;

}

In that case, another version of the margin property lets you specify exactly how much margin there should be on each side of the box.

p {

margin: 6px 12px 6px 12px;

}

In that case, another version of the margin property lets you specify exactly how much margin there should be on each side of the box.

p {

margin: 6px 12px 6px 12px;

}

The top and bottom values of margin are the same (6px) and the left and right value of margin are also the same (12px).

p {

margin: 6px 12px;

}

If you want to be even more specific about the amount of margin on each side of a box, you can use the following properties:

1. margin-top
2. margin-right
3. margin-bottom
4. margin-left

Each property affects the margin on only one side of the box, giving you more flexibility in customization.

p {

border: 3px solid #2D3FA3;

margin-right: 15px;

}

When the margin property is set to auto, the element being styled will center in the page.

div.headline {

margin: auto;

}

User agent stylesheets often have default CSS rules that set default values for padding and margin.

\* {

margin: 0;

padding: 0;

}

All HTML elements can be classified as one of the following: inline elements or block-level elements.

1. Inline elements - elements that display inline with text, without disrupting the flow of the text (like links).
2. Block-level elements - elements that use an entire line of space in a web page and disrupt the natural flow of text. Most of the common HTML elements are block-level elements (headings, paragraphs, divs, and more).

In CSS, you can change the default behavior of elements with the display property

Modifying the display property of an element can help achieve a desired layout for a web page. The displayproperty can take on one of four values:

1. inline - causes block-level elements (like a div) to behave like an inline element (like a link).
2. block - causes inline elements (like a link) to behave like a block element (like a div).
3. inline-block - causes block-level elements to behave like an inline element, but retain the features of a block-level element.
4. none - removes an element from view. The rest of the web page will act as if the element does not exist.

<ul>

<li>Home</li>

<li>Products</li>

<li>Settings</li>

<li>Inbox</li>

</ul>

li {

display: inline;

}

Elements can be hidden from view with the visibilityproperty.

The visibility property can be set to one of the following values:

1. hidden - hides an element.
2. visible - displays an element.

<ul>

<li>Explore</li>

<li>Connect</li>

<li class="future">Donate</li>

<ul>

.future {

visibility: hidden;

}

CHANGING BOX MODEL

In CSS, the box-sizing property controls the type of box model the browser should use when interpreting a web page.

The default value of this property is content-box. This is the same box model that is affected by border thickness and padding.

we can reset the entire box model and specify a new one: border-box.

\* {

box-sizing: border-box;

}