

## MORE CSS

```
<link  
href="filename"  
type="text/css"  
rel="stylesheet"  
>
```

# HTML ID ATTRIBUTE

```
<p>Coding Horror! Coding Horror!</p>  
<p id="mission">Our mission is to combine programming and  
<q>human</q> factors with geekiness!</p>
```

*HTML*

Coding Horror! Coding Horror!

Our mission is to combine programming and “human” factors with geekiness!

*output*

A unique ID for an element on a page

Each ID must be unique; can only be used once in the page

# LINKING TO SECTIONS OF A WEB PAGE

```
<p>Visit <a href=
"http://www.textpad.com/download/index.html#downloads">
textpad.com</a> to get the TextPad editor.</p>
<p><a href="#mission">View our Mission Statement</a></p>
```

*HTML*

Visit [textpad.com](http://www.textpad.com) to get the TextPad editor.

[View our Mission Statement](#)

*output*

Link target can include an ID at the end, preceded by a #

Browser will load that page and scroll to element with given ID

# CSS ID SELECTORS

```
#mission {  
font-style: italic;  
font-family: "Garamond", "Century Gothic", serif;  
}
```

CSS

Coding Horror! Coding Horror!

*Our mission is to combine programming and “human” factors with geekiness!*

output

Applies style only to the paragraph that has the ID of mission

# HTML CLASS ATTRIBUTE

```
<p class="shout">Coding Horror! Coding Horror!</p>  
<p class="special">See our special deal on Droids!</p>  
  
<p class="special">Today only!</p>
```

*HTML*

Coding Horror! Coding Horror!

See our special deal on Droids!

Today only!

*output*

A way to group some elements and give a style to only that group

Unlike an id, a class can be reused as much as you like  
on the page

# CSS CLASS SELECTORS

```
.special {  
background-color: yellow;  
font-weight: bold;  
}  
p.shout {  
color: red;  
font-family: cursive;  
}
```

CSS

Coding Horror! Coding Horror!

**See our special deal on Droids!**

**Today only!**

*output*

# CSS CLASS SELECTORS

```
<p class="shout">Coding Horror! Coding Horror!</p>  
<p class="special">See our special deal on Droids!</p>  
  
<p class="special shout">Today only!</p>
```

*HTML*

**Coding Horror! Coding Horror!**

**See our special deal on Droids!**

**Today only!**

*output*

# CSS ID SELECTORS: PSEUDO CLASSES

A pseudo-class is used to define a special state of an element. For example, it can be used to:

- Style an element when a user mouses over it
- Style visited and unvisited links differently
- Style an element when it gets focus

```
a:link { color: #FF0000; } /* unvisited link */  
a:visited { color: #00FF00; } /* visited link */  
a:hover { color: #FF00FF; } /* mouse over link */  
CSS
```

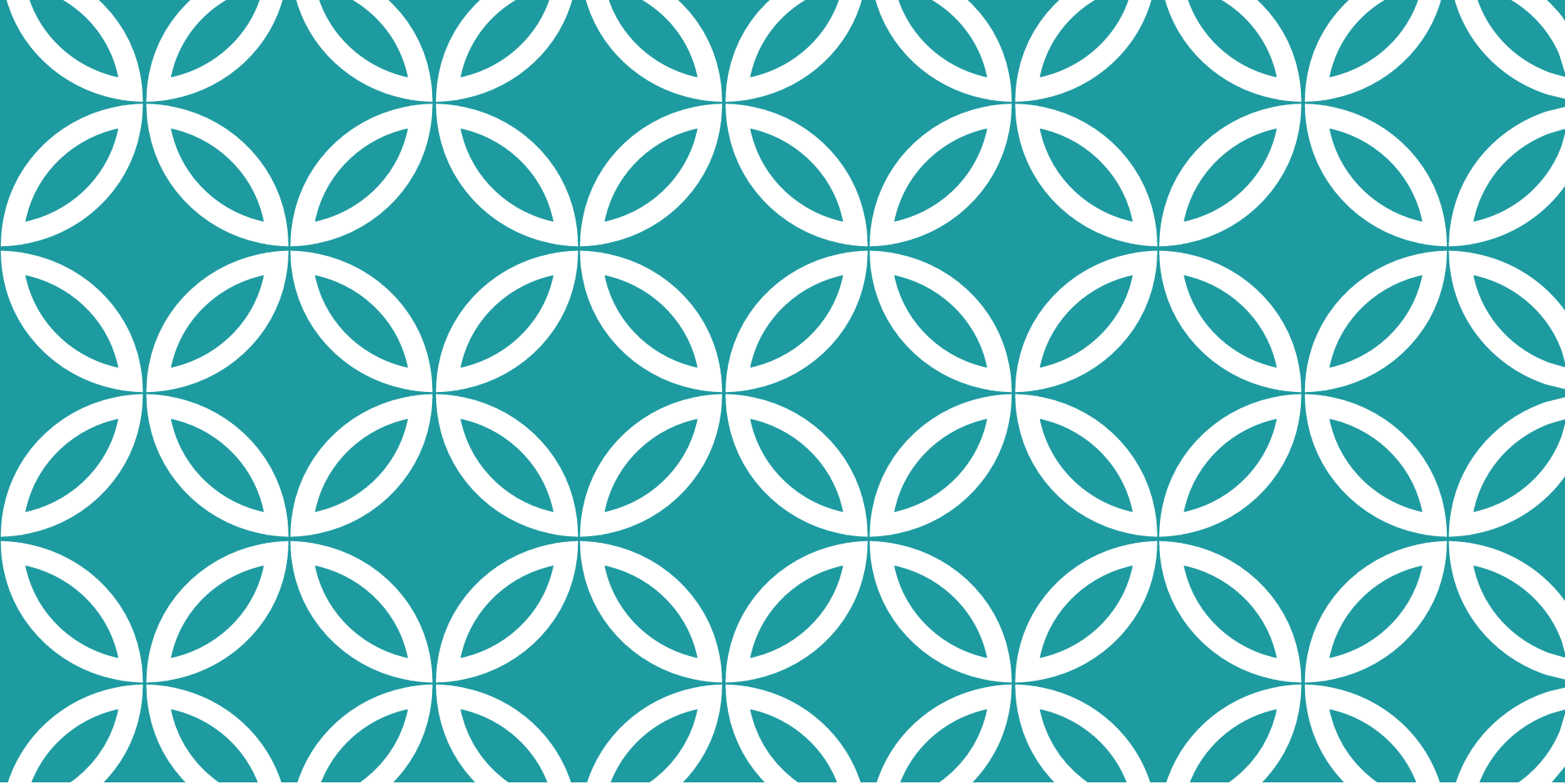
[Buy Early Buy Often!](#)

*output*



# CSS ID SELECTORS: PSEUDO ELEMENT

class	description
:active	an activated or selected element
:focus	an element that has the keyboard focus
:hover	an element that has the mouse over it
:link	a link that has not been visited
:visited	a link that has already been visited
:first-letter	the first letter of text inside an element
:first-line	the first line of text inside an element
:first-child	an element that is the first one to appear inside another

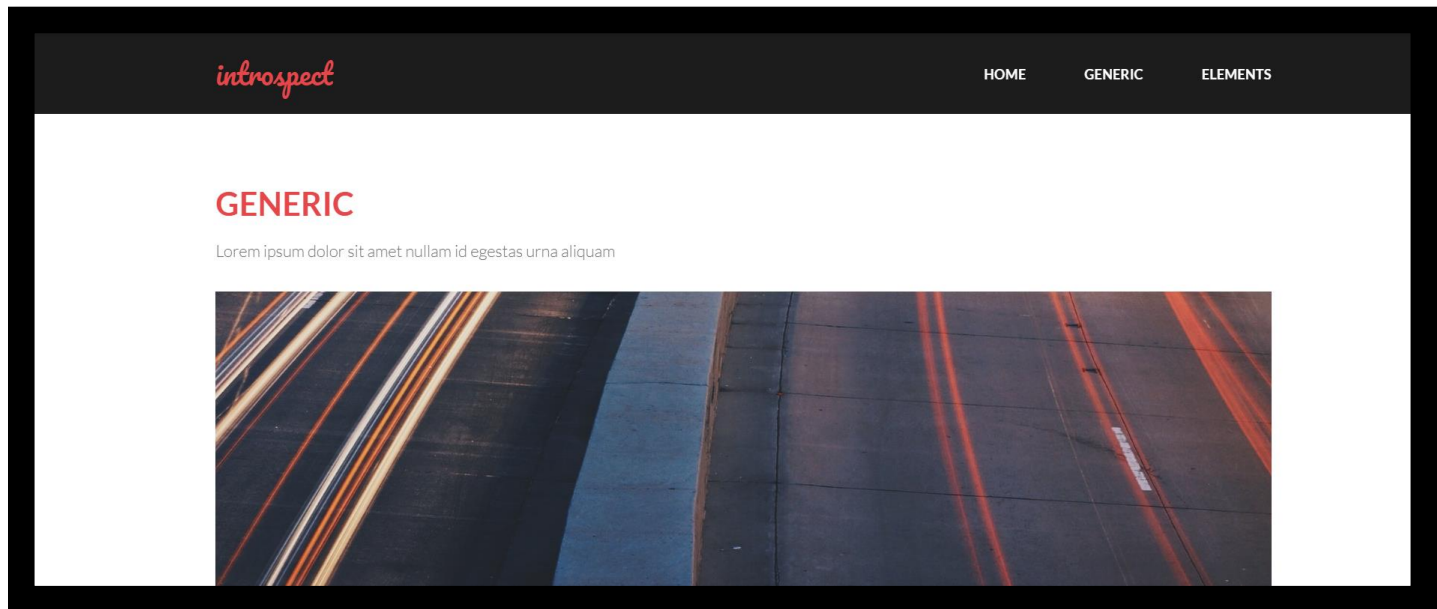


## STYLING PAGE SECTIONS

# WHY DO WE NEED PAGE SECTIONS?

Style individual elements, groups of elements, sections of text or of the page

Create complex page layouts



# SECTIONS OF A PAGE <DIV>

```
<div class="shout">  
<h2>Coding Horror! Coding Horror!</h2>  
<p class="special">See our special deal on Droids!</p>  
<p>We'll beat any advertised price!</p>  
</div>
```

HTML

**Coding Horror! Coding Horror!**

**See our special deal on Droids!**

We'll beat any advertised price!

*output*

Tag used to indicate a logical section or area of a page

Has no appearance by default, but you can apply styles to it

# INLINE SECTIONS <SPAN>

```
<h2>Coding Horror! Coding Horror!</h2>
<p>See our <span class="special">spectacular</span> deal
on Droids!</p>
<p>We'll beat <span class="shout"> any advertised
price</span>!</p>
```

HTML

**Coding Horror! Coding Horror!**

See our **spectacular** deal on Droids!

We'll beat **any advertised price!**

*output*

has no onscreen appearance, but you can apply a style or ID to it, which will be applied to the text inside the span

# CSS CONTEXT SELECTORS — COMBINATOR SELECTORS

```
selector1 selector2 {  
  properties  
}
```

CSS

applies the given properties to selector2 only if it is inside a selector1 on the page

```
selector1 > selector2 {  
  properties  
}
```

CSS

- applies the given properties to selector2 only if it is *directly* inside a selector1 on the page

# CONTEXT SELECTOR EXAMPLE

```
<p>Eat at <strong>Greasy's Burger</strong>...</p>
<ul>
<li>The <strong>greasiest</strong> burgers in town!</li>
<li>Yummy and greasy at the same time!</li>
</ul>
```

HTML

```
li strong { text-decoration: underline; }
```

CSS

Eat at **Greasy's Burger...**

- The **greasiest** burgers in town!
- Yummy and greasy at the same time!

*output*

# MORE COMPLEX EXAMPLE

```
<div id="ad">
<p>Eat at <strong>Greasy's Burger</strong>...</p>
<ul>
<li class="important">The <strong>greasiest</strong>
burgers in town!</li>
<li>Yummy and <strong>greasy at the same time
</strong>!</li>
</ul>
</div>
```

HTML

```
#ad li.important strong { text-decoration: underline; }
```

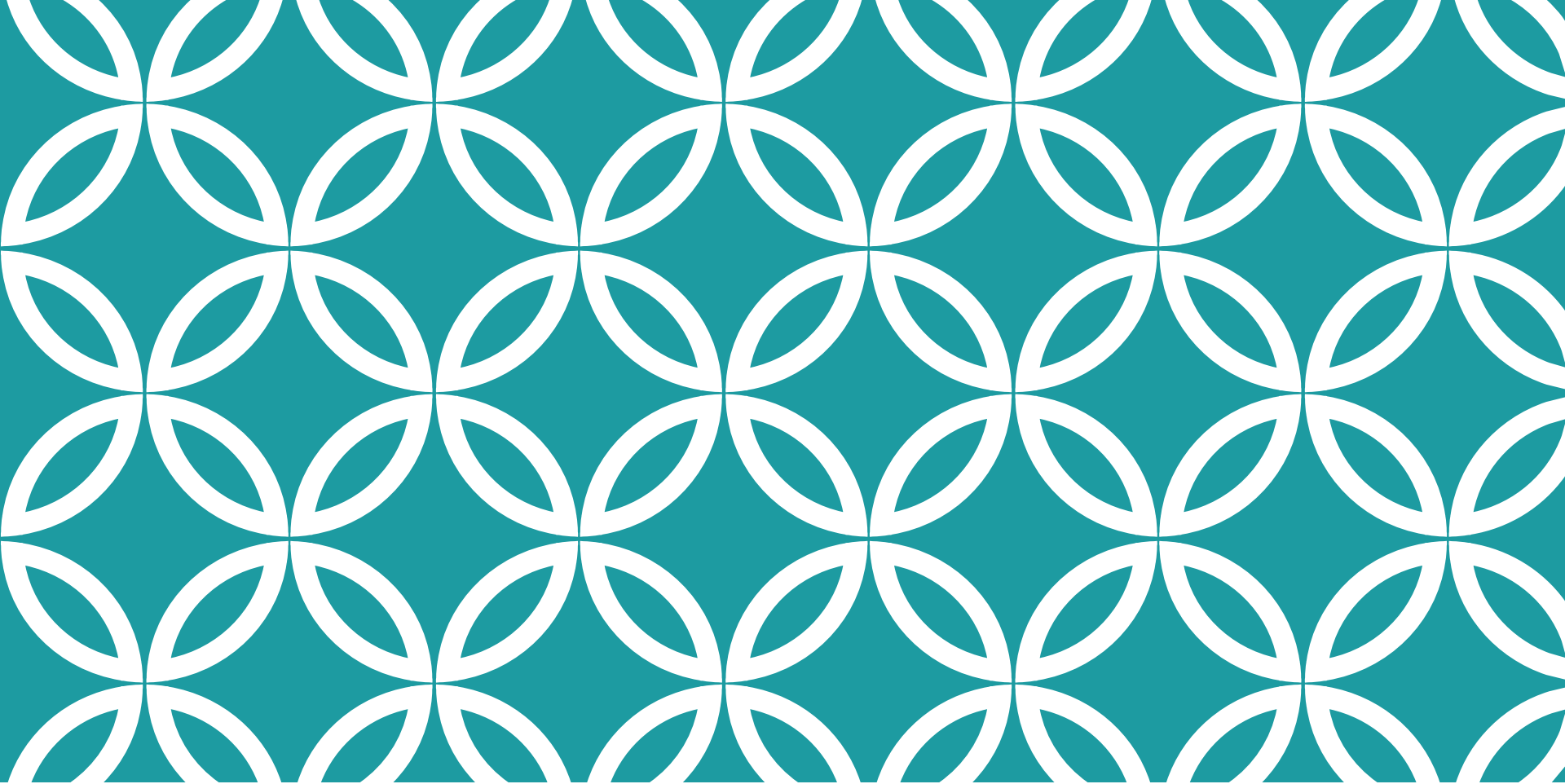
CSS

Eat at **Greasy's Burger...**

- The **greasiest** burgers in town!
- Yummy and **greasy at the same time!**

*output*





# FLOATING ELEMENTS

# THE CSS FLOAT PROPERTY (REFERENCE)

```
img.headericon {  
  float: right; width: 130px;  
}
```

CSS

Ghostbusters is a 1984 American science fiction comedy film written by co-stars Dan Aykroyd and Harold Ramis about three eccentric New York City parapsychologists-turned-ghost capturers.

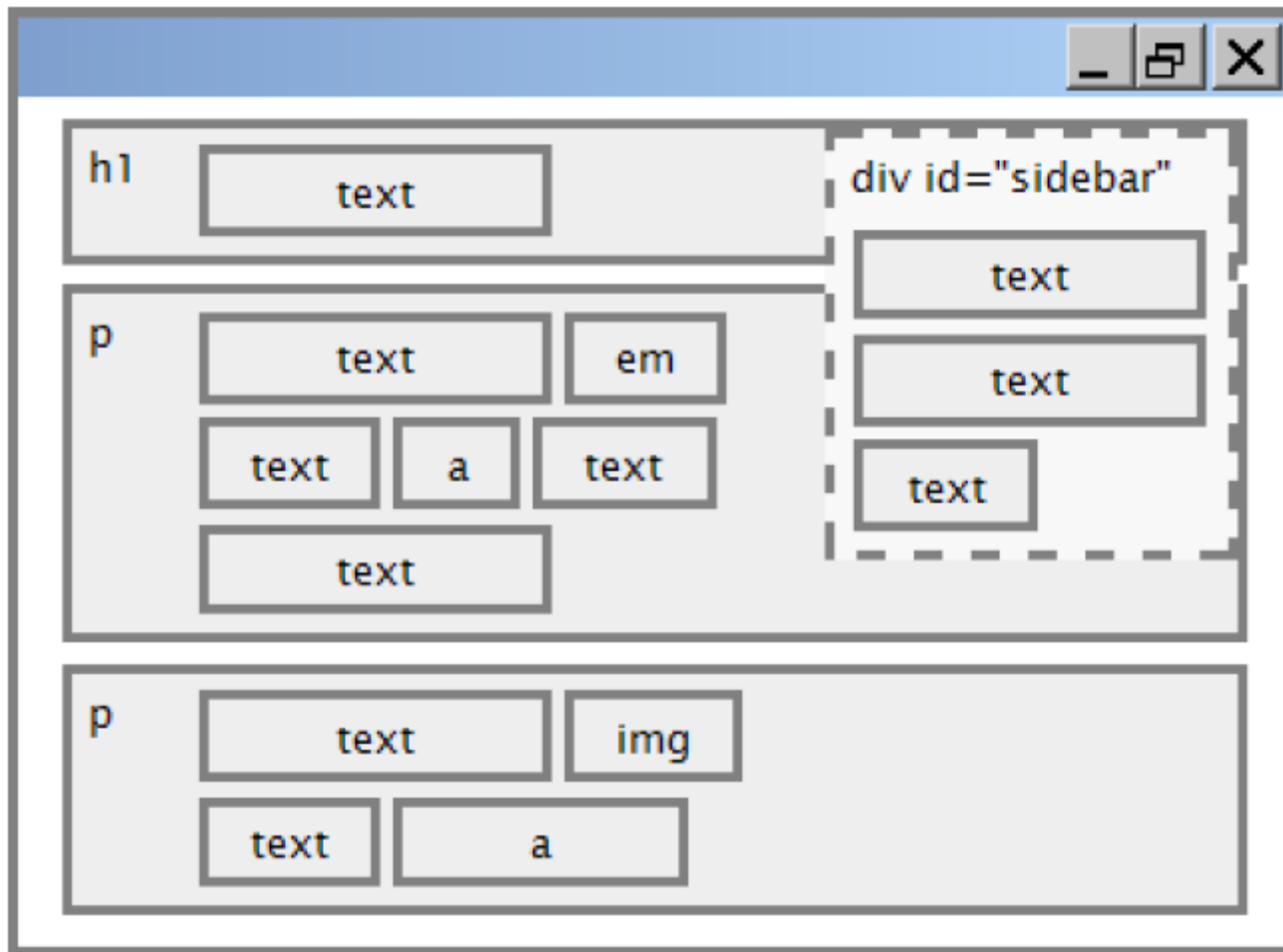


*output*

property	description
float	side to hover on; can be left, right, or none (default)

removed from normal document flow; underlying text wraps around as necessary

# FLOATING ELEMENTS DIAGRAM



# COMMON FLOAT BUG: MISSING WIDTH

I am not floating, no width

I am floating right, no width

I am not floating, 45% width

I am floating right, 45% width

often floating block elements must have a width property value

Let's try "floating"

# THE CLEAR PROPERTY (CONT.)

property	description
clear	disallows floating elements from overlapping this element; can be left, right, or none (default)

# THE CLEAR PROPERTY

```
p { background-color: fuchsia; }  
h2 { clear: right; background-color: yellow; }  
CSS
```

Mario is a fictional character in his video game series. Serving as Nintendo's mascot and the main protagonist of the series, Mario has appeared in over 200 video games since his creation



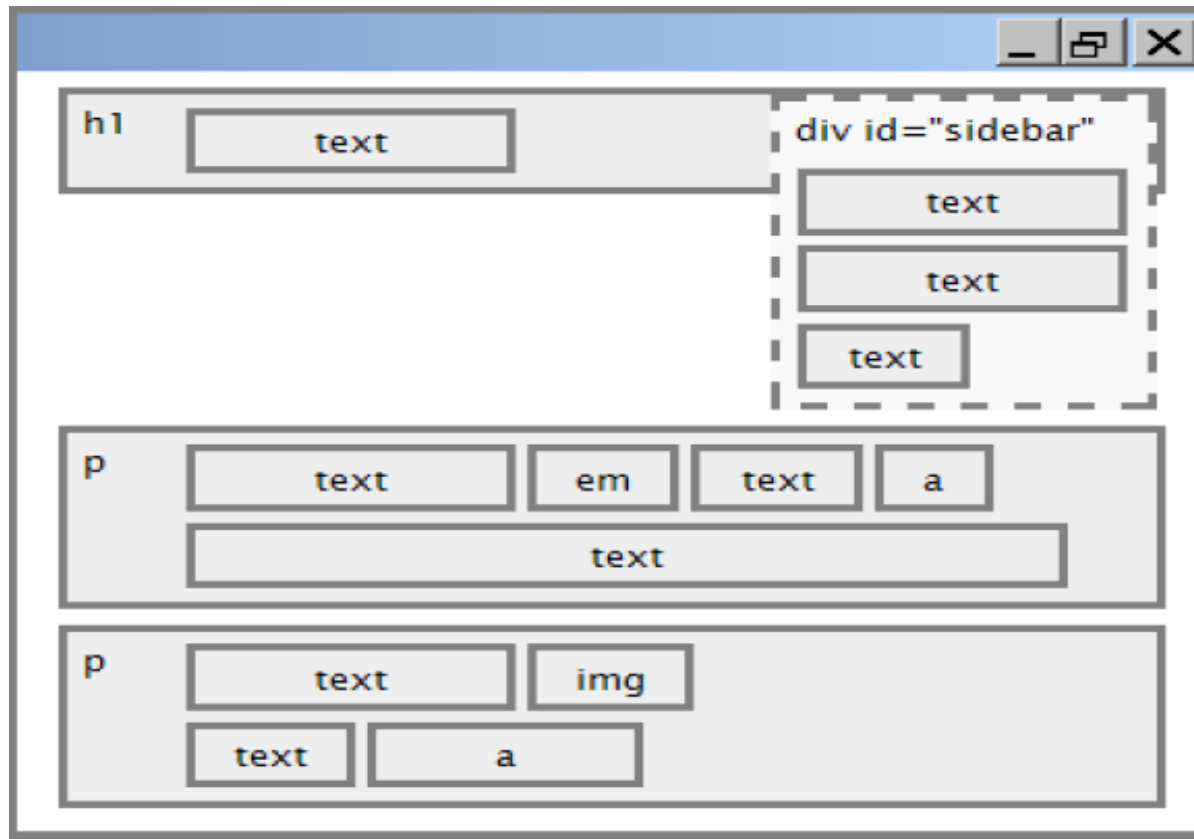
**Super Mario Fan Site!**

*output*

# CLEAR DIAGRAM

```
div#sidebar { float: right; }  
p { clear: right; }
```

CSS



# COMMON ERROR: CONTAINER TOO SHORT

```
<p>
Mario is a fictional character in his video game
series.....</p>
```

HTML

```
p { border: 2px dashed black; }
img { float: right; }
```

CSS

Mario is a fictional character in his video game series. Serving as Nintendo's mascot and the main protagonist of the series, Mario has appeared in over 200 video games since his creation.



output



# THE OVERFLOW PROPERTY

```
p { border: 2px dashed black;  
  overflow: hidden; }
```

CSS

Mario is a fictional character in his video game series. Serving as Nintendo's mascot and the main protagonist of the series, Mario has appeared in over 200 video games since his creation.



output

# THE OVERFLOW PROPERTY (CONT.)

property	description
overflow	specifies what to do if an element's content is too large; can be auto, visible, hidden, or scroll

# MULTI-COLUMN LAYOUTS

```
<div>
  <p>first paragraph</p>
  <p>second paragraph</p>
  <p>third paragraph</p>
  Some other text that is important
</div>
```

*HTML*

```
p { float: right; width: 25%; margin: 0.5em;
border: 2px solid black; }
div { border: 3px dotted green; overflow: hidden; }
```

*CSS*

Some other text that is important

third paragraph

second paragraph

first paragraph

*output*

# Z-INDEX

The **z-index** property specifies the stack order of an element.

```
<h1>The z-index Property</h1>

<p>Because the image has a z-index of -1, it will be
placed behind the heading.</p>
```

*HTML*

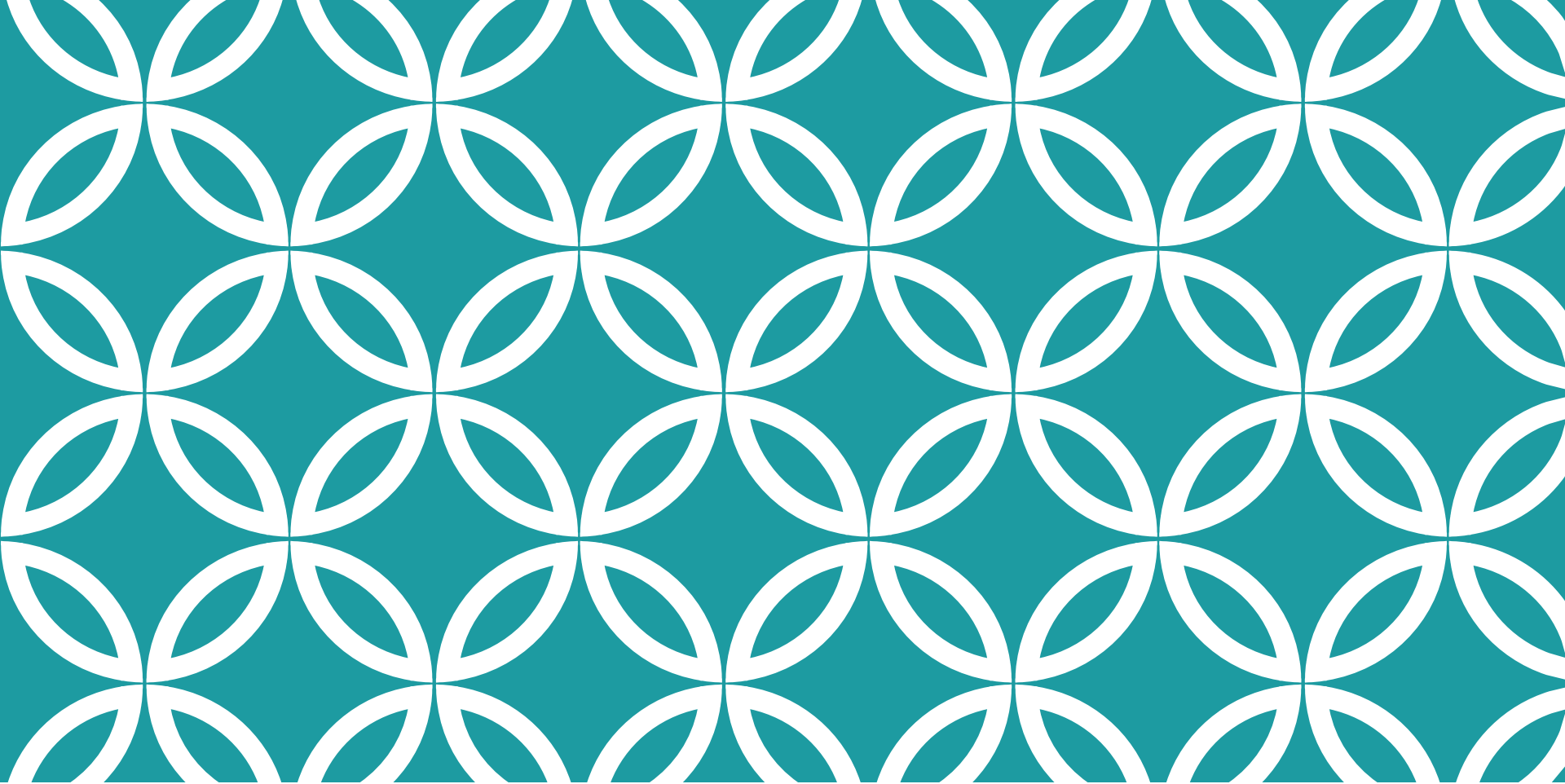
```
img { position: absolute;
left: 0px; top: 0px; z-index: -1;
}
```

*CSS*



## The z-index Property

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



# MEDIA QUERY

# WHAT IS THE VIEWPORT

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

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

```
<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.

# MEDIA QUERY

It uses the `@media` rule to include a block of CSS properties only if a certain condition is true.

```
<head>
<title>Media Queries Example</title>
<meta name="viewport" content="width=device-width,initial-scale=1.0">
<link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <div class="container">
    <header id="header"></header>
    <div class="box box1">Box 1</div>
    <div class="box box2">Box 2</div>
    <div class="box box3">Box 3</div>
    <div class="box box4">Box 4</div>
    <div class="clearfix"></div>

  </div>
</body>
```

HTML

# MEDIA QUERY

It uses the `@media` rule to include a block of CSS properties only if a certain condition is true.

```
* {  
  margin: 0px;  
}  
  
body, html {  
  width: 100%;  
  height: 100%;  
}  
  
.container {  
  width: 100%;  
  max-width: 1100px;  
  margin-left: auto;  
  margin-right: auto;  
  background-color: #ccc;  
}
```

CSS

```
#header {  
  height: 120px;  
  background-color: blue;  
}  
  
.box {  
  width: 50%;  
  height: 140px;  
  float: left;  
}
```

CSS

```
.box1 {  
  background-color: yellow;  
}  
.box2 {  
  background-color: green;  
}  
.box3 {  
  background-color: red;  
}  
.box4 {  
  background-color: orange;  
}  
.clearfix {  
  clear: both;  
}
```

CSS



# MEDIA QUERY

It uses the `@media` rule to include a block of CSS properties only if a certain condition is true.

```
/*screen for display screen or all for all type of devices*/  
/*This is for tablet*/  
@media screen and (max-width:768px){  
  .box1 {  
    width: 100%;  
  }  
  .box2 {  
    width: 100%;  
  }  
}  
  
/* Mobile */  
@media screen and (max-width:320px){  
  .box3{  
    width: 100%;  
  }  
  .box4{  
    width: 100%;  
  }  
}
```