

V-EX TECH

Web Development

Java / Node.js / PHP / .Net / Python

Certification Course

**Assured Placement Program
With International Certificate**

About V-Ex Tech....

V-Ex Tech is an elevated education platform providing rigorous industry-relevant programs

Designed and delivered on collaboration with industry professionals. It has been constantly

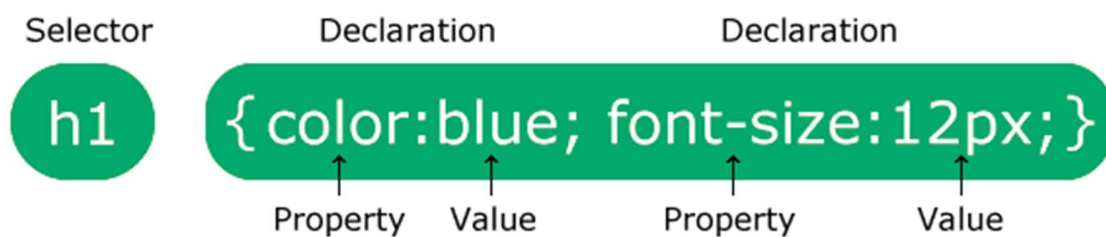
Into process of creating an immersive learning experience binding latest technologies, pedagogy

and services with enormous job placement opportunities too.

Css Course

Why Use CSS?

CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.



CSS Selectors

CSS element Selector

```
p {  
  text-align: center;  
  color: red;  
}
```

CSS id Selector

```
#para1 {  
  text-align: center;  
  color: red;  
}
```

CSS class Selector

```
.center {  
  text-align: center;  
  color: red;  
}
```

CSS Universal Selector

```
* {  
  text-align: center;  
  color: blue;  
}
```

CSS Grouping Selector

```
h1, h2, p {  
  text-align: center;  
  color: red;  
}
```

Three Ways to Insert CSS

External CSS

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Internal CSS

```
<html>
<head>
<style>
body {
    background-color: linen;
}

h1 {
    color: maroon;
    margin-left: 40px;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Inline CSS

```
<!DOCTYPE html>
<html>
<body>

<h1 style="color:blue;text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>

</body>
</html>
```

Multiple Style Sheets

CSS Border Color

Hello World

Hello World

Hello World

Example

```
<h1 style="border:2px solid Tomato;">Hello World</h1>
<h1 style="border:2px solid DodgerBlue;">Hello World</h1>
<h1 style="border:2px solid Violet;">Hello World</h1>
```

CSS Backgrounds

- background-color
-
- background-image
-
- background-repeat
-
- background-attachment
-
- background-position

```
body {  
  background-color: #ffffff;  
  background-image: url("img_tree.png");  
  background-repeat: no-repeat;  
  background-position: right top;  
}
```

CSS Borders

- **dotted** - Defines a dotted border
- **dashed** - Defines a dashed border
- **solid** - Defines a solid border
- **double** - Defines a double border
- **groove** - Defines a 3D grooved border. The effect depends on the border-color value

- **ridge** - Defines a 3D ridged border. The effect depends on the border-color value
- **inset** - Defines a 3D inset border. The effect depends on the border-color value
- **outset** - Defines a 3D outset border. The effect depends on the border-color value
- **none** - Defines no border
- **hidden** - Defines a hidden border

CSS Margins

Margins are used to create space around elements, outside of any defined borders.

```
<style>
div {
  border: 1px solid black;
  margin-top: 100px;
  margin-bottom: 100px;
  margin-right: 150px;
  margin-left: 80px;
  background-color: lightblue;
}
</style>
</head>
<body>
```

<h2>Using individual margin properties</h2>


```
<div>This div element has a top margin of 100px, a right margin of  
150px, a bottom margin of 100px, and a left margin of 80px.</div>
```

```
</body>
```

CSS Padding

Padding is used to create space around an element's content, inside of any defined borders.

```
<style>  
div {  
  padding: 70px;  
  border: 1px solid #4CAF50;  
}  
</style>  
</head>  
<body>  
  
<h2>CSS Padding</h2>  
<div>This element has a padding of 70px.</div>  
  
</body>
```

CSS Height, Width and Max-width

The CSS `height` and `width` properties are used to set the height and width of an element.

The CSS `max-width` property is used to set the maximum width of an element.

CSS Text Alignment

```
h1 {  
  text-align: center;  
}
```

```
h2 {  
  text-align: left;  
}
```

```
h3 {  
  text-align: right;  
}
```

Text Direction

```
p {  
  direction: rtl;  
  unicode-bidi: bidi-override;  
}
```

Text Decoration

```
<style>
h1 {
  text-decoration: overline;
}

h2 {
  text-decoration: line-through;
}

h3 {
  text-decoration: underline;
}

p.ex {
  text-decoration: overline underline;
}
</style>
</head>
<body>

<h1>Overline text decoration</h1>
<h2>Line-through text decoration</h2>
<h3>Underline text decoration</h3>
<p class="ex">Overline and underline text decoration.</p>

<p><strong>Note:</strong> It is not recommended to underline text that is not a link, as this often
confuses
the reader.</p>
```

CSS Text Transformation

```
p.uppercase {  
  text-transform: uppercase;  
}
```

```
p.lowercase {  
  text-transform: lowercase;  
}
```

```
p.capitalize { text-transform: capitalize;  
}
```

CSS Text Spacing

```
text-indent  
letter-spacing  
text-align: justify;  
line-height  
word-spacing  
white-space
```

CSS Text Shadow

Example

```
h1 {  
  text-shadow: 2px 2px;  
}
```

CSS Fonts

CSS font-family Property

```
.p1 {  
  font-family: "Times New Roman", Times, serif;  
}  
  
.p2 {  
  font-family: Arial, Helvetica, sans-serif;  
}  
  
.p3 {  
  font-family: "Lucida Console", "Courier New", monospace;  
}
```

CSS Links

- [Text Link](#) [Text Link](#) [Link Button](#) [Link Button](#)

-
-

- **a:link** - a normal, unvisited link
- **a:visited** - a link the user has visited
- **a:hover** - a link when the user mouses over it
- **a:active** - a link the moment it is clicked

```
a:link {  
  color: red;
```

```
}

/* visited link */
a:visited {
    color: green;
}

/* mouse over link */
a:hover {
    color: hotpink;
}

/* selected link */
a:active {
    color: blue;
}
```

Text Decoration

```
a:link {
    text-decoration: none;
}
```

CSS Lists

```
<style>
ul {
    list-style-image: url('sqpurple.gif');
}
</style>
</head>
<body>
```

<h2>The list-style-image Property</h2>

<p>The list-style-image property specifies an image as the list item marker:</p>


```
<li>Coffee</li>  
<li>Tea</li>  
<li>Coca Cola</li>  
</ul>
```

```
</body>
```

`visibility:hidden;`

`display` property to `none`

width and max-width

The position Property

There are five different position values:

- static
- relative
- fixed
- absolute
- sticky

position: static;

```
div.static {  
  position: static;  
  border: 3px solid #73AD21;  
}
```

position: relative;

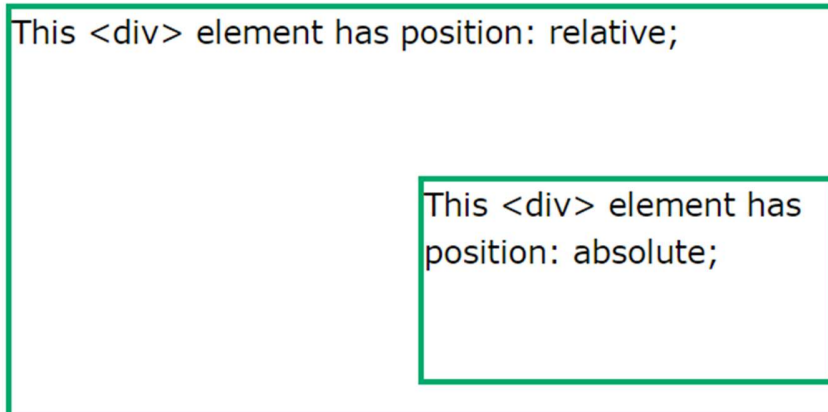
```
position: relative;  
  
left: 30px;  
  
border: 3px solid #73AD21;
```

position: fixed;

```
div.fixed {  
  position: fixed;  
  bottom: 0;  
  right: 0;  
  width: 300px;  
  border: 3px solid #73AD21;  
}
```


position: absolute;

Here is a simple example:



```
<style>
div.relative {
  position: relative;
  width: 400px;
  height: 200px;
  border: 3px solid #73AD21;
}
```

```
div.absolute {
  position: absolute;
  top: 80px;
```

```
right: 0;
width: 200px;
height: 100px;
border: 3px solid #73AD21;
}
</style>
</head>
<body>
```

```
<h2>position: absolute;</h2>
```

<p>An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed):</p>

```
<div class="relative">This div element has position: relative;
  <div class="absolute">This div element has position: absolute;</div>
</div>

</body>
```

position: sticky;

```
<style>
div.sticky {
  position: -webkit-sticky;
  position: sticky;
  top: 0;
  padding: 5px;
  background-color: #cae8ca;
  border: 2px solid #4CAF50;
}
</style>
</head>
<body>
```

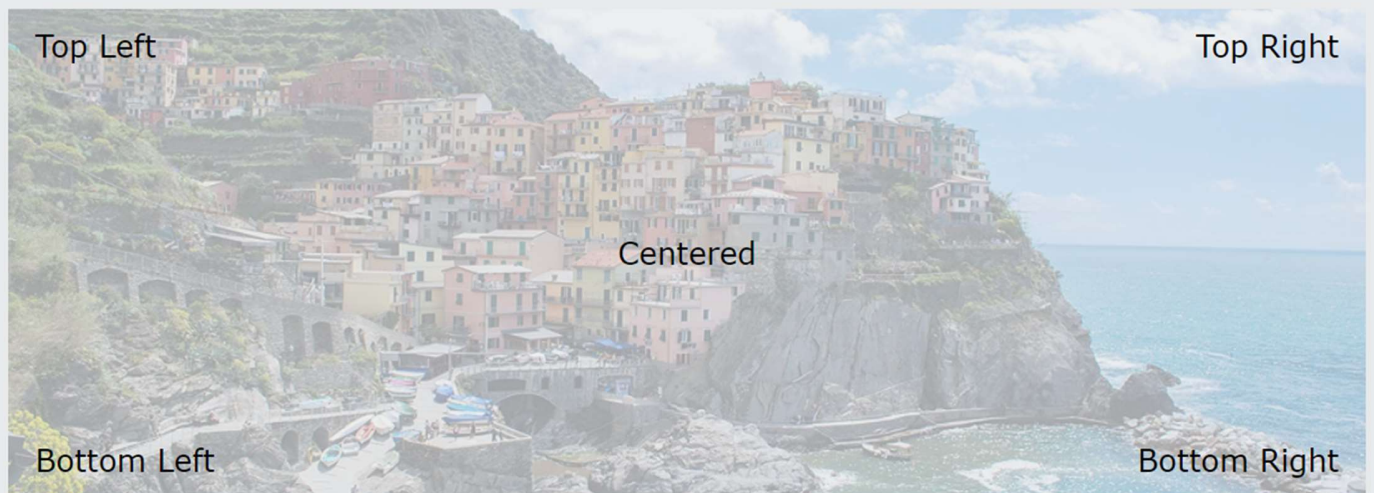
<p>Try to scroll inside this frame to understand how sticky positioning works.</p>

```
<div class="sticky">I am sticky!</div>
```

```
<div style="padding-bottom:2000px">
  <p>In this example, the sticky element sticks to the top of the page (top: 0), when you reach its scroll position.</p>
  <p>Scroll back up to remove the stickyness.</p>
  <p>Some text to enable scrolling.. Lorem ipsum dolor sit amet, illum definitiones no quo, maluisset concludaturque et eum, altera fabulas ut quo. Atqui causae gloriatur ius te, id agam omnis evertitur eum. Affert laboramus repudiandae nec et. Inciderint efficiantur his ad. Eum no molestiae voluptatibus.</p>
  <p>Some text to enable scrolling.. Lorem ipsum dolor sit amet, illum definitiones no quo, maluisset concludaturque et eum, altera fabulas ut quo. Atqui causae gloriatur ius te, id agam omnis evertitur eum. Affert laboramus repudiandae nec et. Inciderint efficiantur his ad. Eum no molestiae voluptatibus.</p>
</div>

</body>
```

Example



Positioning Text In an Image

The z-index Property

This is a heading

Because the image has a z-index of -1, it will be placed behind the text.



```
img {  
  position: absolute;  
  left: 0px;  
  top: 0px;  
  z-index: -1;  
}
```

Overflow

overflow: visible

```
div {  
  width: 200px;  
  height: 65px;  
  background-color: coral;  
  overflow: visible;  
}
```

overflow: hidden

```
div {  
  width: 200px;  
  height: 65px;  
  background-color: coral;  
  overflow: hidden;  
}
```

overflow: scroll

```
div {  
  width: 200px;  
  height: 65px;  
  background-color: coral;  
  overflow: scroll;  
}
```

float

float: right;

float: left;

clear and clearfix

```
<style>
  .div1 {
    float: left;
    padding: 10px;
    border: 3px solid #73AD21;
  }

  .div2 {
    padding: 10px;
    border: 3px solid red;
    clear: left;
  }
</style>
<body>

  <div class="div1">div1</div>
  <div class="div2">div2</div>

</body>
```

Clearfix

```
<style>
div {
  border: 3px solid #4CAF50;
  padding: 5px;
}

.img1 {
  float: right;
}

.img2 {
  float: right;
}

.clearfix {
  overflow: auto;
}
</style>
</head>
<body>

<div>
  
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet...
</div>

<div class="clearfix">
  
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet...
</div>

</body>
```


Example

```
<style>
  * {
    box-sizing: border-box;
  }

  .box {
    float: left;
    width: 50%;
    padding: 50px;
    height: 300px;
  }

</style>
</head>
<body>

<h2>Equal Height Boxes</h2>
<p>Floating boxes with equal heights:</p>

<div class="clearfix">
  <div class="box" style="background-color:#bbb">
    <h2>Box 1</h2>
    <p>Some content, some content, some content</p>
  </div>
  <div class="box" style="background-color:#ccc">
    <h2>Box 2</h2>
    <p>Some content, some content, some content</p>
    <p>Some content, some content, some content</p>
    <p>Some content, some content, some content</p>
  </div>
</div>

</body>
```

Flex property

```
<style>
.flex-container {
  display: flex;
  flex-wrap: nowrap;
  background-color: DodgerBlue;
}

.flex-container .box {
  background-color: #f1f1f1;
  width: 50%;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>

<h1>Flexible Boxes</h1>

<div class="flex-container">
  <div class="box">Box 1 - This is some text </div>
  <div class="box">Box 2 - My height will follow Box 1.</div>
</div>

</body>
```

Practise

Chania

The Flight

The City

The Island

The Food

The City

Chania is the capital of the Chania region on the island of Crete. The city can be divided in two parts, the old town and the modern city.

You will learn more about web layout and responsive web pages in a later chapter.

Using inline-block to Create Navigation Links

```
<style>
.nav {
  background-color: yellow;
  list-style-type: none;
  text-align: center;
  margin: 0;
  padding: 0;
}

.nav li {
  display: inline-block;
  font-size: 20px;
  padding: 20px;
}
</style>
</head>
<body>

<ul class="nav">
  <li><a href="#home">Home</a></li>
  <li><a href="#about">About Us</a></li>
  <li><a href="#clients">Our Clients</a></li>
  <li><a href="#contact">Contact Us</a></li>
</ul>

</body>
```

Center Align Elements

```
<style>
  .center {
    margin: auto;
    width: 60%;
    border: 3px solid #73AD21;
    padding: 10px;
  }
</style>
</head>
<body>

  <div class="center">
    <p>Hello World!</p>
  </div>

</body>
```

Center Align Text

```
<style>
  .center {
    text-align: center;
    border: 3px solid green;
  }
</style>
</head>
<body>

  <div class="center">
    <p>This text is centered.</p>
  </div>

</body>
```

Center an Image

```
<style>
  img {
    display: block;
    margin-left: auto;
    margin-right: auto;
  }
</style>
</head>
<body>

</body>
```

Left and Right Align - Using position

```
<style>
  .right {
    position: absolute; //float right
    right: 0px;
    width: 300px;
    border: 3px solid #73AD21;
    padding: 10px;
  }
</style>
</head>
<body>

  <div class="right">
    <p>In my younger and more vulnerable years my father gave me some
advice that I've been turning over in my mind ever since.</p>
  </div>

</body>
```


Center vertically

```
<style>
  .center {
    display: flex;
    justify-content: center;
    align-items: center;
    height: 200px;
    border: 3px solid green;
  }
</style>
</head>
<body>

  <div class="center">
    <p>I am vertically and horizontally centered.</p>
  </div>

</body>
```



CSS Combinators

A combinator is something that explains the relationship between the selectors.

There are four different combinators in CSS:

- descendant selector (space)
- child selector (>)
- adjacent sibling selector (+)
- general sibling selector (~)

Descendant Selector

```
<style>
  div p {
    background-color: yellow;
  }
</style>

<body>

  <h2>Descendant Selector</h2>

  <p>The descendant selector matches all elements that are
  descendants of a specified element.</p>

  <div>
    <p>Paragraph 1 in the div.</p>
    <p>Paragraph 2 in the div.</p>
    <section><p>Paragraph 3 in the div.</p></section>
  </div>

  <p>Paragraph 4. Not in a div.</p>
  <p>Paragraph 5. Not in a div.</p>

</body>
```

Child Selector (>)

```
<style>
div > p {
  background-color: yellow;
}
</style>

<body>

<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
  <section>
    <!-- not Child but Descendant -->
    <p>Paragraph 3 in the div (inside a section element).</p>
  </section>
  <p>Paragraph 4 in the div.</p>
</div>

<p>Paragraph 5. Not in a div.</p>
<p>Paragraph 6. Not in a div.</p>

</body>
```

Adjacent Sibling Selector (+)

```
<style>
  div + p {
    background-color: yellow;
  }
</style>

<body>

  <div>
    <p>Paragraph 1 in the div.</p>
    <p>Paragraph 2 in the div.</p>
  </div>

  <p>Paragraph 3. After a div.</p>
  <p>Paragraph 4. After a div.</p>

  <div>
    <p>Paragraph 5 in the div.</p>
    <p>Paragraph 6 in the div.</p>
  </div>

  <p>Paragraph 7. After a div.</p>
  <p>Paragraph 8. After a div.</p>

</body>
```

General Sibling Selector (~)

```
<style>
  div ~ p {
    background-color: yellow;
  }
</style>

<body>

  <p>Paragraph 1.</p>

  <div>
    <p>Paragraph 2.</p>
  </div>
  <p>Paragraph 3.</p>

  <code>Some code.</code>
  <p>Paragraph 4.</p>

</body>
```

Practice: use all combinator in 5 div

CSS Pseudo-classes

Syntax

```
selector:pseudo-class {  
  property: value;  
}
```

Pseudo-classes and HTML Classes

```
<style>  
  p:hover{  
    font-size: 48px;  
    color: brown;  
  }  
</style>  
  
<body>  
  <p>Hello Student</p>  
</body>
```

```
<style>
  div {
    background-color: green;
    color: white;
    padding: 25px;
    text-align: center;
  }

  div:hover {
    background-color: blue;
  }
</style>

<body>

  <div>Mouse Over Me</div>

</body>
```


Simple Tooltip Hover

```
<style>
  p {
    display: none;
    background-color: yellow;
    padding: 20px;
  }

  div:hover p {
    display: block;
  }
</style>

<body>

  <div>Hover over this div element to show the p element
    <p>Tada! Here I am!</p>
  </div>

</body>
```

```
<style>
  p:first-child {
    color: blue;
  }
</style>

<body>

<p>This is some text.</p>
<p>This is some text.</p>

<div>
  <p>This is some text.</p>
  <p>This is some text.</p>
</div>

</body>
```

```
<style>
p i:first-child {
  color: blue;
}
</style>
</head>
<body>

<p>I am a <i>strong</i> person. I am a <i>strong</i>
person.</p>
<p>I am a <i>strong</i> person. I am a <i>strong</i>
person.</p>

</body>
```

```
<style>
p:first-child i {
  color: blue;
}
</style>
<body>

<p>I am a <i>strong</i> person. I am a <i>strong</i>
person.</p>
<p>I am a <i>strong</i> person. I am a <i>strong</i>
person.</p>

<div>
  <p>I am a <i>strong</i> person. I am a <i>strong</i>
person.</p>
  <p>I am a <i>strong</i> person. I am a <i>strong</i>
person.</p>
</div>

</body>
```

```
<style>
p i:nth-child(2) {
  color: blue;
}
</style>
</head>
<body>

<p>I am a <i>strong</i> person. I am a <i>strong</i>
person.</p>
<p>I am a <i>strong</i> person. I am a <i>strong</i>
person.</p>

<div>
  <p>I am a <i>strong</i> person. I am a <i>strong</i>
person.</p>
  <p>I am a <i>strong</i> person. I am a <i>strong</i>
person.</p>
</div>

</body>
```

CSS Pseudo-elements

Syntax

```
selector::pseudo-element {  
  property: value;  
}
```

The ::first-line Pseudo-element

```
<style>  
  p::first-line {  
    color: #ff0000;  
    font-variant: small-caps;  
  }  
</style>  
  
<body>  
  
  <p>You can use the ::first-line pseudo-element to add a  
special effect to the first line of a text. Some more text. And  
even more, and more, and more, and more, and more, and more,  
and more, and more, and more, and more, and more, and more.</p>  
  
</body>
```

The ::first-letter Pseudo-element

```
<style>
  p::first-letter {
    color: #ff0000;
    font-size: xx-large;
  }
</style>

<body>

  <p>You can use the ::first-letter pseudo-element to add a
special effect to the first character of a text!</p>

</body>
```

CSS - The ::before Pseudo-element

```
<style>
  h1::before {
    content: url(smiley.gif);
  }
</style>
</head>
<body>

  <h1>This is a heading</h1>

  <h1>This is a heading</h1>

</body>
```


CSS - The ::after Pseudo-element

```
<style>
  h1::after {
    content: url(smiley.gif);
  }
</style>
</head>
<body>

  <h1>This is a heading</h1>

  <h1>This is a heading</h1>

</body>
```

CSS - The ::marker Pseudo-element

```
<style>
  ::marker {
    color: red;
    font-size: 23px;
  }
</style>
</head>
<body>

<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>

<ol>
  <li>First</li>
  <li>Second</li>
  <li>Third</li>
</ol>

</body>
```

CSS Opacity / Transparency

Transparent Image

```
<style>
  img {
    opacity: 0.5;
  }
</style>
<body>

</body>
```

```
<style>
  img {
    opacity: 0.5;
  }

  img:hover {
    opacity: 1.0;
  }
</style>

<body>

</body>
```



Navigation Bar

Navigation Bar = List of Links

Vertical Navigation Bar

```
<style>
  ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
    width: 200px;
    background-color: #f1f1f1;
  }

  li a {
    display: block;
    color: #000;
    padding: 8px 16px;
    text-decoration: none;
```

```
}

/* Change the link color on hover */
li a:hover {
    background-color: #555;
    color: white;
}
</style>

<body>

<ul>
    <li><a href="#home">Home</a></li>
    <li><a href="#news">News</a></li>
    <li><a href="#contact">Contact</a></li>
    <li><a href="#about">About</a></li>
</ul>

</body>
```

Active/Current Navigation Link

```
li a.active {

    background-color: #04AA6D;

    color: white;

}
```

Horizontal Navigation Bar

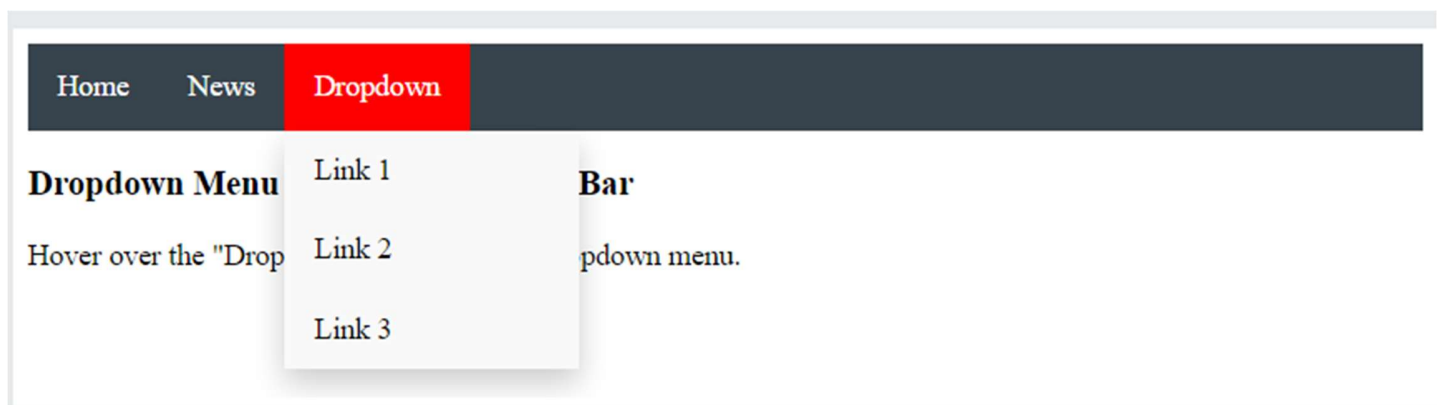
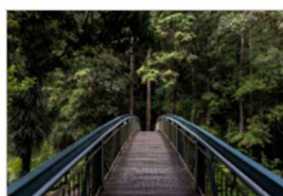


Image dropdown

task



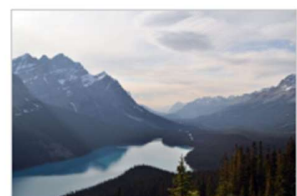
Add a description
of the image here



Add a description
of the image here



Add a description
of the image here



Add a description
of the image here

Image Sprites

Image Sprites - Simple Example

Instead of using three separate images, we use this single image ("img_navsprites.gif"):



Attribute Selectors

[attribute="value"] Selector


```
<style>
  a[target="_blank"] {
    background-color: yellow;
  }
</style>

<body>

  <a href="https://www.w3schools.com">w3schools.com</a>
  <a href="http://www.disney.com"
target="_blank">disney.com</a>

</body>
```

[attribute~="value"] Selector

```
<style>
  [title~="flower"] {
    border: 5px solid yellow;
  }
</style>

<body>

</body>
```

[attribute]="value"] Selector

```
<style>
  [class="top"] {
    background: yellow;
  }
</style>

<body>

  <h1 class="top-header">Welcome</h1>
  <p class="top-text">Hello world!</p>
  <p class="topcontent">Are you learning CSS?</p>

</body>
```

[attribute^="value"] Selector

```
<style>
  [class^="top"] {
    background: yellow;
  }
</style>

<body>

  <h1 class="top-header">Welcome</h1>
  <p class="top-text">Hello world!</p>
  <p class="topcontent">Are you learning CSS?</p>

</body>
```

Styling Forms

```
<style>
  input[type="text"] {
    width: 150px;
    display: block;
    margin-bottom: 10px;
    background-color: yellow;
  }

  input[type="button"] {
    width: 120px;
    margin-left: 35px;
    display: block;
  }
</style>

<body>

  <form name="input" action="" method="get">
    Firstname:<input type="text" name="Name" value="Peter"
size="20">Lastname:<input type="text" name="Name"
value="Griffin" size="20">
```

```
<input type="button" value="Example Button">
```

First Name

Last Name

Country



Try it Yourself »

```
</form>
```

```
</body>
```

task

```
<style>
input[type=text] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  box-sizing: border-box;
  border: 1px solid #555;
  outline: none;
}

input[type=text]:focus {
  background-color: lightblue;
}
</style>

<body>
  <form>
    <label for="fname">First Name</label>
    <input type="text" id="fname" name="fname"
value="John">
    <label for="lname">Last Name</label>
    <input type="text" id="lname" name="lname"  </form>
  </body>
```

```
<style>
  input[type=text] {
    width: 100%;
    box-sizing: border-box;
    border: 2px solid #ccc;
    border-radius: 4px;
    font-size: 16px;
    background-color: white;
    background-image: url('searchicon.png');
    background-position: 10px 10px;
    background-repeat: no-repeat;
    padding: 12px 20px 12px 40px;
  }
</style>

<body>

  <form>
    <input type="text" name="search" placeholder="Search..">
  </form>
</body>
```


Animated Search Input

```
<style>
  input[type=text] {
    width: 130px;
    box-sizing: border-box;
    border: 2px solid #ccc;
    border-radius: 4px;
    font-size: 16px;
    background-color: white;
    background-image: url('searchicon.png');
    background-position: 10px 10px;
    background-repeat: no-repeat;
    padding: 12px 20px 12px 40px;
    transition: width 0.4s ease-in-out;
  }

  input[type=text]:focus {
    width: 100%;
  }
</style>

<body>

  <form>
    <input type="text" name="search" placeholder="Search..">
  </form>
```

```
</body>
```

Styling Select Menus

```
<style>
  select {
    width: 100%;
    padding: 16px 20px;
    border: none;
    border-radius: 4px;
    background-color: #f1f1f1;
  }
</style>

<body>

  <form>
    <select id="country" name="country">
      <option value="au">Australia</option>
      <option value="ca">Canada</option>
      <option value="usa">USA</option>
    </select>
  </form>

</body>
```

V-Ex Tech

Practice

STUDENT REGISTRATION FORM

FIRST NAME: (max 30 characters A-Z and a-z)

LAST NAME: (max 30 characters A-Z and a-z)

DATE OF BIRTH:

EMAIL ID:

MOBILE NO: (10 digits number)

GENDER: ☐ Female ☐ Male

ADDRESS:

CITY:

PIN CODE: (6 digits number)

STATE:

COUNTRY:

HOBBIES ☐ Singing ☐ Dancing ☐ Drawing ☐ Sketching
☐ Others

S.NO	Examination	Board	Percentage	Year of Passing
1.	Class X	<input type="text"/>	<input type="text"/>	<input type="text"/>
2.	Class XII	<input type="text"/>	<input type="text"/>	<input type="text"/>
3.	Graduation	<input type="text"/>	<input type="text"/>	<input type="text"/>
4.	Masters	<input type="text"/>	<input type="text"/>	<input type="text"/>

(10 char max) (upto to decimal)

COURSES: ☐ BCA ☐ B.Com ☐ B.Sc ☐ B.A

CSS Counters

Automatic Numbering With Counters

- `counter-reset` - Creates or resets a counter
- `counter-increment` - Increments a counter value
- `content` - Inserts generated content
- `counter()` or `counters()` function - Adds the value of a counter to an element

```
• <style>
•   body {
•     counter-reset: section;
•   }
•
•   h2::before {
•     counter-increment: section;
•     content: "Section " counter(section) ": ";
•   }
• </style>
•
• <body>
•
•   <h1>Using CSS Counters</h1>
•
•   <h2>HTML Tutorial</h2>
•   <h2>CSS Tutorial</h2>
•   <h2>JavaScript Tutorial</h2>
•   <h2>Python Tutorial</h2>
•   <h2>SQL Tutorial</h2>
•
• </body>
```

Nesting Counters

```
<style>
  body {
    counter-reset: section;
  }

  h1 {
    counter-reset: subsection;
  }

  h1::before {
    counter-increment: section;
    content: "Section " counter(section) ". ";
  }

  h2::before {
    counter-increment: subsection;
    content: counter(section) "." counter(subsection) " ";
  }
</style>
</head>
<body>

<h1>HTML/CSS Tutorials</h1>
<h2>HTML</h2>
<h2>CSS</h2>
<h2>Bootstrap</h2>
<h2>W3.CSS</h2>
```

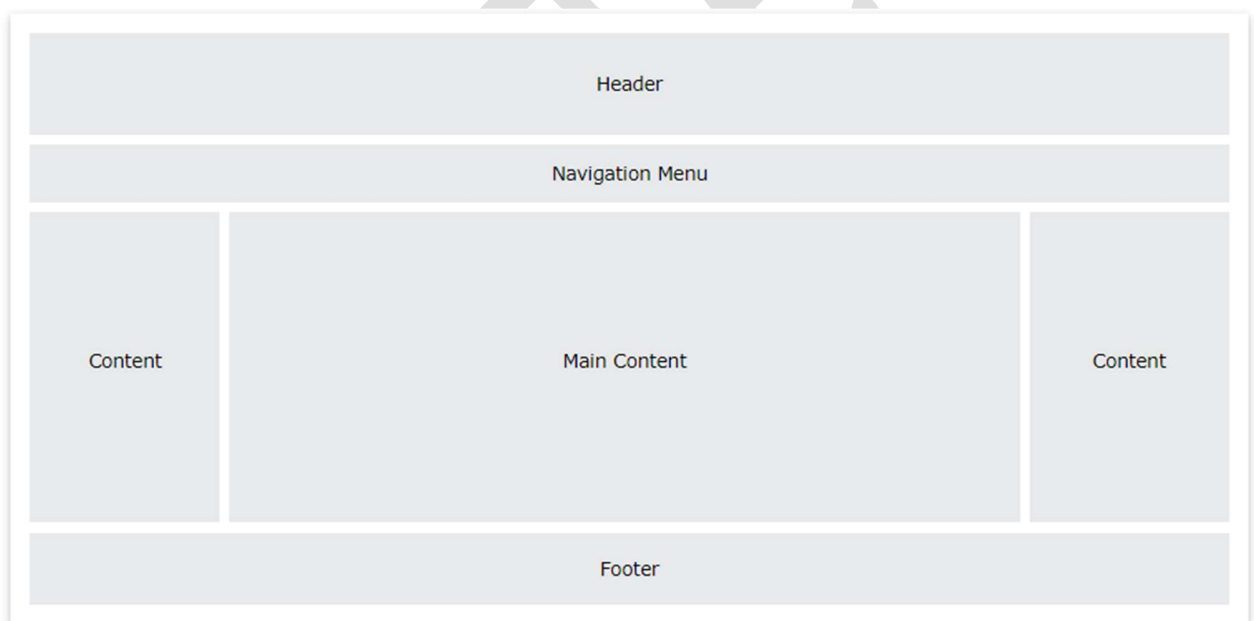
```
<h1>Scripting Tutorials</h1>
<h2>JavaScript</h2>
<h2>jQuery</h2>
<h2>React</h2>

<h1>Programming Tutorials</h1>
<h2>Python</h2>
<h2>Java</h2>
<h2>C++</h2>

</body>
```

Website Layout

task



Task

https://www.w3schools.com/css/css_website_layout.asp

CSS Units

Many CSS properties take "length" values, such as **width**, **margin**, **padding**, **font-size**, etc.

Rem,px,vh,%

CSS Specificity

highest priority

```
<style>
.test {color: green;}
p {color: red;}
</style>
</head>
<body>
<p class="test">Hello World!</p></body>
```

```
<style>
#demo {color: blue;}
.test {color: green;}
p {color: red;}
</style>
</head>
<body>

<p id="demo" class="test">Hello World!</p>

</body>
```

```
<style>
#demo {color: blue;}
.test {color: green;}
p {color: red;}
</style>
</head>
<body>

<p id="demo" class="test" style="color: pink;">Hello World!</p>

</body>
```

Selector	Specificity Value	Calculation
p	1	1
p.test	11	1 + 10
p#demo	101	1 + 100
<p style="color: pink;">	1000	1000
#demo	100	100
.test	10	10
p.test1.test2	21	1 + 10 + 10
#navbar p#demo	201	100 + 1 + 100
*	0	0 (the universal selector is ignored)

The selector with the highest specificity value will win and take effect!

```
<style>
h1 {background-color: yellow;}
h1 {background-color: red;}
</style>
</head>
<body>

<h1>This is heading 1</h1>

</body>
```

The !important Rule

What is !important?

The **!important** rule in CSS is used to add more importance to a property/value than normal.

```
<style>
#myid {
  background-color: blue;
}

.myclass {
  background-color: gray;
}

p {
  background-color: red !important;
}
</style>
</head>
<body>

<p>This is some text in a paragraph.</p>

<p class="myclass">This is some text in a paragraph.</p>

<p id="myid">This is some text in a paragraph.</p>

</body>
```

CSS Math Functions

The calc() Function

```
<style>
#div1 {
  position: absolute;
  left: 50px;
  width: calc(100% - 300px);
  border: 1px solid black;
  background-color: yellow;
  padding: 5px;
}
</style>
</head>
<body>

<h1>The calc() Function</h1>

<p>Create a div that stretches across the window, with a 50px
gap between both sides of the div and the edges of the
window:</p>

<div id="div1">Some text...</div>

</body>
```

The max() Function

```
<style>
#div1 {
  background-color: yellow;
  height: 100px;
  width: max(50%, 300px);
}
</style>
</head>
<body>

<h1>The max() Function</h1>

<p>Use max() to set the width of #div1 to whichever value is
largest, 50% or 300px:</p>

<div id="div1">Some text...</div>

<p>Resize the browser window to see the effect.</p>

</body>
```

The min() Function

```
<style>
#div1 {
  background-color: yellow;
  height: 100px;
  width: min(50%, 300px);
}
</style>
</head>
<body>

<h1>The min() Function</h1>

<p>Use min() to set the width of #div1 to whichever value is
smallest, 50% or 300px:</p>

<div id="div1">Some text...</div>

<p>Resize the browser window to see the effect.</p>

</body>
```