html
<html>
<head id="html_head">
<title>Website name</title>
<noscript><meta http-equiv="X-Frame
<script type="text/javascript">
</script>
pageNow = new Date().getTime(
bar">
ackground"

HTML Basics

Teacher: Mr. Ahmad Osman

Date: 14/12/2018

Overview & Purpose

In this Document, the participant will learn the basics of HTML like Tags, Elements, structure, layouts, and other skills to be able to build a complete HTML page

Education Standards

the participant should have the basics about editing programs and how to save and show the HTML Page.
In some chapters, the participant should have the basics about CSS.

Objectives

1. Create HTML file
2. Tags
3. Elements and attributes
4. layouts

Software Needed

1. Visual Studio Code
2. Web Browser (Chrome Recommended)

Verification

1. the student should create an HTML page and be able to browse it.
2. Also, she/he should have the knowledge about most common HTML tags and techniques to build a complete HTML page

Activity

Learning using coding and exercises, discover the Html Elements and how to use it.

HTML Documents

All HTML documents must start with a document type declaration: <!DOCTYPE html>.

The HTML document itself begins with <html> and ends with </html>.

The visible part of the HTML document is between <body> and </body>.

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

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

HTML Elements

An HTML element usually consists of a start tag and end tag, with the content inserted in between:

```
<tagname>Content goes here...</tagname>
```

The HTML element is everything from the start tag to the end tag:

```
<p>My first paragraph.</p>
```

Nested HTML Elements

HTML elements can be nested (elements can contain elements).

All HTML documents consist of nested HTML elements.

This example contains four HTML elements:

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

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

Do Not Forget the End Tag

Some HTML elements will display correctly, even if you forget the end tag:

```
<html>
<body>

<p>This is a paragraph
<p>This is a paragraph

</body>
</html>
```

Empty HTML Elements

HTML elements with no content are called empty elements.

<br> is an empty element without a closing tag (the <br> tag defines a line break).

Empty elements can be "closed" in the opening tag like this: <br />.

HTML5 does not require empty elements to be closed. But if you want stricter validation, or if you need to make your document readable by XML parsers, you must close all HTML elements properly.

Use Lowercase Tags

HTML tags are not case sensitive: <P> means the same as <p>.

The HTML5 standard does not require lowercase tags, but it is recommended to use lowercase in HTML and demands lowercase for stricter document types like XHTML.

HTML Attributes

- All HTML elements can have attributes
- Attributes provide additional information about an element
- Attributes are always specified in the start tag
- Attributes usually come in name/value pairs like: name="value"

Example

```
<a
href="https://www.digitalcareerinstitute.org">
This is a link</a>
```

HTML links are defined with the <a> tag. The link address is specified in the href attribute

HTML Headings

Headings are defined with the <h1> to <h6> tags.

<h1> defines the most important heading. <h6> defines the least important heading.

```
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
```

HTML Paragraphs

The HTML <p> element defines a paragraph

```
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
```

HTML Style Attribute

Setting the style of an HTML element, can be done with the style attribute.

The HTML style attribute has the following syntax

```
<tagname style="property:value;">
```

Example

The background-color property defines the background color for an HTML element

```
<body style="background-color:powderblue;">

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

</body>
```

HTML Text Formatting

- <b> - Bold text
- <strong> - Important text
- <i> - Italic text
- <em> - Emphasized text
- <mark> - Marked text
- <small> - Small text
- <del> - Deleted text
- <ins> - Inserted text
- <sub> - Subscript text
- <sup> - Superscript text

HTML Quotation and Citation Elements

HTML <q>

The HTML <q> element defines a short quotation.

Browsers usually insert quotation marks around the <q> element.

```
<p>DCI Aims: <q>Build the new Web Developers </q></p>
```

HTML <blockquote> for Quotations

The HTML <blockquote> element defines a section that is quoted from another source.

Browsers usually indent <blockquote> elements.

```html
<p>Here is a quote from DCI:</p>
<blockquote cite="https://digitalcareerinstitute.org">
The Digital Career Institute was born as an initiative to integrate
refugees into digital jobs.</blockquote>
```

HTML <abbr> for Abbreviations

The HTML <abbr> element defines an abbreviation or an acronym.

Marking abbreviations can give useful information to browsers, translation systems and search-engines.

```html
<p>The <abbr title="Digital Career Institute">DCI</abbr> was born as an
initiative to integrate refugees into digital jobs.</p>
```

HTML <address>

The HTML <address> element defines contact information (author/owner) of a document or an article.

The <address> element is usually displayed in italic. Most browsers will add a line break before and after the element.

```html
<address>
Written by Ahmad Osman.<br>
Visit us at:<br>
digitalcareerinstitute.org<br>
Lange Reihe 14 , Hamburg<br>
Germany
</address>
```

HTML <cite>

The HTML <cite> element defines the title of a work.

Browsers usually display <cite> elements in italic.

```html
<p><cite>Html Basics</cite> by Ahmad Osman. Written in 15.12.2018.</p>
```

HTML Comment Tags

You can add comments to your HTML source by using the following syntax:

```
<!-- Write your comments here -->
```

Note: Comments are not displayed by the browser, but they can help document your HTML source code.

Example

```
<!-- This is a comment -->

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

<!-- Remember to add more information here -->
```

Comments are also great for debugging HTML, because you can comment out HTML lines of code, one at a time, to search for errors:

```
<!-- Do not display this at the moment
<img border="0" src="pic_trulli.jpg" alt="Trulli">
-->
```

HTML Links

In HTML, links are defined with the **<a>** tag

```
<a href="url">link text</a>
```

Example

```
<a href="https://digitalcareerinstitute.org">Visit our website</a>
```

Target Attribute

The target attribute specifies where to open the linked document.

- _blank - Opens the linked document in a new window or tab
- _self - Opens the linked document in the same window/tab as it was clicked (this is default)
- _parent - Opens the linked document in the parent frame
- _top - Opens the linked document in the full body of the window
- framename - Opens the linked document in a named frame

This example will open the linked document in a new browser window/tab:

```
<a href="https://digitalcareerinstitute.org" target="_blank">DCI</a>
```

HTML Links - Create a Bookmark

HTML bookmarks are used to allow readers to jump to specific parts of a Web page.

Bookmarks can be useful if your webpage is very long.

To make a bookmark, you must first create the bookmark, and then add a link to it.

When the link is clicked, the page will scroll to the location with the bookmark.
Example
a link to the bookmark ("Jump to Chapter 4"), from within the same page

```
<h2 id="C4">Chapter 4</h2>
```

link to the bookmark ("Jump to Chapter 4"), from another page

```
<a href="html_demo.html#C4">Jump to Chapter 4</a>
```

HTML Images

```
<img src="Image URL" alt="Text will be show if there is a problem with the
Image">
```

The width and height attributes always defines the width and height of the image in pixels.

```
<img src="img.jpg" alt="image" width="500" height="600">
```

HTML Table

An HTML table is defined with the <table> tag.

Each table row is defined with the <tr> tag. A table header is defined with the <th> tag. By default, table headings are bold and centered. A table data/cell is defined with the <td> tag.
Example

```
<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Ahmad</td>
    <td>Osman</td>
    <td>32</td>
  </tr>
  <tr>
    <td>Basel</td>
    <td>Zaghmot</td>
    <td>28</td>
  </tr>
</table>
```

Cells that Span Many Columns

To make a cell span more than one column, use the colspan attribute:

```
<table style="width:100%">
  <tr>
    <th>Name</th>
    <th colspan="2">Telephone</th>
  </tr>
  <tr>
    <td>Ahmad Osman</td>
    <td>0123154</td>
    <td>0215478</td>
  </tr>
</table>
```

Cells that Span Many Rows

To make a cell span more than one row, use the rowspan attribute:

```
<table style="width:100%">
  <tr>
    <th>Name:</th>
    <td>Ahmad Osman</td>
  </tr>
  <tr>
    <th rowspan="2">Telephone:</th>
    <td>02145</td>
  </tr>
  <tr>
    <td>0124587</td>
  </tr>
</table>
```

Adding a Caption

To add a caption to a table, use the <caption> tag:

```
<table style="width:100%">
  <caption>Monthly savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>February</td>
    <td>$50</td>
  </tr>
</table>
```

HTML Lists

Unordered HTML List

An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.

The list items will be marked with bullets (small black circles) by default:

```
<ul>
  <li>Item1</li>
  <li>Item2</li>
  <li>Item3</li>
</ul>
```

The CSS list-style-type property is used to define the style of the list item marke

Value	Description
disc	Sets the list item marker to a bullet (default)
circle	Sets the list item marker to a circle
square	Sets the list item marker to a square
none	The list items will not be marked

Example

```
<ul style="list-style-type:circle">
  <li>Item1</li>
  <li>Item2</li>
  <li>Item3</li>
</ul>
```

Ordered HTML List

An ordered list starts with the <ol> tag. Each list item starts with the <li> tag.

The list items will be marked with numbers by default:

Example

```
<ol>
  <li>Item1</li>
  <li>Item2</li>
  <li>Item3</li>
</ol>
```

Type Attribute

The type attribute of the <ol> tag, defines the type of the list item marker:

Type	**Description**
type="1"	The list items will be numbered with numbers (default)
type="A"	The list items will be numbered with uppercase letters
type="a"	The list items will be numbered with lowercase letters
type="I"	The list items will be numbered with uppercase roman numbers
type="i"	The list items will be numbered with lowercase roman numbers

Example

```
<ol type="1">
  <li>Item1</li>
  <li>Item2</li>
  <li>Item3</li>
</ol>
```

HTML Description Lists

HTML also supports description lists.

A description list is a list of terms, with a description of each term.

The <dl> tag defines the description list, the <dt> tag defines the term (name), and the <dd> tag describes each term:

Example

```
<dl>
  <dt>HTML</dt>
  <dd>- Hybertext markup language</dd>
  <dt>Javascript</dt>
  <dd>- Programming language</dd>
</dl>
```

Nested HTML Lists

List can be nested (lists inside lists):

Example

```
<ul>
  <li>Item1</li>
  <li>Item2
    <ul>
      <li>subitem1</li>
      <li>subitem2</li>
    </ul>
  </li>
  <li>Item3</li>
</ul>
```

Control List Counting

By default, an ordered list will start counting from 1. If you want to start counting from a specified number, you can use the start attribute:

```
<ol start="50">
  <li>Item1</li>
  <li>Item2</li>
  <li>Item3</li>
</ol>
```

HTML 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).

Example

```
<div>Hello</div>
<div>World</div>
```

Block level elements in HTML:

```
<address> <article> <aside> <blockquote> <canvas> <dd> <div> <dl> <dt>
<fieldset> <figcaption> <figure> <footer> <form> <h1>-<h6> <header> <hr>
<li> <main> <nav> <noscript> <ol> <output> <p> <pre> <section> <table>
<tfoot> <ul> <video>
```

HTML Inline Elements

Example

```
<span>Hello</span>
<span>World</span>
```

Inline elements in HTML:

```
<a> <abbr> <acronym> <b> <bdo> <big> <br> <button> <cite> <code> <dfn> <em>
<i> <img> <input> <kbd> <label> <map> <object> <q> <samp> <script> <select>
<small> <span> <strong> <sub> <sup> <textarea> <time> <tt> <var>
```

HTML Iframes

An iframe is used to display a web page within a web page.

```
<iframe src="URL"></iframe>
```

Use the height and width attributes to specify the size of the iframe.

The attribute values are specified in pixels by default

```
<iframe src="URL" height="200" width="300"></iframe>
```

Iframe - Target for a Link

An iframe can be used as the target frame for a link.

The target attribute of the link must refer to the name attribute of the iframe:

```
<iframe src="https://www.google.com" name="iframe_a"></iframe>

<p><a href="https://digitalcareerinstitute.org"
target="iframe_a">DCI</a></p>
```

HTML File Paths

Path	Description
<img src="picture.jpg">	picture.jpg is located in the same folder as the current page
<img src="images/picture.jpg">	picture.jpg is located in the images folder in the current folder
<img src="/images/picture.jpg">	picture.jpg is located in the images folder at the root of the current web
<img src="../picture.jpg">	picture.jpg is located in the folder one level up from the current folder

HTML Head

The <head> element is a container for metadata (data about data) and is placed between the <html> tag and the <body> tag.

HTML metadata is data about the HTML document. Metadata is not displayed.

Metadata typically define the document title, character set, styles, links, scripts, and other meta information.

The following tags describe metadata: <title>, <style>, <meta>, <link>, <script>, and <base>.

The HTML <title> Element

The <title> element defines the title of the document, and is required in all HTML/XHTML documents.

The <title> element:

- defines a title in the browser tab
- provides a title for the page when it is added to favorites
- displays a title for the page in search engine results

Example

```
<!DOCTYPE html>
<html>

<head>
  <title>Page Title</title>
</head>

<body>
The content of the document......
</body>

</html>
```

The HTML <style> Element

The <style> element is used to define style information for a single HTML page:

```
<style>
  body {background-color: powderblue;}
  h1 {color: red;}
  p {color: blue;}
</style>
```

The HTML <link> Element

The <link> element is used to link to external style sheets:

```
<link rel="stylesheet" href="mystyle.css">
```

The HTML <meta> Element

The <meta> element is used to specify which character set is used, page description, keywords, author, and other metadata.

Metadata is used by browsers (how to display content), by search engines (keywords), and other web services.

Define the character set used:

```
<meta charset="UTF-8">
```

Define a description of your web page:

```
<meta name="description" content="HTML tutorials">
```

Define keywords for search engines:

```
<meta name="keywords" content="HTML, CSS, XML, JavaScript">
```

Define the author of a page:

```
<meta name="author" content="Ahmad Osman">
```

Refresh document every 15 seconds:

```
<meta http-equiv="refresh" content="15">
```

Viewport

HTML5 introduced a method to let web designers take control over the viewport, through the <meta> tag.

The viewport is the user's visible area of a web page. It varies with the device, and will be smaller on a mobile phone than on a computer screen.

You should include the following <meta> viewport element in all your web pages:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

A <meta> viewport element gives the browser instructions on how to control the page's dimensions and scaling.

The width=device-width part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).

The initial-scale=1.0 part sets the initial zoom level when the page is first loaded by the browser.

The HTML <script> Element

The <script> element is used to define client-side JavaScripts.

This JavaScript writes "Hello JavaScript!" into an HTML element with id="demo":

```
<script>
function myFunction {
  document.getElementById("demo").innerHTML = "Hello JavaScript!";
}
</script>
```

The HTML <base> Element

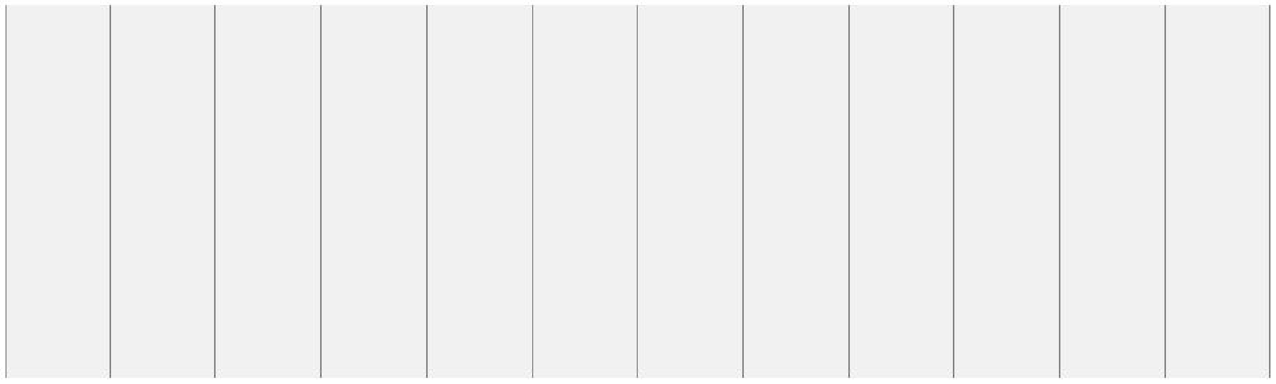
The <base> element specifies the base URL and base target for all relative URLs in a page:

```
<base href="https://digitalcareerinstitute.org/" target="_blank">
```

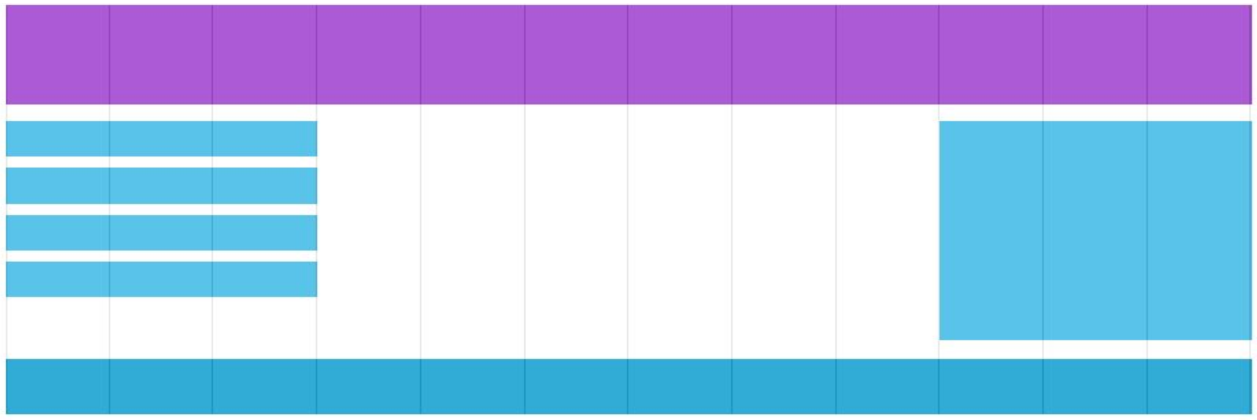
HTML Layout Techniques

Float

Many web pages are based on a grid-view, which means that the page is divided into columns:



Using a grid-view is very helpful when designing web pages. It makes it easier to place elements on the page.



A responsive grid-view often has 12 columns, and has a total width of 100%, and will shrink and expand as you resize the browser window.

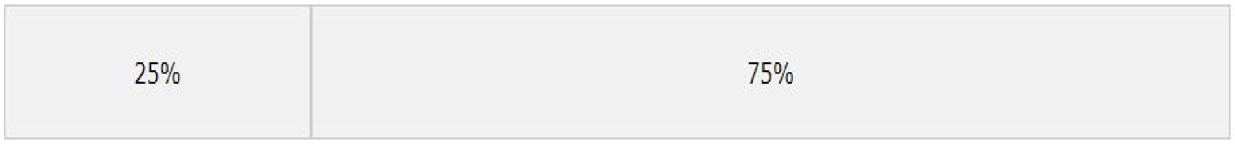
Building a Responsive Grid-View

First ensure that all HTML elements have the box-sizing property set to border-box. This makes sure that the padding and border are included in the total width and height of the elements.

Add the following code in your CSS:

```
* {
  box-sizing: border-box;
}
```

The following example shows a simple responsive web page, with two columns:



```
.menu {
  width: 25%;
  float: left;
}
.main {
  width: 75%;
  float: left;
}
```

we want to use a responsive grid-view with 12 columns, to have more control over the web page.

First we must calculate the percentage for one column: 100% / 12 columns = 8.33%.

Then we make one class for each of the 12 columns, class="col-" and a number defining how many columns the section should span:

CSS

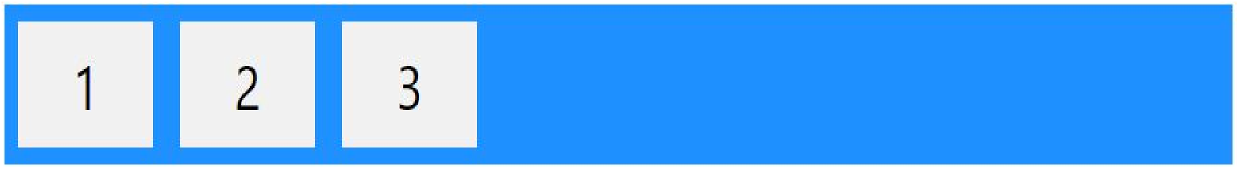
```
.col-1 {width: 8.33%;}
.col-2 {width: 16.66%;}
.col-3 {width: 25%;}
.col-4 {width: 33.33%;}
.col-5 {width: 41.66%;}
.col-6 {width: 50%;}
.col-7 {width: 58.33%;}
.col-8 {width: 66.66%;}
.col-9 {width: 75%;}
.col-10 {width: 83.33%;}
.col-11 {width: 91.66%;}
.col-12 {width: 100%;}

/*All these columns should be floating to the left, and have a padding of 15px*/

[class*="col-"] {
  float: left;
  padding: 15px;
}
```

HTML

```
<div class="row">
```

```
  <div class="col-3">...</div> <!-- 25% -->
  <div class="col-9">...</div> <!-- 75% -->
</div>
```

Flexbox Layout



HTML

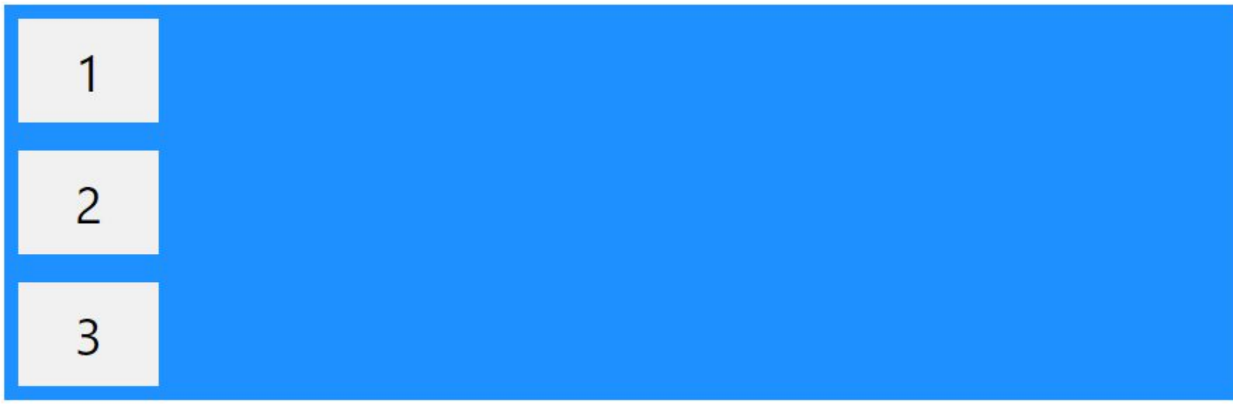
```
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>
```

CSS

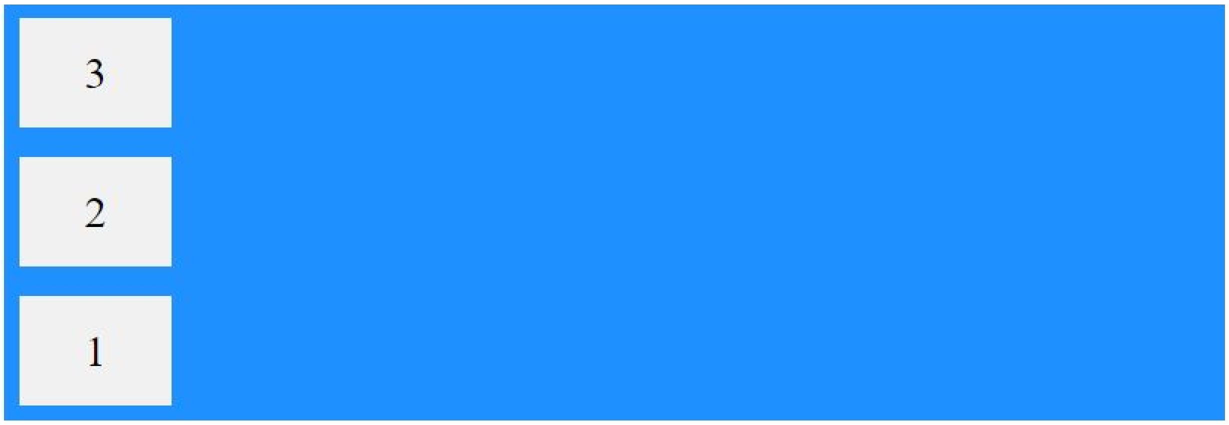
```
.flex-container {
  display: flex;
  background-color: DodgerBlue;
}

.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  padding: 20px;
  font-size: 30px;
}
```

flex-direction Property

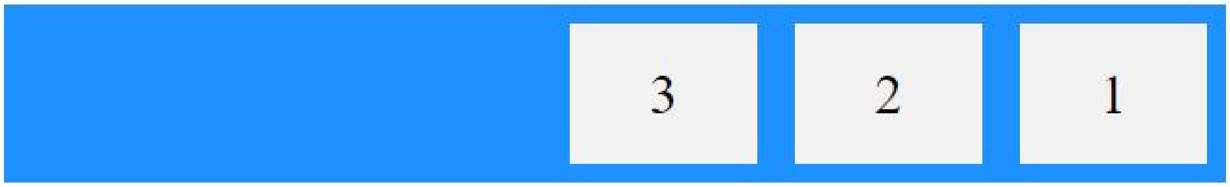
The flex-direction property defines in which direction the container wants to stack the flex items.



```
.flex-container {
  display: flex;
  flex-direction: column;
}
```



```
.flex-container {
  display: flex;
  flex-direction: column-reverse;
}
```

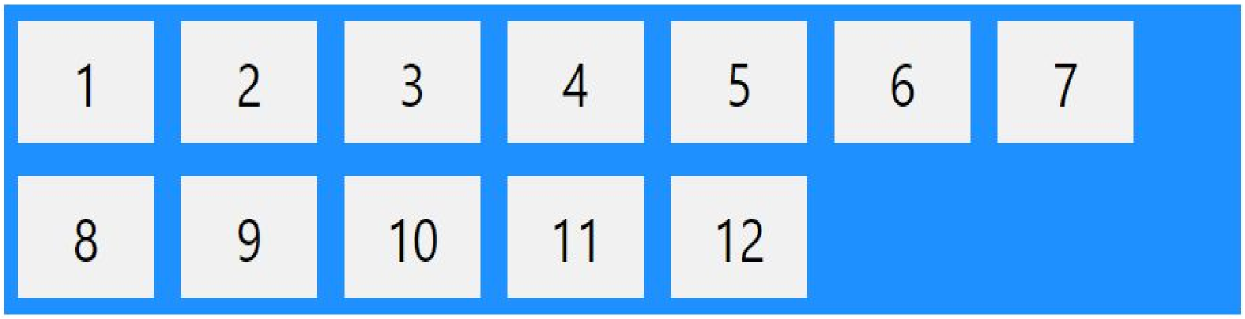


```
.flex-container {
  display: flex;
  flex-direction: row-reverse;
}
```

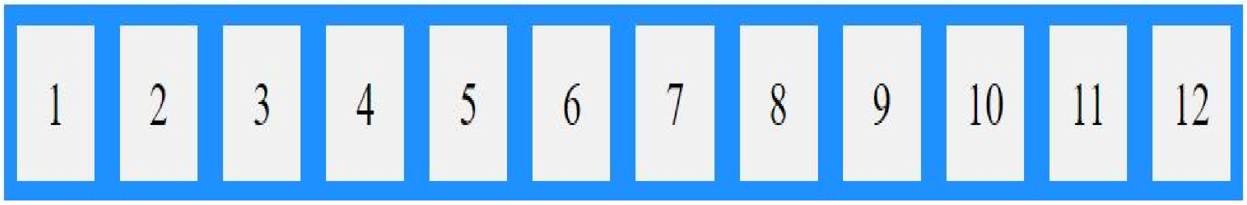
flex-wrap Property

The flex-wrap property specifies whether the flex items should wrap or not.

The examples below have 12 flex items, to better demonstrate the flex-wrap property.



```
.flex-container {
  display: flex;
  flex-wrap: wrap;
}
```



```
.flex-container {
  display: flex;
  flex-wrap: nowrap;
}
```

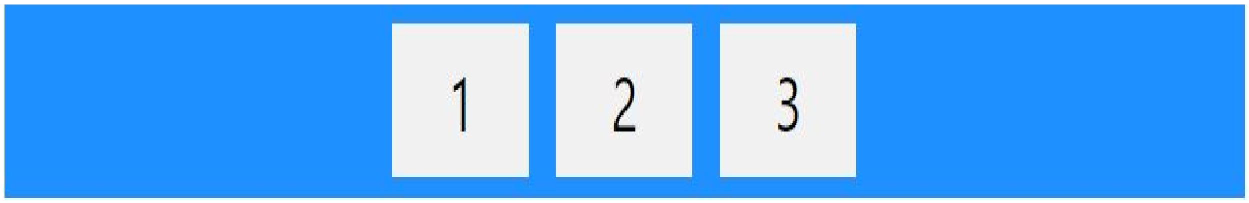
flex-flow Property

The flex-flow property is a shorthand property for setting both the flex-direction and flex-wrap properties.

```
.flex-container {
  display: flex;
  flex-flow: row wrap;
}
```

justify-content Property

The justify-content property is used to align the flex items:



```
.flex-container {
  display: flex;
  justify-content: center;
}
```

It can have the following values:
Flex-start:
The flex-start value aligns the flex items at the beginning of the container (this is default)

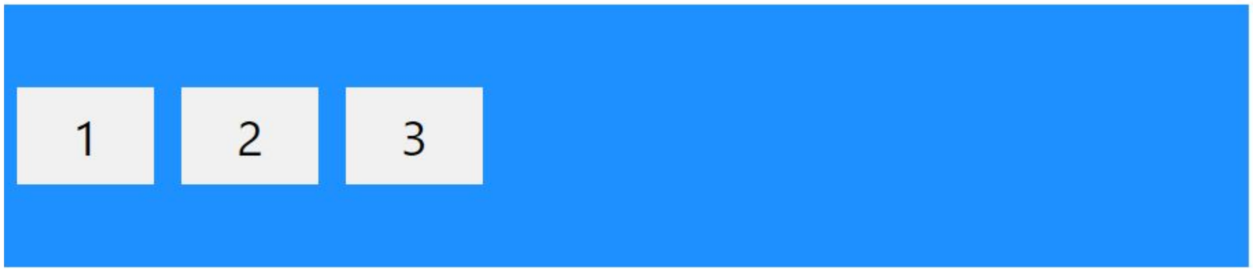
Flex-end:
The flex-end value aligns the flex items at the end of the container

Space-around :
The space-around value displays the flex items with space before, between, and after the lines

Space-between:
The space-between value displays the flex items with space between the lines

align-items Property

The align-items property is used to align the flex items vertically.



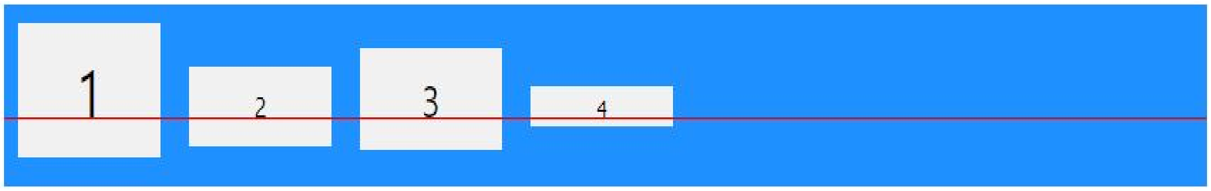
```
.flex-container {
  display: flex;
  height: 200px;
  align-items: center;
}
```

It can have the following values:
Flex-start:
The flex-start value aligns the flex items at the top of the container

Flex-end:
The flex-end value aligns the flex items at the bottom of the container

Stretch:
The stretch value stretches the flex items to fill the container (this is default)

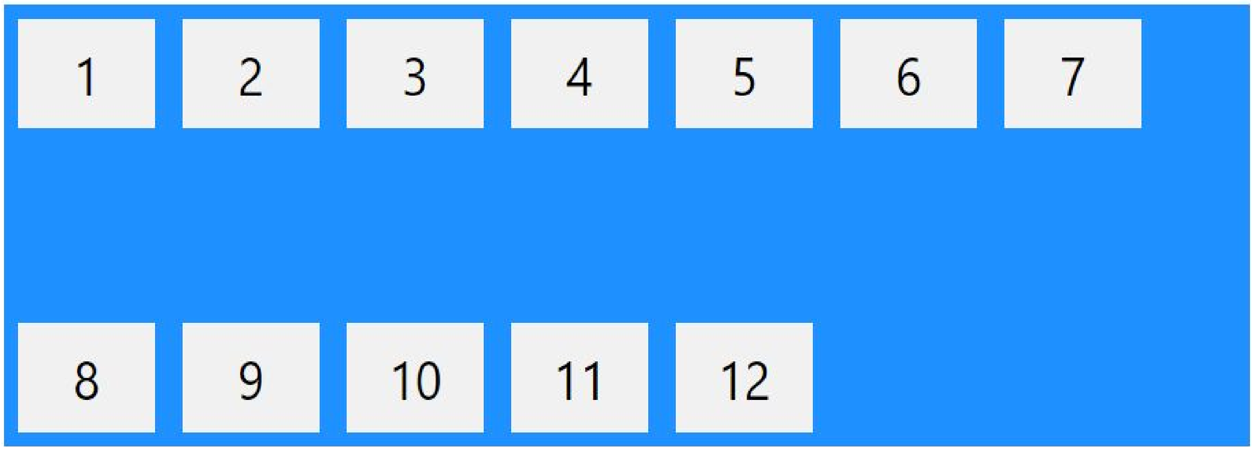
Baseline:
The baseline value aligns the flex items such as their baselines aligns.
the example uses different font-size to demonstrate that the items gets aligned by the text baseline



```
.flex-container {
  display: flex;
  height: 200px;
  align-items: baseline;

}
```

align-content Property

The align-content property is used to align the flex lines.



```css
.flex-container {
  display: flex;
  height: 600px;
  flex-wrap: wrap;
  align-content: space-between;
}
```

It can have the following values:
Space-around:
The space-around value displays the flex lines with space before, between, and after them

Stretch:
The stretch value stretches the flex lines to take up the remaining space (this is default)

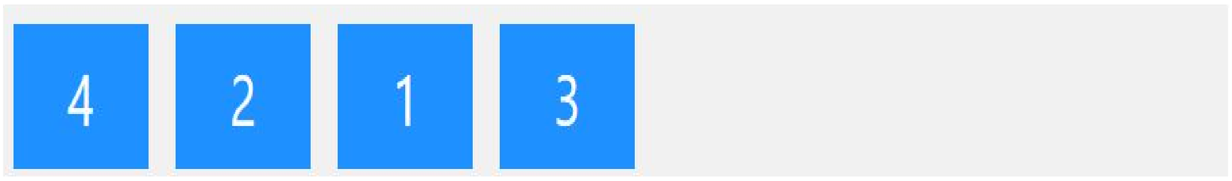
Center:
The center value displays display the flex lines in the middle of the container:

Flex-start:
The flex-start value displays the flex lines at the start of the container

Flex-end:
The flex-end value displays the flex lines at the end of the container

order Property

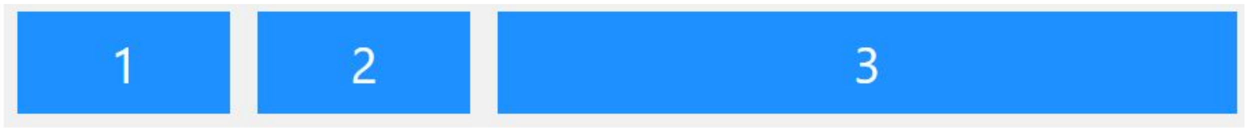
The order property specifies the order of the flex items.



```
<div class="flex-container">
  <div style="order: 3">1</div>
  <div style="order: 2">2</div>
  <div style="order: 4">3</div>
  <div style="order: 1">4</div>
</div>
```

flex-grow Property

The flex-grow property specifies how much a flex item will grow relative to the rest of the flex items.

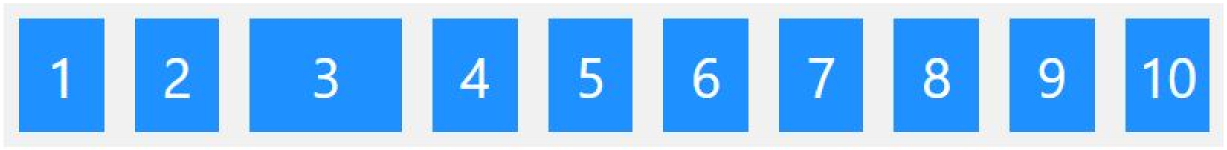


The value must be a number, default value is 0.

```
<div class="flex-container">
  <div style="flex-grow: 1">1</div>
  <div style="flex-grow: 1">2</div>
  <div style="flex-grow: 8">3</div>
</div>
```

flex-shrink Property

The flex-shrink property specifies how much a flex item will shrink relative to the rest of the flex items.



The value must be a number, default value is 1.

```
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div style="flex-shrink: 0">3</div>
  <div>4</div>
  <div>5</div>
  <div>6</div>
  <div>7</div>
  <div>8</div>
  <div>9</div>
  <div>10</div>
</div>
```

flex-basis Property

The flex-basis property specifies the initial length of a flex item.

```
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div style="flex-basis: 200px">3</div>
  <div>4</div>
</div>
```

flex Property

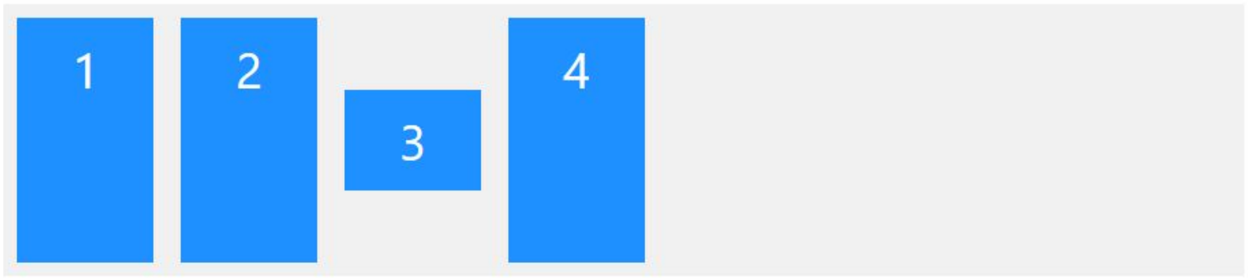
The flex property is a shorthand property for the flex-grow, flex-shrink, and flex-basis properties.

```
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div style="flex: 0 0 200px">3</div>
  <div>4</div>
</div>
```

align-self Property

The align-self property specifies the alignment for the selected item inside the flexible container.

The align-self property overrides the default alignment set by the container's align-items property.

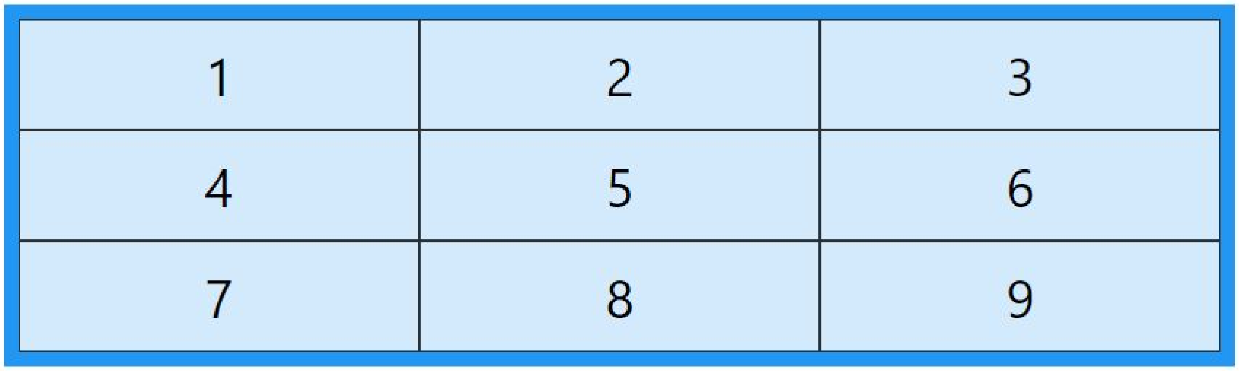


In these examples we use a 200 pixels high container, to better demonstrate the align-self property:

```
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div style="align-self: center">3</div>
  <div>4</div>
</div>
```

Grid Layout

The CSS Grid Layout Module offers a grid-based layout system, with rows and columns, making it easier to design web pages without having to use floats and positioning.



HTML

```
<div class="grid-container">
  <div class="grid-item">1</div>
  <div class="grid-item">2</div>
  <div class="grid-item">3</div>
  <div class="grid-item">4</div>
  <div class="grid-item">5</div>
  <div class="grid-item">6</div>
  <div class="grid-item">7</div>
  <div class="grid-item">8</div>
  <div class="grid-item">9</div>
</div>
```

CSS

```
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto;
  background-color: #2196F3;
  padding: 10px;
}
.grid-item {
  background-color: rgba(255, 255, 255, 0.8);
  border: 1px solid rgba(0, 0, 0, 0.8);
  padding: 20px;
  font-size: 30px;
  text-align: center;
}
```

Display Property

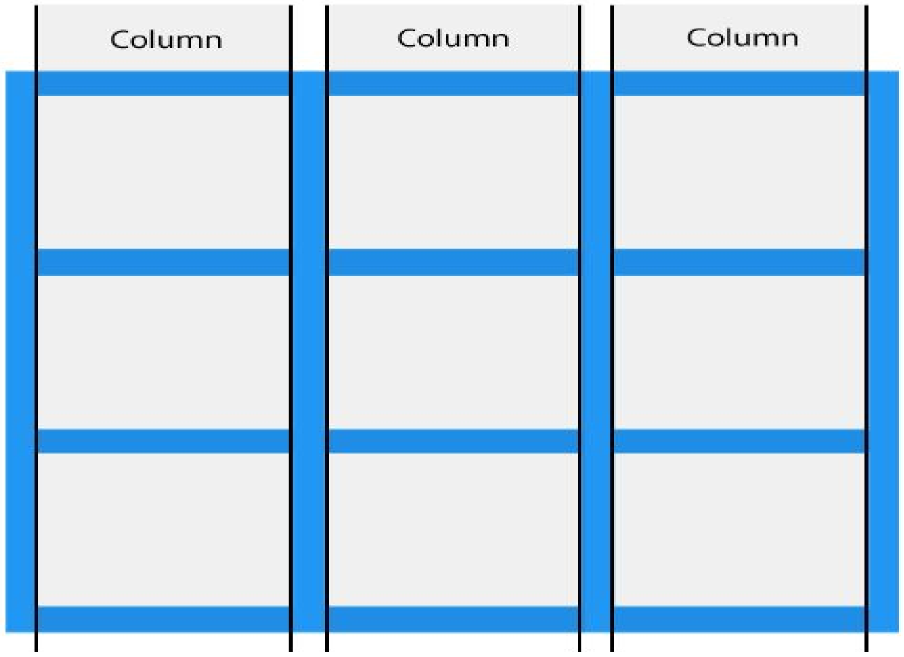
An HTML element becomes a grid container by setting the display property to grid or inline-grid.

```
.grid-container {
  display: grid;
}
/* or */
.grid-container {
  display: inline-grid;
}
```

All direct children of the grid container automatically become grid items.

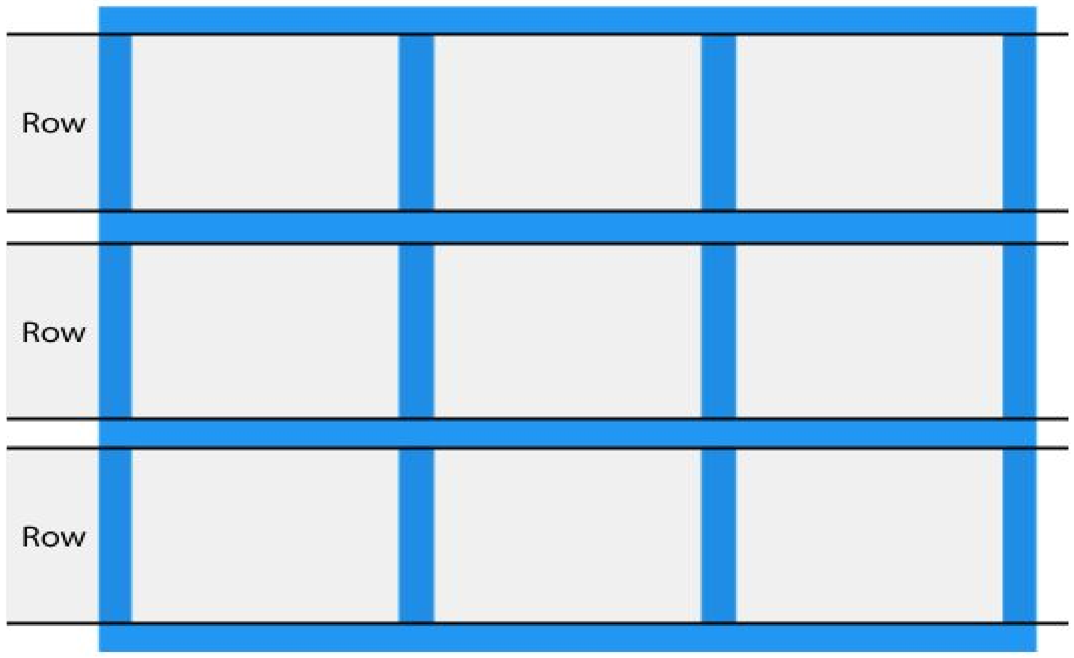
Grid Columns

The vertical line of grid items are called columns.



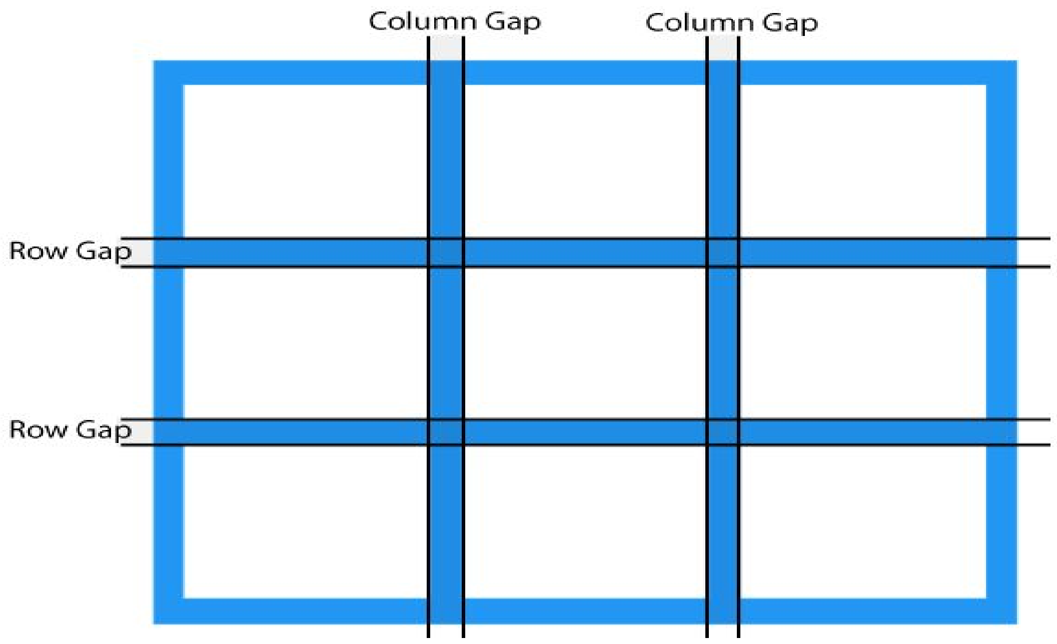
Grid Rows

The horizontal line of grid items are called rows.



Grid Gaps

The space between each column/row are called gaps.



You can adjust the gap size by using one of the following properties:

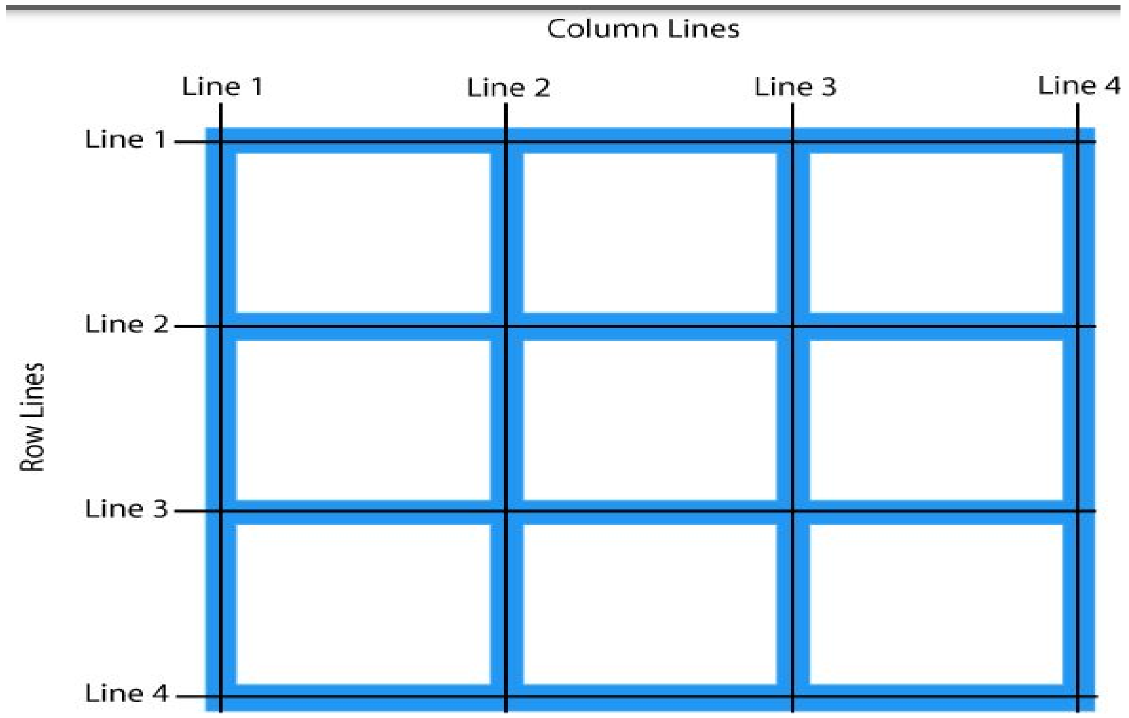
grid-column-gap

grid-row-gap
Grid-gap

Grid Lines

The line between columns are called column lines.

The line between rows are called row lines.



Refer to line numbers when placing a grid item in a grid container:

Example

Place a grid item at column line 1, and let it end on column line 3:

```
.item1 {
  grid-column-start: 1;
  grid-column-end: 3;
}
```

Place a grid item at row line 1, and let it end on row line 3:

```
.item1 {
  grid-row-start: 1;
  grid-row-end: 3;
}
```

grid-template-columns Property

The grid-template-columns property defines the number of columns in your grid layout, and it can define the width of each column.
The value is a space-separated-list, where each value defines the length of the respective column.

If you want your grid layout to contain 4 columns, specify the width of the 4 columns, or "auto" if all columns should have the same width.
Example
Make a grid with 4 columns:

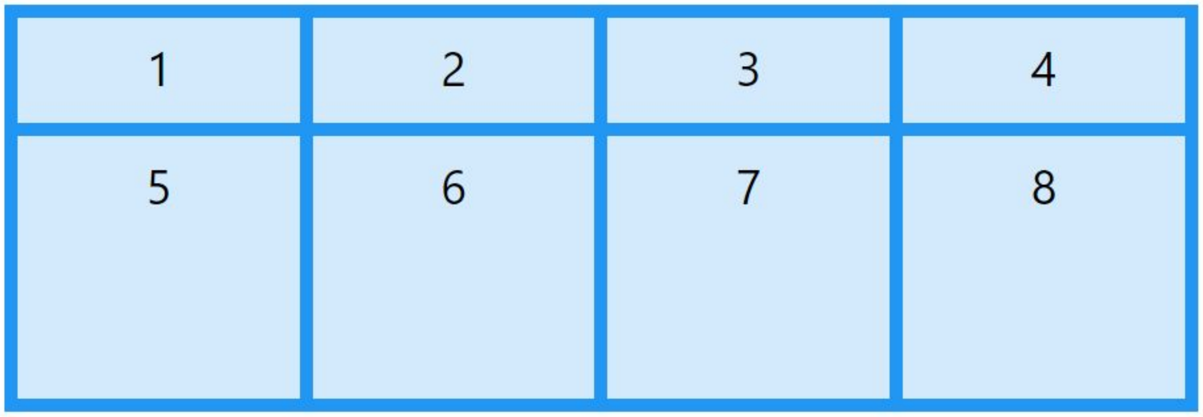
```
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto auto;
}
```

Note: If you have more than 4 items in a 4 columns grid, the grid will automatically add a new row to put the items in.
Example
Set a size for the 4 columns:

```
.grid-container {
  display: grid;
  grid-template-columns: 80px 200px auto 40px;
}
```

grid-template-rows Property

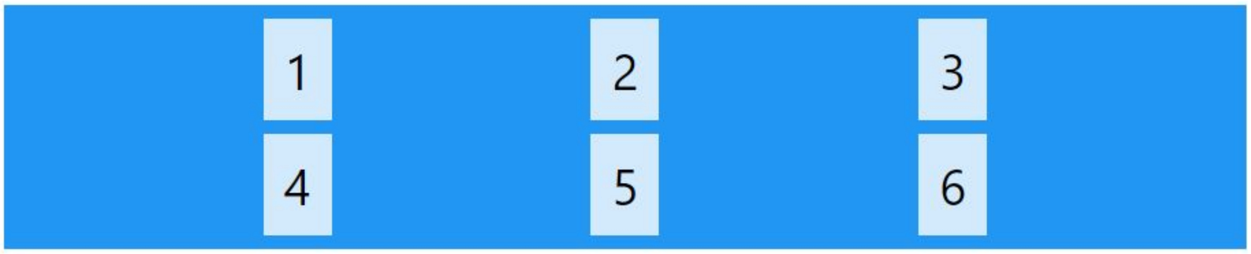
The grid-template-rows property defines the height of each row.



```
.grid-container {
  display: grid;
  grid-template-rows: 80px 200px;
}
```

justify-content Property

The justify-content property is used to align the whole grid inside the container.



Note: The grid's total width has to be less than the container's width for the justify-content property to have any effect.

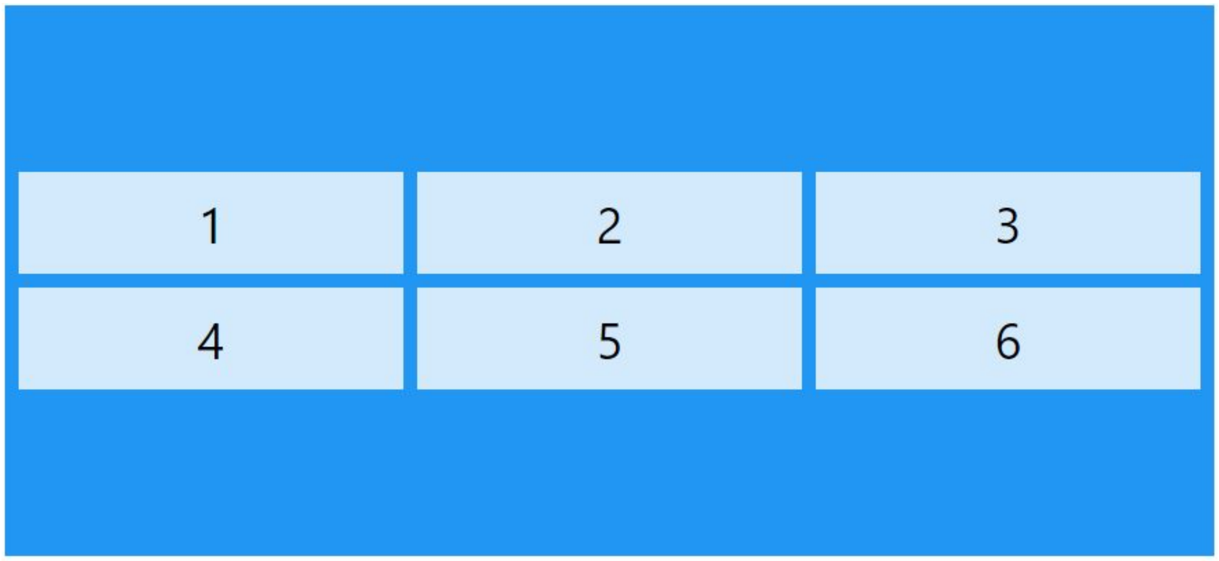
```
.grid-container {
  display: grid;
  justify-content: space-evenly;
}
```

It can have the following values:

- Space-around
- Space-between
- Center
- Start
- End

Align-content Property

The align-content property is used to vertically align the whole grid inside the container.



Note: The grid's total height has to be less than the container's height for the align-content property to have any effect.

```
.grid-container {
  display: grid;
  height: 400px;
  align-content: center;
}
```

It can have the following values:

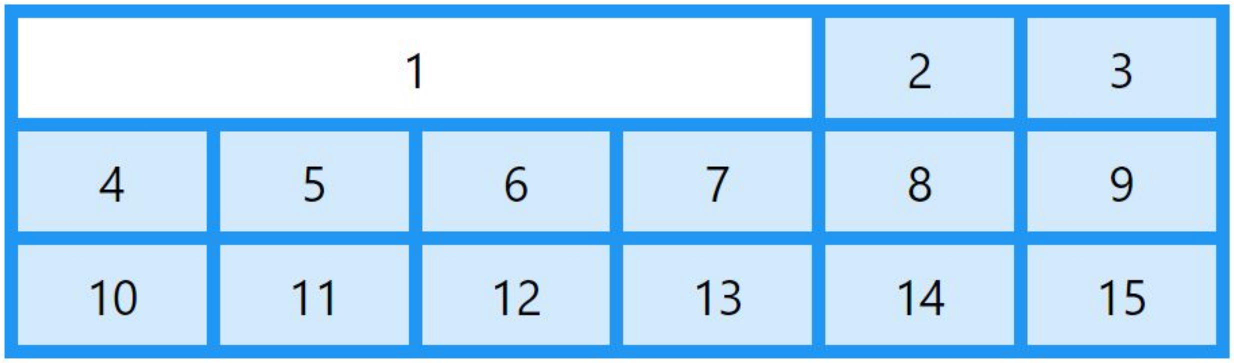
- Space-evenly
- Space-around
- Space-between
- Start
- End

Child Elements

grid-column Property

The grid-column property defines on which column(s) to place an item.

You define where the item will start, and where the item will end.



Example

Make "item1" start on line 1 and end on line 5:

```
.item1 {
  grid-column: 1 / 5;
}
```

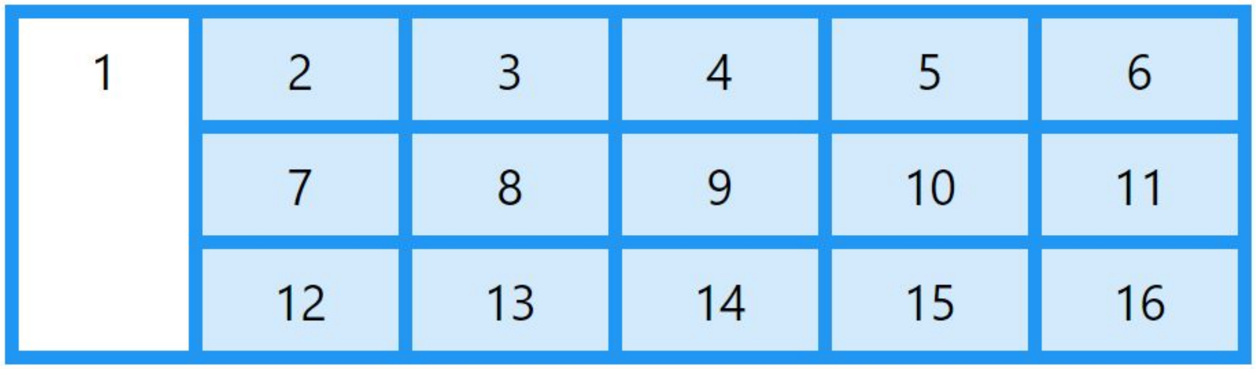
Make "item1" start on column 1 and span 3 columns:

```
.item1 {
  grid-column: 1 / span 3;
}
```

grid-row Property

The grid-row property defines on which row to place an item.

You define where the item will start, and where the item will end.



Note: The grid-row property is a shorthand property for the grid-row-start and the grid-row-end properties.

Example

Make "item1" start on row-line 1 and end on row-line 4:

```
.item1 {
  grid-row: 1 / 4;
}
```

Make "item1" start on row 1 and span 2 rows:

```
.item1 {
  grid-row: 1 / span 2;
}
```

HTML Multimedia

Multimedia comes in many different formats. It can be almost anything you can hear or see.

Examples: Images, music, sound, videos, records, films, animations, and more.

Web pages often contain multimedia elements of different types and formats.

In this chapter you will learn about the different multimedia formats.

Common Video Formats

- MPEG .mpg .mpeg
- .AVI
- WMV
- RealVideo .rm
- .ram
- Flash .swf
- .flv
- Ogg .ogg
- MPEG-4
- MP4 .mp4

Audio Formats

- MIDI .mid .midi
- RealAudio .rm
- .ram
- WMA .wma
- AAC .aac
- WAV .wav
- Ogg .ogg
- MP3 .mp3
- MP4

HTML5 Video

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
Your browser does not support the video tag.
</video>
```

The controls attribute adds video controls, like play, pause, and volume.

It is a good idea to always include width and height attributes. If height and width are not set, the page might flicker while the video loads.

The <source> element allows you to specify alternative video files which the browser may choose from. The browser will use the first recognized format.

The text between the <video> and </video> tags will only be displayed in browsers that do not support the <video> element.

Autoplay

To start a video automatically use the autoplay attribute:

```
<video width="320" height="240" autoplay>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
Your browser does not support the video tag.
</video>
```

Video - Browser Support

Browser	**MP4**	**WebM**	**Ogg**
Internet Explorer	YES	NO	NO
Chrome	YES	YES	YES
Firefox	YES	YES	YES
Safari	YES	NO	NO
Opera	YES (from Opera 25)	YES	YES

Video - Media Types

File Format	**Media Type**
MP4	video/mp4
WebM	video/webm
Ogg	video/ogg

HTML5 Audio

Before HTML5, audio files could only be played in a browser with a plug-in (like flash).

The HTML5 <audio> element specifies a standard way to embed audio in a web page.

```
<audio controls>
  <source src="audio.ogg" type="audio/ogg">
  <source src="audio.mp3" type="audio/mpeg">
Your browser does not support the audio element.
</audio>
```

The controls attribute adds audio controls, like play, pause, and volume.

The <source> element allows you to specify alternative audio files which the browser may choose from. The browser will use the first recognized format.

The text between the <audio> and </audio> tags will only be displayed in browsers that do not support the <audio> element.

Audio - Browser Support

Browser	**MP3**	**WAV**	**OGG**
Internet Explorer	YES	NO	NO
Chrome	YES	YES	YES
Firefox	YES	YES	YES
Safari	YES	YES	NO
Opera	YES	YES	YES

Audio - Media Types

MP3	audio/mpeg
OGG	audio/ogg
WAV	audio/wav

CSS Colors

Color Names

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>
<p style="background-color:Tomato;">Lorem ipsum...</p>
```

RGB Value

rgb(red, green, blue)
Each parameter (red, green, and blue) defines the intensity of the color between 0 and 255.