

CSS Box Model

All HTML elements can be considered as boxes and consists of:



❖ CSS Text

CSS has a lot of properties for formatting text.

- **Text Color**
- **CSS Text Alignment and Text Direction**

- `text-align`
- `text-align-last`
- `direction`
- `unicode-bidi`
- `vertical-align`

1. Text Alignment

is used to set the horizontal alignment of a text. text can be left or right aligned, centered, or justified

Example

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

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

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

Heading 1 (center)

Heading 2 (left)

Heading 3 (right)

The three headings above are aligned center, left and right.

2. Text Align Last :The `text-align-last` property specifies how to align the last line of a text

Example

```
p.a {  
  text-align-last: right;  
}  
p.b {  
  text-align-last: center;  
}  
p.c {  
  text-align-last: justify;  
}
```

text-align-last: right:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut

text-align-last: center:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.

text-align-last: justify:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut

3. Text Direction :THE `direction` and `unicode-bidi` properties can be used to change the text direction of an element:

Example

```
p.ex1 {  
  direction: rtl;  
  unicode-bidi: bidi-override;  
}
```

This is the default text direction.

.noitcerid txet tfel-ot-thgir si sihT

<p class="ex1">This is right-to-left text direction.</p>


4. Vertical Alignment

The **vertical-align** property sets the vertical alignment of an element.


```
img.a {  
  vertical-align: baseline;  
}  
img.b {  
  vertical-align: text-top;  
}  
img.c {  
  vertical-align: text-bottom;  
}  
img.d {  
  vertical-align: sub;  
}  
img.e {  
  vertical-align: super;  
}
```

The vertical-align Property


vertical-align: baseline (default):

An  image with a default alignment.


vertical-align: text-top:

An  image with a text-top alignment.


vertical-align: text-bottom:

An  image with a text-bottom alignment.

vertical-align: sub:

An  image with a sub alignment.

vertical-align: sup:

An  image with a super alignment.

- text-decoration-line
- text-decoration-color
- text-decoration-style
- text-decoration-thickness
- text-decoration

5. CSS Text Decoration

- ✓ **Add Decoration Line to Text:** The **text-decoration-line** property is used to add a decoration line to text.

```
h1 {  
  text-decoration-line: overline;  
}  
h2 {  
  text-decoration-line: line-through;  
}  
h3 {  
  text-decoration-line: underline;  
}  
p {  
  text-decoration-line: overline underline;  
}
```

- CSS Text Decoration

The **text-decoration-line** property is used to add a decoration line to text.

Example

```
h1 {  
  text-decoration-line: overline;  
}
```

```
h2 {  
  text-decoration-line: line-through;  
}
```

```
h3 {  
  text-decoration-line: underline;  
}
```

```
p {  
  text-decoration-line: overline underline;  
}
```

- **text-decoration-line**
- **text-decoration-color**
- **text-decoration-style**
- **text-decoration-thickness**
- **text-decoration**

The **text-decoration-color** property is used to set the color of the decoration line.

Example

```
h1 {  
  text-decoration-line: overline;  
  text-decoration-color: red;  
}
```

```
h2 {  
  text-decoration-line: line-through;  
  text-decoration-color: blue;  
}
```

```
h3 {  
  text-decoration-line: underline;  
  text-decoration-color: green;  
}
```

```
}
```

```
p {  
  text-decoration-line: overline underline;  
  text-decoration-color: purple;  
}
```

Overline text decoration

Line-through text decoration

Underline text decoration

Overline and underline text decoration.

Note: It is not recommended to underline text that is not a link, as this often confuses the reader.

The **text-decoration-style** property is used to set the style of the decoration line.

```
h1 {  
  text-decoration-line: underline;  
  text-decoration-style: solid;  
}
```

```
h2 {  
  text-decoration-line: underline;  
  text-decoration-style: double;  
}
```

```
h3 {  
  text-decoration-line: underline;  
  text-decoration-style: dotted;  
}
```

```
p.ex1 {  
  text-decoration-line: underline;  
  text-decoration-style: dashed;  
}
```

```
p.ex2 {
```

```
  text-decoration-line: underline;  
  text-decoration-style: wavy;  
}
```

```
p.ex3 {  
  text-decoration-line: underline;  
  text-decoration-color: red;  
  text-decoration-style: wavy;  
}
```

Overline text decoration

Line-through text decoration

Underline text decoration

Overline and underline text decoration.

The **text-decoration-thickness** property is used to set the thickness of the decoration line.

Example

```
h1 {
  text-decoration-line: underline;
  text-decoration-thickness: auto;
}
h2 {
  text-decoration-line: underline;
  text-decoration-thickness: 5px;
}
h3 {
  text-decoration-line: underline;
  text-decoration-thickness: 25%;
}
p {
  text-decoration-line: underline;
  text-decoration-color: red;
  text-decoration-style: double;
}
```

Heading 1

Heading 2

Heading 3

A paragraph.

Another paragraph.

Another paragraph.

The **text-decoration** property is a shorthand property for:

- **text-decoration-line** (required)
- **text-decoration-color** (optional)
- **text-decoration-style** (optional)
- **text-decoration-thickness** (optional)

Example

```
h1 {
  text-decoration: underline;
}
h2 {
  text-decoration: underline red;
}
h3 {
  text-decoration: underline red double;
}
p {
  text-decoration: underline red double 5px;
}
```

Heading 1

Heading 2

Heading 3

A paragraph.

- **CSS Text Transformation:** The **text-transform** property is used to specify uppercase and lowercase letters in a text.

Example

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

p.lowercase {
```

Using the text-transform property

THIS TEXT IS TRANSFORMED TO UPPERCASE.

this text is transformed to lowercase.

This Text Is Capitalized.

```
text-transform: lowercase;
}
```

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

- CSS Text Spacing

- CSS Text Shadow

The `text-shadow` property adds shadow to text.

Example

```
h1 {
  shadow: 2px 2px;
```

Example

```
h1 {
  shadow: 2px 2px red;
}
```

Example

```
h1 {
  text-shadow: 2px 2px 5px red;
}
```

- text-indent
- letter-spacing
- line-height
- word-spacing
- white-space

Text shadow effect!

text-
}

Text shadow effect!

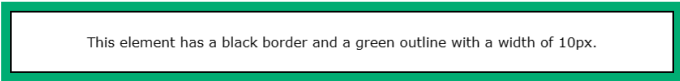
text-

Text shadow effect!

Text shadow effect!

❖ CSS Outline

An outline is a line drawn outside the element's border.



This element has a black border and a green outline with a width of 10px.

```
<style>
p {
  border: 2px solid black;
  outline: #4CAF50 solid 10px;
  margin: auto;
  padding: 20px;
  text-align: center;
}
</style>
```

CSS has the following outline properties:

- **outline-style**
- **outline-color**
- **outline-width**
- **outline-offset**
- **outline**

```
<h2>CSS Outline</h2>
```

```
<p>This element has a 2px black border and a green outline with a width of 10px.</p>
```

- CSS Outline Style

can have one of the following values:

- **dotted** - Defines a dotted outline
- **dashed** - Defines a dashed outline
- **solid** - Defines a solid outline
- **double** - Defines a double outline
- **groove** - Defines a 3D grooved outline
- **ridge** - Defines a 3D ridged outline
- **inset** - Defines a 3D inset outline
- **outset** - Defines a 3D outset outline
- **none** - Defines no outline
- **hidden** - Defines a hidden outline

```
p.dotted {outline-style: dotted;}
p.dashed {outline-style: dashed;}
p.solid {outline-style: solid;}
p.double {outline-style: double;}
p.groove {outline-style: groove;}
p.ridge {outline-style: ridge;}
p.inset {outline-style: inset;}
p.outset {outline-style: outset;}
```

A dotted outline.

A dashed outline.

A solid outline.

A double outline.

A groove outline. The effect depends on the outline-color value.

A ridge outline. The effect depends on the outline-color value.

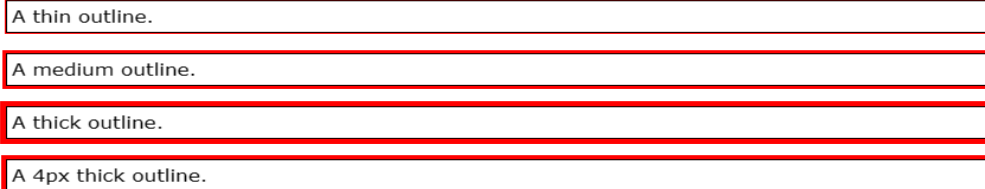
An inset outline. The effect depends on the outline-color value.

An outset outline. The effect depends on the outline-color value.

- CSS Outline Width

can have one of the following values:

- thin (typically 1px)
- medium (typically 3px)
- thick (typically 5px)
- A specific size (in px, pt, cm, em, e



```
p.ex1 {  
  border: 1px solid black;  
  outline-style: solid;  
  outline-color: red;  
  outline-width: thin;  
}
```

```
p.ex3 {  
  border: 1px solid black;  
  outline-style: solid;  
  outline-color: red;  
  outline-width: thick;  
}
```

```
p.ex2 {  
  border: 1px solid black;  
  outline-style: solid;  
  outline-color: red;  
  outline-width: medium;  
}
```

```
p.ex4 {  
  border: 1px solid black;  
  outline-style: solid;  
  outline-color: red;  
  outline-width: 4px;  
}
```

in body:

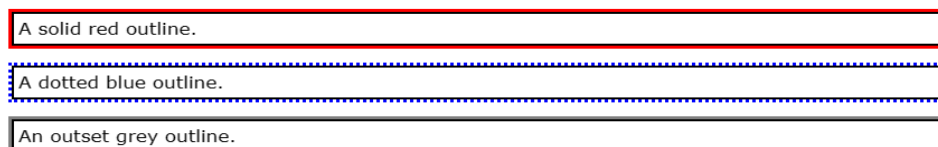
```
<p class="ex1">A thin outline.</p>
```

```
<p class="ex3">A thick outline.</p>
```

```
<p class="ex2">A medium outline.</p>
```

```
<p class="ex4">A 4px thick outline.</p>
```

- **CSS Outline Color:** The `outline-color` property is used to set the color of the outline.



```
p.ex1 {  
  border: 2px solid black;  
  outline-style: solid;  
  outline-color: red;  
}
```

```
p.ex3 {  
  border: 2px solid black;  
  outline-style: outset;  
  outline-color: grey;  
}
```

```
p.ex2 {  
  border: 2px solid black;  
  outline-style: dotted;  
  outline-color: blue;  
}
```

- **CSS Outline - Shorthand property:** The `outline` property is a shorthand property for setting the following individual outline properties:

```
p.ex1 {outline: dashed;}
p.ex2 {outline: dottedred;}
p.ex3 {outline: 5px solid yellow;}

p.ex4 {outline: thick ridge pink;}
```

- `outline-width`
- `outlinestyle` (require)
- `outline-color`

- CSS Outline Offset:

خاصية `outline-offset` تضيف مسافة بين الخط الخارجي (`outline`) وحافة العنصر أو حدوده. المسافة بين العنصر والخط الخارجي شفافة

This paragraph has an outline 15px outside the border edge.

```
p {
  margin: 30px;
  border: 1px solid black;
  outline: 1px solid red;
  outline-offset: 15px;
}
```

i. CSS Fonts

Font Selection is Important

Generic Font Families

In CSS there are five generic font families:

1. **Serif** fonts have a small stroke at the edges of each letter. They create a sense of formality and elegance.
2. **Sans-serif** fonts have clean lines (no small strokes attached). They create a modern and minimalistic look.
3. **Monospace** fonts - here all the letters have the same fixed width. They create a mechanical look.
4. **Cursive** fonts imitate human handwriting.
5. **Fantasy** fonts are decorative/playful fonts.

Difference Between Serif and Sans-serif Fonts



Font Examples

Generic Font Family	Examples of Font Names
Serif	Times New Roman, Georgia, Garamond
Sans-serif	Arial, Verdana, Helvetica

Monospace	Courier New, Lucida Console, Monaco
Cursive	Brush Script MT, Lucida Handwriting
Fantasy	Copperplate, Papyrus

CSS font-family Property

In CSS, we use the `font-family` property to specify the font of a text.

```
.p1 {
  font-family: "Times New Roman", Times, serif;
}

.p2 {
  font-family: Arial, Helvetica, sans-serif;
}

.p3 {
  font-family: "Lucida Console", "Courier New", monospace;
}
```

CSS Web Safe Fonts

Web safe fonts are fonts that are universally installed across all browsers and devices.

Fallback Fonts

```
p {
  font-family: Tahoma, Verdana, sans-serif;
}
```

CSS Fallback Fonts

This is a paragraph.

This is another paragraph.

Best Web Safe Fonts for HTML and CSS

best web safe fonts for HTML and CSS:

- Arial (sans-serif)
- Verdana (sans-serif)
- Tahoma (sans-serif)
- Trebuchet MS (sans-serif)

- Times New Roman (serif)
- Georgia (serif)
- Garamond (serif)

- Courier New (monospace)
- Brush Script MT (cursiv)

I. Arial (sans-serif)

```
body {  
  font-family: Arial, sans-serif;  
}
```

Lorem ipsum dolor sit amet

Lorem ipsum dolor sit amet.

0 1 2 3 4 5 6 7 8 9

II. Verdana (sans-serif)

```
body {  
  font-family: Verdana, sans-serif;  
}
```

Lorem ipsum dolor sit amet

Lorem ipsum dolor sit amet.

0 1 2 3 4 5 6 7 8 9

ii. Tahoma (sans-serif)

```
body {  
  font-family: Tahoma, sans-serif;  
}
```

Lorem ipsum dolor sit amet

Lorem ipsum dolor sit amet.

0 1 2 3 4 5 6 7 8 9

III. Trebuchet MS (sans-serif)

```
body {  
  font-family: 'Trebuchet MS', sans-serif;  
}
```

Lorem ipsum dolor sit amet

Lorem ipsum dolor sit amet.

0 1 2 3 4 5 6 7 8 9

iii. Times New Roman (serif)

Lorem ipsum dolor sit amet

Lorem ipsum dolor sit amet.

0 1 2 3 4 5 6 7 8 9

IV. Georgia (serif)

Lorem ipsum dolor sit amet

Lorem ipsum dolor sit amet.

0 1 2 3 4 5 6 7 8 9

i. Garamond (serif)

Lorem ipsum dolor sit amet

Lorem ipsum dolor sit amet.
0 1 2 3 4 5 6 7 8 9

V. Courier New (monospace)

Lorem ipsum dolor sit amet

Lorem ipsum dolor sit amet.
0 1 2 3 4 5 6 7 8 9

VI. Brush Script MT (cursive)

Lorem ipsum dolor sit amet

Lorem ipsum dolor sit amet.
0 1 2 3 4 5 6 7 8 9

- CSS Font Style

The **font-style** property is mostly used to specify italic text.

three values:

- normal - The text is shown normally
- italic - The text is shown in italics
- oblique - The text is "leaning" (oblique is very similar to italic, but less supported)

```
p.normal {  
  font-style: normal;  
}  
p.italic {  
  font-style: italic;  
}  
p.oblique {  
  font-style: oblique;  
}
```

The font-style property

This is a paragraph in normal style.
This is a paragraph in italic style.
This is a paragraph in oblique style.

- Font Weight

```
p.normal {  
  font-weight: normal;  
}
```

```
p.thick {  
  font-weight: bold;  
}
```

- Font Variant

```
p.normal {
  font-variant: normal;
}
```

```
p.small {
  font-variant: small-caps;
}
```

- **CSS Font Size**

The **font-size** property sets the size of the text.

```
h1 {
  font-size: 40px;
}
```

```
body {
  font-size: 100%;
}
```

```
h1 {
  font-size: 2.5em; /* 40px/16=2.5em */
}
```

- **CSS Google Fonts**

If you do not want to use any of the standard fonts in HTML, you can use Google Fonts.

Google Fonts are free to use, and have more than 1000 fonts to choose from.

How To Use Google Fonts

```
<head>
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Sofia"
>
</style>
```

```
body {
  font-family: "Sofia", sans-serif;
}
</style>
</head>
```



- **Enabling Font Effects**

Google has also enabled different font effects that you can use.

```
<head>
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Sofia&effect=fire">
</style>
body {
```

```

    font-family: "Sofia", sans-serif;
    font-size: 30px;
}
</style>
</head>
<body>

<h1 class="font-effect-fire">Sofia on Fire</h1>

</body>

```

Result:



multiple font effects

```

<head>
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Sofia&effect=
neon|outline|emboss|shadow-multiple">
<style>
body {
    font-family: "Sofia", sans-serif;
    font-size: 30px;
}
</style>
</head>
<body>
<h1 class="font-effect-neon">Neon Effect</h1>
<h1 class="font-effect-outline">Outline
Effect</h1>
<h1 class="font-effect-emboss">Emboss
Effect</h1>
<h1 class="font-effect-shadow-multiple">Multiple Shadow Effect</h1>

</body>

```



- **CSS Font Property**

font property is a shorthand property for:

- font-style
- font-variant
- font-weight
- font-size/line-height
- font-family

```
p.a {
  font: 20px Arial, sans-serif;
}
```

❖ **CSS Icons** :Icons can easily be added to your HTML page, by using an icon library.

إضافة أيقونة, اتبع الخطوات التالية:

1. اختر مكتبة أيقونات :هناك العديد من مكتبات الأيقونات المجانية والمدفوعة المتاحة, بما في ذلك Font Awesome, Material Design Icons, and Bootstrap Icons. توفر الأيقونات التي تحتاجها وتتناسب مع تصميم موقع الويب الخاص بك.
2. احصل على كود الأيقونة :بمجرد اختيار مكتبة الأيقونات, ستحتاج إلى الحصول على كود الأيقونة التي تريد استخدامها. عادةً ما توفر مكتبات الأيقونات مواقع ويب أو وثائق حيث يمكنك استعراض الأيقونات المتاحة ونسخ كود الأيقونة المطلوب.
3. أضف كود الأيقونة إلى HTML الخاص بك :بعد الحصول على كود الأيقونة, أضفه إلى عنصر HTML مضمن في صفحتك. غالبًا ما تستخدم العناصر `<i>` أو `` لهذا الغرض. على سبيل المثال, لإضافة أيقونة نجمة باستخدام Font Awesome, يمكنك استخدام الكود التالي:

Example

```
<!DOCTYPE html>
<html>
<head>
<script src="https://kit.fontawesome.com/a076d05399.js" crossorigin="anonymous"></script>
</head>
<body>
<i class="fas fa-cloud"></i>
<i class="fas fa-heart"></i>
<i class="fas fa-car"></i>
<i class="fas fa-file"></i>
<i class="fas fa-bars"></i>
</body>
</html>
```

Bootstrap Icons

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
</head>
<body>
<i class="glyphicon glyphicon-cloud"></i>
<i class="glyphicon glyphicon-remove"></i>
```

```

<i class="glyphicon"           glyphicon-user"></i>
<i class="glyphicon"           glyphicon-envelope"></i>
<i class="glyphicon"           glyphicon-thumbs-up"></i>

</body>
</html>

```

Google Icons

```

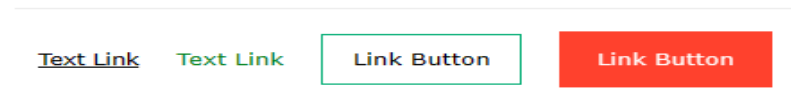
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">
</head>
<body>
<i class="material-icons">cloud</i>
<i class="material-icons">favorite</i>
<i class="material-icons">attachment</i>
<i class="material-icons">computer</i>
<i class="material-icons">traffic</i>

</body>
</html>

```

❖ CSS Links

With CSS, links can be styled in many different ways.



Styling Links

Links can be styled with any CSS property (e.g. **color**, **font-family**, **background**, etc.).

```

a {
  color: hotpink;
}

```

The four links states are:

- **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

```
/* unvisited link */
```

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

```
/* visited link */
```

```
a:visited {
  color: green;
}
```

```
/* mouse over link */
```

```
a:hover {
  color: hotpink;
}
```

```
/* selected link */
```

```
a:active {
  color: blue;
}
```

○ Link Buttons

Example

```
a:link, a:visited {
  background-color: #f44336;
  color: white;
  padding: 14px 25px;
  text-align: center;
  text-decoration: none;
}
```

```
display: inline-block;
}
```

```
a:hover, a:active {
  background-color: red;
}
```

○ More Examples

Example

This example demonstrates how to add other styles to hyperlinks:

```
a.one:link {color: #ff0000;}
a.one:visited {color: #0000ff;}
a.one:hover {color: #ffcc00;}
```

```
a.two:link {color: #ff0000;}
a.two:visited {color: #0000ff;}
a.two:hover {font-size: 150%;}
```

```
a.three:link {color: #ff0000;}
a.three:visited {color: #0000ff;}
a.three:hover {background: #66ff66;}
```

```
a.four:link {color: #ff0000;}
a.four:visited {color: #0000ff;}
a.four:hover {font-family: monospace;}
```

```
a.five:link {color: #ff0000; text-
decoration: none;}
a.five:visited {color: #0000ff; text-
decoration: none;}
a.five:hover {text-
decoration: underline;}
```


❖ CSS Lists

Unordered Lists:

- Coffee
- Tea
- Coca Cola

- Coffee
- Tea
- Coca Cola

Ordered Lists:

1. Coffee
2. Tea
3. Coca Cola

- I. Coffee
- II. Tea
- III. Coca

list-style-type property specifies the type of list item marker.

Example

```
ul.a {  
    list-style-type: circle;  
}  
  
ul.b {  
    list-style-type: square;  
}  
  
ol.c {  
    list-style-type: upper-
```

```
roman;  
}
```

```
ol.d {  
    list-style-type: lower-alpha;  
}
```

○ An Image as The List Item Marker

```
ul {  
    list-style-image: url('sqpurple.gif');  
}
```

○ Position The List Item Markers

```
ul.a {  
    list-style-position: outside;  
}
```

```
ul.b {  
    list-style-position: inside;  
}
```

Remove Default Settings

The **list-style-type:none** property can also be used to remove the markers/bullets. Note that the list also has default margin and padding. To remove this, add **margin:0** and **padding:0** to or :

Example

```
ul {  
    list-style-type: none;  
    margin: 0;  
    padding: 0;  
}
```

❖ CSS Tables

- Table Borders

Firstname	Lastname
Peter	Griffin
Lois	Griffin

```
table, th, td {  
    border: 1px solid;  
}
```

Full-Width Table

The table above might seem small in some cases. If you need a table that should span the entire screen (full-width), add **width: 100%** to the `<table>` element:

Example

```
table {  
    width: 100%;  
}
```

Firstname	Lastname
Peter	Griffin
Lois	Griffin

- Collapse Table Borders

The **border-collapse** property sets whether the table borders should be collapsed into a single border:

Example

```
table {  
    border-collapse: collapse;  
}
```

Firstname	Lastname
Peter	Griffin
Lois	Griffin

- CSS Table Size

Table Width and Height: The width and height of a table are defined by the **width** and **height** properties.

The example below sets the width of the table to 100%, and the height of the <th> elements to 70px:

Firstname	Lastname	Savings
Peter	Griffin	\$100
Lois	Griffin	\$150
Joe	Swanson	\$300

Example

```
table {  
  width: 100%;  
}
```

```
th {  
  height: 70px;  
}
```

CSS Table Alignment

▪ Horizontal Alignment

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

Firstname	Lastname	Savings
Peter	Griffin	\$100
Lois	Griffin	\$150
Joe	Swanson	\$300

▪ Vertical Alignment

```
td {  
  height: 50px;  
  vertical-align: bottom;  
}
```

Firstname	Lastname	Savings
Peter	Griffin	\$100
Lois	Griffin	\$150
Joe	Swanson	\$300

▪ CSS Table Style

Table Padding

```
th, td {  
  padding: 15px;  
  text-align: left;  
}
```

First Name	Last Name	Savings
Peter	Griffin	\$100
Lois	Griffin	\$150
Joe	Swanson	\$300

Add a container element (like <div>) with **overflow-x:auto** around the <table> element to make it responsive:

Example

```
<div style="overflow-x:auto;">
<table>
... table content ...
</table>

</div>
```

❖ CSS Layout – The display Property

خاصية العرض (display) هي واحدة من أهم خواص CSS التي تتحكم في تخطيط عناصر الصفحة.

تستخدم خاصية العرض لتغيير سلوك العرض الافتراضي لعناصر HTML.

The display Property

The **display** property is used to specify how an element is shown on a web page.

تحدد خاصية العرض (display) كيفية عرض عنصر HTML على صفحة الويب. لكل عنصر HTML قيمة عرض افتراضية تعتمد على نوع العنصر. على سبيل المثال، تكون قيمة العرض الافتراضية لمعظم العناصر إما "block" (مستوى كتلة) أو "inline" (مستوى خط).

The **display** property is used to change the default display behavior of HTML elements.

▪ Block-level Elements

A block-level element ALWAYS starts on a new line and takes up the full width available (stretches out to the left and right as far as it can).

The <div> element is a block-level element.

Examples of block-level elements:

- <div>
- <h1> - <h6>
- <p>
- <form>
- <header>
- <footer>
- <section>

▪ Inline Elements

An inline element DOES NOT start on a new line and only takes up as much width as necessary.

This is an inline element inside a paragraph.

Examples of inline elements:

- ``
- `<a>`
- ``

▪ CSS Layout - width and max-width

setting width, max-width and margin: auto;

As mentioned in the previous chapter; a block-level element always takes up the full width available (stretches out to the left and right as far as it can).

Setting the **width** of a block-level element will prevent it from stretching out to the edges of its container. Then, you can set the margins to auto, to horizontally center the element within its container. The element will take up the specified width, and the remaining space will be split equally between the two margins:

This `<div>` element has a width of 500px, and margin set to auto.

Note: The problem with the `<div>` above occurs when the browser window is smaller than the width of the element. The browser then adds a horizontal scrollbar to the page.

Using **max-width** instead, in this situation, will improve the browser's handling of small windows. This is important when making a site usable on small devices:

This `<div>` element has a max-width of 500px, and margin set to auto.

Tip: Resize the browser window to less than 500px wide, to see the difference between the two divs!

Here is an example of the two divs above:

Example

```
div.ex1 {  
  width: 500px;  
  margin: auto;  
  border: 3px solid #73AD21;  
}
```

```
div.ex2 {  
  max-width: 500px;  
  margin: auto;  
  border: 3px solid #73AD21;  
}
```

CSS Max-width

This div element has width: 500px;

This div element has max-width: 500px;

Tip: Drag the browser window to smaller than 500px wide, to see the difference between the two divs!

▪ CSS Layout - The position Property

The **position** property specifies the type of positioning method used for an element (static, relative, fixed, absolute or sticky).

The **position** property specifies the type of positioning method used for an element.

There are five different position values:

- static
- relative
- fixed
- absolute
- sticky

Elements are then positioned using the top, bottom, left, and right properties. However, these properties will not work unless the **position** property is set first. They also work differently depending on the position value.

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

- **position: sticky;**

An element with **position: sticky;** is positioned based on the user's scroll position.

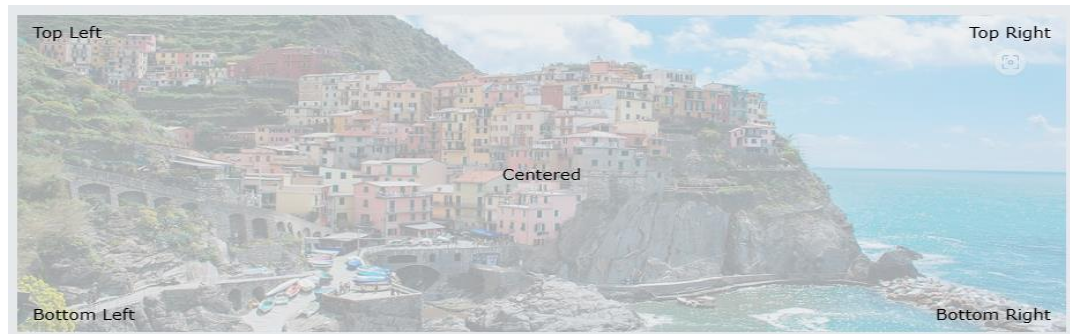
A sticky element toggles between **relative** and **fixed**, depending on the scroll position. It is positioned relative until a given offset position is met in the viewport - then it "sticks" in place (like position:fixed).

In this example, the sticky element sticks to the top of the page (**top: 0**), when you reach its scroll position.

Example

```
div.sticky {  
  position: -webkit-sticky; /* Safari */  
  position: sticky;  
  top: 0;  
  background-color: green;  
  border: 2px solid #4CAF50;}
```

❖ Positioning Text In an Image



```
<style>
}
.container {
  position: relative;
}
.topleft {
  position: absolute;
  top: 8px;
  left: 16px;
  font-size: 18px;
}
img {
  width: 100%;
  height: auto;
  opacity: 0.3;
}
</style>
</head>
<body>
<h2>Image Text</h2>
<p>Add some text to an image in the top
left corner:</p>
<div class="container">
  
  <div class="topleft">Top Left</div>
</div>
</body>
</html>
```

❖ CSS Layout - float and clear

The float Property

The **float** property is used for positioning and formatting content e.g. let an image float left to the text in a container.

The **float** property can have one of the following values:

- **left** - The element floats to the left of its container
- **right** - The element floats to the right of its container
- **none** - The element does not float (will be displayed just where it occurs in the text). This is default

- **inherit** - The element inherits the float value of its parent

Example - float: right;

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisl, sed ullamcorper ipsum dignissim ac...



```
img {  
  float: right;  
}
```

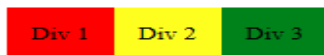
Example - Float Next To Each Other

Example

```
div {  
  float: left;  
  padding: 15px;  
}  
  
.div1 {  
  background: red;  
}  
  
.div2 {  
  background: yellow;  
}  
  
.div3 {  
  background: green;  
}
```

Float Next To Each Other

In this example, the three divs will float next to each other.



CSS Layout - clear and clearfix

The clear Property

When we use the **float** property, and we want the next element below (not on right or left), we will have to use the **clear** property.

The **clear** property specifies what should happen with the element that is next to a floating element.

The **clear** property can have one of the following values:

- **none** - The element is not pushed below left or right floated elements. This is default
- **left** - The element is pushed below left floated elements
- **right** - The element is pushed below right floated elements
- **both** - The element is pushed below both left and right floated elements
- **inherit** - The element inherits the clear value from its parent

```
div1 {
  float: left;
}
```

```
div2 {
  clear: left;
}
```

Without clear

div1 div2 - Notice that div2 is after div1 in the HTML code. However, since div1 floats to the left, the text in div2 flows around div1.

With clear

div3
div4 - Here, clear: left; moves div4 down below the floating div3. The value "left" clears elements floated to the left. You can also clear "right" and "both".

The clearfix Hack

```
.clearfix {
  overflow: auto;
}
```

```
.clearfix::after {
  content: "";
  clear: both;
  display: table;
}
```

Without Clearfix

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum...

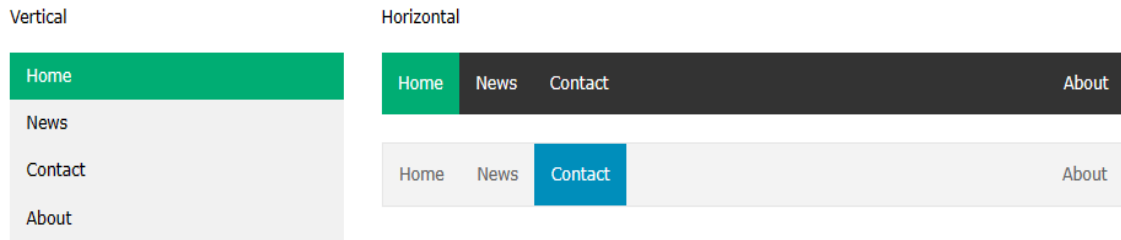


With Clearfix

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum...



❖ CSS Navigation Bar



Having easy-to-use navigation is important for any web site.

With CSS you can transform boring HTML menus into good-looking navigation bars.

تعد قوائم التنقل (navigation bars) سهلة الاستخدام عنصرًا مهمًا لأي موقع ويب. فهي تساعد الزوار على الوصول إلى أقسام الموقع المختلفة بسرعة وسهولة.

يمكن أن تكون قوائم التنقل التي يتم إنشاؤها باستخدام HTML فقط ذات مظهر ممل وغير جذابة.

لكن باستخدام لغة تصميم وتنسيق المواقع المت (CSS) cascading style sheets يمكنك تحويل قوائم التنقل البسيطة إلى قوائم تنقل جذابة وواضحة.

▪ Navigation Bar = List of Links

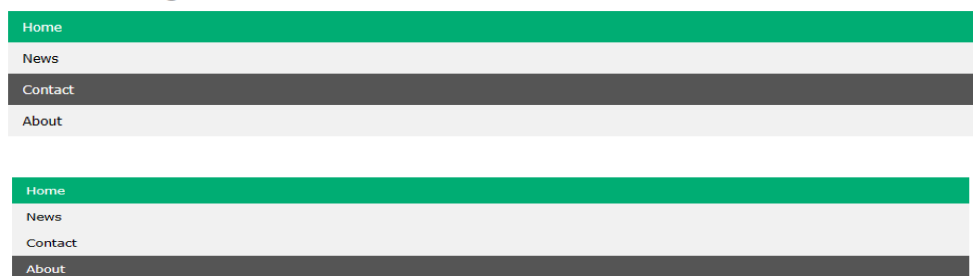
```
<ul>
  <li><a href="default.asp">Home</a></li>
  <li><a href="news.asp">News</a></li>
  <li><a href="contact.asp">Contact</a></li>
  <li><a href="about.asp">About</a></li>
</ul>
```

remove the bullets and the margins and padding from the list:

```
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
}
```

▪ CSS Vertical Navigation Bar

Vertical Navigation Bar

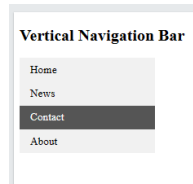


To build a vertical navigation bar, you can style the `<a>` elements inside the list, in addition to the code from the previous page:

```
li a {
  display: block;
  width: 60px;
}
```

```
<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>
```

Example:

```
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;
}
```

```
<h2>Vertical Navigation Bar</h2>
```

```
<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>
```

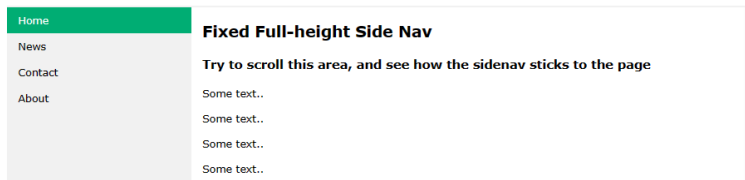
```
</ul>
```

Example :

```
ul {
  border: 1px solid #555;
}
```

```
li {
  text-align: center;
  border-bottom: 1px solid #555;
}
```

```
li:last-child {
  border-bottom: none;
}
```



```
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  width: 25%;
  background-color: #f1f1f1;
  height: 100%; /* Full height */
  position: fixed; /* Make it stick, even on scroll */
  overflow: auto; /* Enable scrolling if the sidenav has too much
content */
}
```

CSS Horizontal Navigation Bar



- Inline List Items

Example

```
li {
  display: inline;
}
```

list-style-type: none;

margin: 0;

padding: 0;

overflow: hidden;

}

li {

float: left;

}

li a {

display: block;

padding: 8px;

background-color: #dddddd;

}

[Home](#) [News](#) [Contact](#) [About](#)

Note: If a !DOCTYPE is not specified, floating items can produce unexpected results.

A background color is added to the links to show the link area. The whole link area is clickable, not just the text.

Note: overflow:hidden is added to the ul element to prevent li elements from going outside of the list.

```
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #dddddd;
}
```

```
li {
```

[Home](#) [News](#) [Contact](#) [About](#)

A background color is added to the list instead of each link to create a full-width background color.

```
float: left;
```

```
}
```

```
li a {
```

```
display: block;
```

```
padding: 8px;
```

```
}
```

Horizontal Navigation Bar Examples



```
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333;
```

```
li {
```

```
float: left;
```

```
}
```

```
li a {
```

```
display: block;
```

```
color: white;
```

```
text-align: center;
```

```
padding: 14px 16px;
```

```
text-decoration: none;
```

```
}
```

```
li a:hover:not(.active) {
```

```
background-color: #111;
```

```
}
```

```
.active {
```

```
background-color: #04AA6D;
```

```
}
```

Right-Align Links

```
<ul>
  <li><a href="#home">Home</a>
</li>
  <li><a href="#news">News</a>
</li>
  <li><a href="#contact">Conta
ct</a></li>
```

```
  <li style="float:right"><a c
lass="active" href="#about">Ab
out</a></li>
</ul>
```

```
ul {
```

```
list-style-type: none;
```

```

margin: 0;
padding: 0;
overflow: hidden;
background-color: #333;
}
li {
float: left;
}
li a {
display: block;
color: white;

```

```

text-align: center;
padding: 14px 16px;
text-decoration: none;
}

li a:hover:not(.active) {
background-color: #111;
}

.active {
background-color: #04AA6D;
}

```

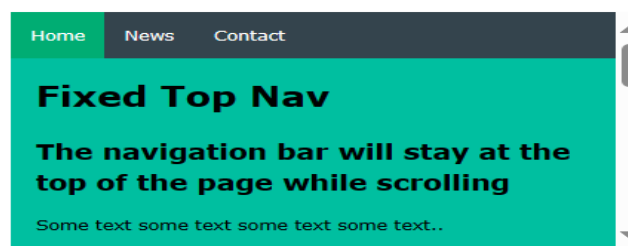
Fixed Navigation Bar

Fixed Top

```

ul {
position: fixed;
top: 0;
width: 100%;
}

```



Fixed Bottom

```

ul {
position: fixed;
bottom: 0;
width: 100%;
}

```

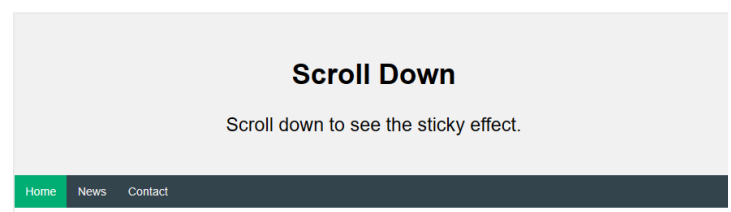


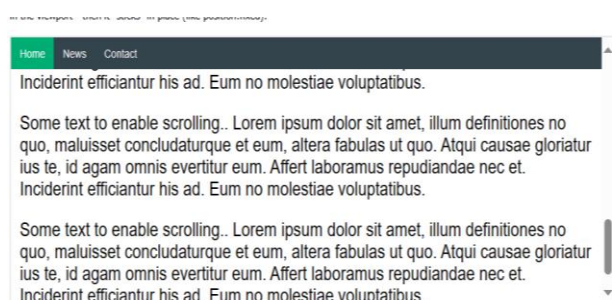
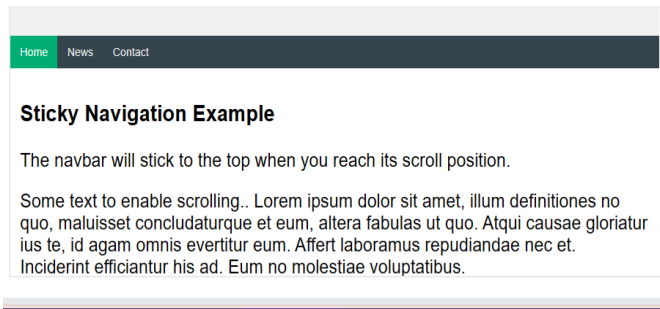
Sticky Navbar

```

ul {
position: -webkit-sticky; /*
Safari */
position: sticky;
top: 0;
}

```





```
ul {
  position: -webkit-sticky; /* Safari */
  position: sticky;
  top: 0;
}
```

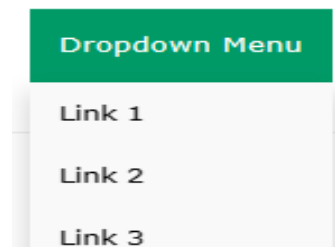
CSS Dropdowns

```
<style>
.dropdown {
  position: relative;
  display: inline-block;
}

.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f9f9f9;
  min-width: 160px;
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
  padding: 12px 16px;
  z-index: 1;
}

.dropdown:hover .dropdown-content {
  display: block;
}
</style>

<div class="dropdown">
  <span>Mouse over me</span>
  <div class="dropdown-content">
    <p>Hello World!</p>
  </div>
</div>
```



More Examples

```
.dropdown {  
  position: relative;  
  display: inline-block;  
}  
.dropdown-content {  
  display: none;  
  position: absolute;  
  background-color: #f9f9f9;  
  min-width: 160px;  
  box-shadow: 0px 8px 16px 0px  
  rgba(0,0,0,0.2);
```

Dropdown Navbar

```
ul {  
  list-style-type: none;  
  margin: 0;  
  padding: 0;  
  overflow: hidden;  
  background-color: #333;  
}  
li {  
  float: left;  
}  
li a, .dropbtn {  
  display: inline-block;  
  color: white;  
  text-align: center;  
  padding: 14px 16px;  
  text-decoration: none;  
}
```

```
z-  
1;  
}
```

```
.dropdown:hover .dropdown-content {  
  display: block;  
}  
.desc {  
  padding: 15px;  
  text-align: center;  
}
```

```
li a:hover, .dropdown:hover .dropbtn {  
  background-color: red;  
}
```

```
li.dropdown {  
  display: inline-block;  
}
```

```
.dropdown-content {  
  display: none;  
  position: absolute;  
  background-color: #f9f9f9;  
  min-width: 160px;  
  box-shadow: 0px 8px 16px 0px  
  rgba(0,0,0,0.2);  
  z-index: 1;  
}
```



index:

```

.dropdown-content a {
  color: black;
  padding: 12px 16px;
  text-decoration: none;
  display: block;
  text-align: left;
}

.dropdown-content a:hover {background-color: #f1f1f1;}

.dropdown:hover .dropdown-content {
  display: block;
}

</style>

</head>

<body>

<ul>

```

```

<li><a href="#home">Home</a></li>

<li><a href="#news">News</a></li>

<li class="dropdown">

  <a href="javascript:void(0)"
class="dropbtn">Dropdown</a>

  <div class="dropdown-content">

    <a href="#">Link 1</a>

    <a href="#">Link 2</a>

    <a href="#">Link 3</a>

  </div>

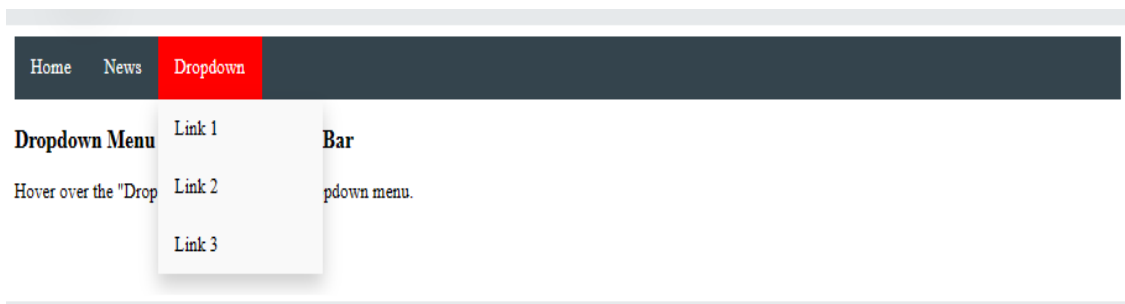
</li>

</ul>

<h3>Dropdown Menu inside a Navigation
Bar</h3>

<p>Hover over the "Dropdown" link to
see the dropdown menu.</p>

```



❖ CSS Forms

Styling Input Fields

Use the `width` property to determine the width of the input field:

Padded Inputs

Use the `padding` property to add space inside the text field.

```
input[type=text] {  
  width: 100%;  
  padding: 12px 20px;  
  margin: 8px 0;  
  box-sizing: border-box;  
}
```

Bordered Inputs

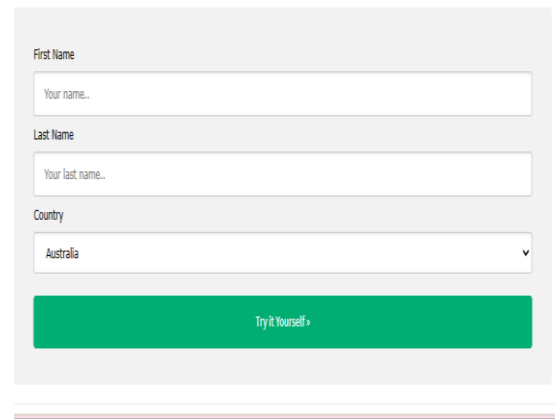
```
input[type=text] {  
  border: 2px solid red;  
  border-radius: 4px;  
}  
  
input[type=text] {  
  border: none;  
  border-bottom: 2px solid red;  
}
```

Colored Inputs

```
input[type=text] {  
  background-color: #3CBC8D;  
  color: white;  
}
```

Focused Inputs

```
input[type=text]:focus {  
  background-color: lightblue;  
}  
  
input[type=text]:focus {  
  border: 3px solid #555;  
}
```

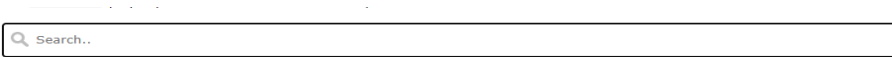


Input with icon/image



```
input[type=text] {  
  background-color: white;  
  background-image: url('searchicon.png');  
  background-position: 10px 10px;  
  background-repeat: no-repeat;  
  padding-left: 40px;  
}
```

Animated Search Input



```
input[type=text] {  
  transition: width 0.4s ease-in-out;  
}  
  
input[type=text]:focus {  
  width: 100%;  
}
```