

Display Property

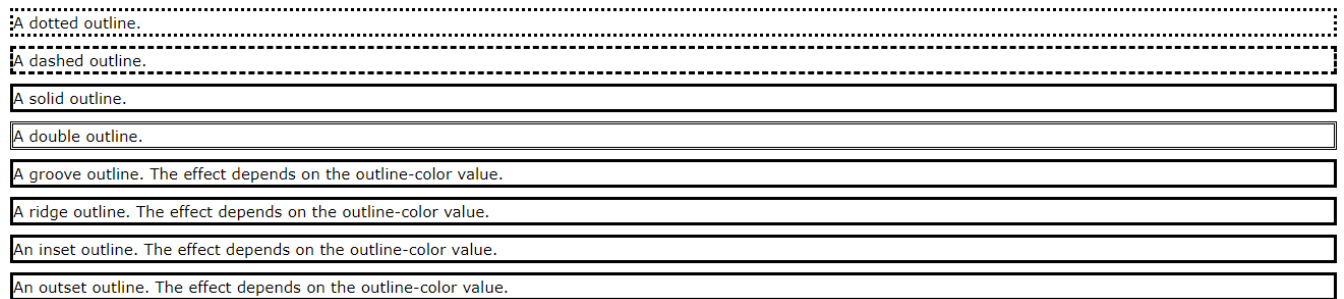
CSS Position

# Outline Style

The `outline-style` property specifies the style of the outline, and 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

The following example shows the different `outline-style` values:



# Setting The Viewport

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

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

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

## Media Queries Simple Examples

One way to use media queries is to have an alternate CSS section right inside your style sheet.

```
@media screen and (min-width: 480px) {  
  body {  
    background-color: lightgreen;  
  }  
}
```

### Example

```
/* Extra small devices (phones, 600px and down) */  
@media only screen and (max-width: 600px) {.example {background: red;}}  
  
/* Small devices (portrait tablets and large phones, 600px and up) */  
@media only screen and (min-width: 600px) {.example {background: green;}}  
  
/* Medium devices (landscape tablets, 768px and up) */  
@media only screen and (min-width: 768px) {.example {background: blue;}}  
  
/* Large devices (laptops/desktops, 992px and up) */  
@media only screen and (min-width: 992px) {.example {background: orange;}}  
  
/* Extra large devices (large laptops and desktops, 1200px and up) */  
@media only screen and (min-width: 1200px) {.example {background: pink;}}  
  
<p class="example">Resize the browser window to see how the background color of  
this paragraph changes on different screen sizes.</p>
```

# Using The width Property

If the `width` property is set to a percentage and the height is set to "auto", the image will be responsive and scale up and down:

## Example

```
img {  
  width: 100%;  
  height: auto;  
}
```

# Using The max-width Property

If the `max-width` property is set to 100%, the image will scale down if it has to, but never scale up to be larger than its original size:

## Example

```
img {  
  max-width: 100%;  
  height: auto;  
}
```

# Background Images

```
div {  
  width: 100%;  
  height: 400px;  
  background-image: url('img_flowers.jpg');  
  background-repeat: no-repeat; -> tekrarlama modları  
  background-size: contain; -> sadece içerik için  
  border: 1px solid red;  
}
```

## Example

```
div {  
  width: 100%;  
  height: 400px;  
  background-image: url('img_flowers.jpg');  
  background-size: cover; -> resmi tam boyut yaymak için.  
  border: 1px solid red;  
}
```

# Different Images for Different Devices

## Example

```
/* For width smaller than 400px: */
body {
  background-image: url('img_smallflower.jpg');
}

/* For width 400px and larger: */
@media only screen and (min-width: 400px) {
  body {
    background-image: url('img_flowers.jpg');
  }
}
```

# The display Property

Display: none;

Visibility: hidden;

## Example

```
h1.hidden {
  display: none; -> sayfada olmaması
}
```

## Example

```
h1.hidden {
  visibility: hidden; -> sayfada gizlenmesi
}
```

Display: inline; Display: block;

```
a {
  display: block; -> satır haline
}
```

```
li {
  display: inline; -> sütun halinde
}
```

# The position Property

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

There are five different position values:

- `static`
- `relative`
- `fixed`
- `absolute`
- `sticky`

## `position: fixed;`

### Example

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

## `position: relative-absolute;`

### Example

- ```
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;  
}
```

position: sticky;

## Example

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

## Overlapping Elements

### Example

```
img {  
  position: absolute; -> fixed'te aynı etkiyi veriyor -> ama aynı yerde kalır.  
  left: 0px;  
  top: 0px;  
  z-index: -1;  
}
```

## Positioning Text In an Image



opacity ?

top ?

right ?

bottom

# CSS Layout - Overflow

**visible** - Default. The overflow is not clipped. The content renders outside the element's box

**hidden** - The overflow is clipped, and the rest of the content will be invisible

**scroll** - The overflow is clipped, and a scrollbar is added to see the rest of the content

**auto** - Similar to **scroll**, but it adds scrollbars only when necessary

## Example

```
div {  
  width: 200px;  
  height: 50px;  
  background-color: #eee;  
  overflow: visible;  
}
```

# CSS Layout - float and clear

## Example

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

## Example

```
div {  
  clear: right;  
}
```

## Example

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

## Example

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



## Example

```
* {  
  box-sizing: border-box;  
}  
  
.box {  
  float: left;  
  width: 33.33%; /* three boxes (use 25% for four, and 50% for two, etc) */  
  padding: 50px; /* if you want space between the images */  
}
```

# Chania

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

## Images Side by Side

Float images side by side:



Footer Text