



<div>

HTML TAGS & CSS SELECTORS

DEFINITION



- The `<div>` element defines logical divisions within the document.
- This means that when you use a `<div>` element, you are indicating that the enclosed content is a specific section of the page.
- The `<div>` element is typically used in XHTML+CSS documents to position and stylize portions of the page.

JUST LIKE OTHER ELEMENTS



- Box elements, or block-level elements are key ingredients to building well-formed XHTML documents that are cleanly laid out.
- Block elements are stand alone pieces of content, usually separated by newlines.
- Standard block-level elements are:
 - `<h1>`, `<h2>`, `<h3>`
 - `<p>`
 - ``

IN-LINE EXAMPLE

To position using in-line <div>



```
<div style="position:relative;left:50px;top:30px;height:100px;width:100px">
```

```
<p> some HTML content </p>
```

```
</div
```

DIV ID:

- Gives a unique name for a SINGLE element on an HTML/XHTML page
- A div ID name can only be used once on a single HTML page
- A div ID can be used on multiple pages
- Is always linked to CSS using “#” before the name of the tag



DIV ID:

HTML:

```
<div id="div1">  
  
<h1> example headline </h1>  
  
<p> example text </p>  
  
</div>
```

CSS:

```
#div1 {  
  
width:400px;  
  
height: 200px;  
  
}
```



DIV CLASS:



- Use IDs when there is only one occurrence per page.
- Use classes when there are one or more occurrences per page.
- If you have a similar style you want to use in a few separate places on a page you can enclose the information you want stylized in multiple locations in the same document within the same `<div class>`
- Identify the class in CSS using “.”

DIV CLASS:

HTML:

```
<div class="div2">  
  
<h1> example headline </h1>  
  
<p> example text </p>  
  
</div>
```

CSS:

```
.div1 {  
  
width:400px;  
  
height: 200px;  
  
}
```



SIZING DIV TAGS

`<div class> example...`



CSS:

```
.div1 {  
    width:400px;  
    height: 200px;  
}
```

POSITIONING DIV TAGS



position: static

- The default positioning for all elements is *position:static*, which means the element is not positioned and occurs where it normally would in the document.
- Normally you wouldn't specify this unless you needed to override a positioning that had been previously set.

```
#div-1 {  
    position:static;  
}
```

POSITIONING ATTRIBUTES

position:relative

- If you specify *position:relative*, then you can use *top* or *bottom*, and *left* or *right* to move the element relative to where it would normally occur in the document.
- To move div-1 down 20 pixels, and to the left 40 pixels:

```
#div-1 {  
    position:relative;  
    top:20px;  
    left:-40px;  
}
```



POSITIONING ATTRIBUTES

position:absolute

- When you specify *position:absolute*, the element is removed from the document and placed exactly where you tell it to go.
- To position div-1a at the top right of the page:

```
#div-1a {  
    position:absolute;  
    top:0;  
    right:0;  
    width:200px;  
}
```



POSITIONING ATTRIBUTES

position:absolute

- When you specify *position:absolute*, the element is removed from the document and placed exactly where you tell it to go.
- To move div-1a to the top right of the page:

```
#div-1a {  
    position: absolute;  
    top:0;  
    right:0;  
    width:200px;  
}
```



POSITIONING ATTRIBUTES

position: relative + position: absolute

- If we set *relative* positioning on div-1, any elements within div-1 will be positioned relative to div-1.
- Then if we set absolute positioning on div-1a, we can move it to the top right of div-1



position: relative + position: absolute

HTML

```
<div id="div-1">  
    some content  
    <div id="div-1a">  
        more content  
    </div>  
</div>
```

CSS

```
#div-1 {  
    position:relative;  
}  
  
#div-1a {  
    position:absolute;  
    top:0;  
    right:0;  
    width:200px;  
}
```



POSITIONING ATTRIBUTES



id = div-before

id = div-1

id = div-1a

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Integer pretium dui sit amet felis. Integer sit amet diam.
Phasellus ultrices viverra velit.

id = div-1b

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Integer pretium dui sit amet felis. Integer sit amet diam.
Phasellus ultrices viverra velit. Nam mattis, arcu ut bibendum
commodo, magna nisi tincidunt tortor, quis accumsan augue
ipsum id lorem.

id = div-1c

id = div-after

two column absolute

```
#div-1 {  
    position:relative;  
}
```

```
#div-1a {  
    position:absolute;  
    top:0;  
    right:0;  
    width:200px;  
}
```

```
#div-1b {  
    position:absolute;  
    top:0;  
    left:0;  
    width:200px;  
}
```



POSITIONING ATTRIBUTES



id = div-before

id = div-1b

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Integer pretium dui sit amet
felis. Integer sit amet diam.
Phasellus ultrices viverra velit.
Nam mattis, arcu ut bibendum
commodo, magna nisi tincidunt
tortor, quis accumsan augue
ipsum id lorem.

id = div-1a

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Integer pretium dui sit amet
felis. Integer sit amet diam.
Phasellus ultrices viverra velit.

id = div-after

STYLE PROPERTIES



- height
- width
- position
- top
- bottom
- left
- right
- margin
- padding
- background-color
- color
- font-weight
- text-align
- word

NESTING



- If the CSS is structured well, there shouldn't be a need to use many class or ID selectors. This is because you can specify properties to selectors *within* other selectors.
- With nesting, you can have different `<p>`, `<h1>`, and other HTML tags stylized differently through out the document by placing different sections within different `<div>` tags.

NESTING

```
<div id="div1">
```

```
<h1>example heading</h1>
```

```
<p>example content</p>
```

```
</div>
```



NESTING



```
#div1 {  
    background-color: #ccc;  
    padding: 1em  
}  
#div1 h1 {  
    color: #ff0;  
}  
#div1 p {  
    color: red;  
    font-weight: bold;  
}
```

PSEUDO CLASSES

- **Pseudo classes** are bolted on to selectors to specify a state or relation to the selector.
- Getting a hover color on your links would happen with pseudo classes.



PSEUDO CLASSES: HTML



```
<div class="div1">
```

```
<a href="towson.edu"> TOWSON UNIVERSITY </a>
```

```
</div>
```


PSEUDO CLASSES: CSS



```
a.div1:link {  
    color: blue;  
}  
  
a.div1:visited {  
    color: purple;  
}  
  
a.div1:active {  
    color: red;  
}  
  
a.div1:hover {  
    text-decoration: none;  
    color: blue;  
    background-color: yellow;  
}
```

SPAN TAGS (IN-LINE STYLES)



- The `` tag has very similar properties to the `<div>` tag, in that it changes the style of the text it encloses.
- But without any style attributes, the `` tag won't change the enclosed items at all.
- The primary difference between the `` and `<div>` tags is that `` doesn't do any formatting of its own.
- The `<div>` tag acts as a paragraph break, because it is defining a logical division in the document.
- The `` tag simply tells the browser to apply the style rules to whatever is within the ``.

SPAN TAGS



- Use `` when you want to change the style of elements without placing them in a new block-level element in the document. For example, if you had a Level 3 Heading (`<h3>`) that you wanted the word “change” above to be red, you could surround that word with

```
<span style="color : #f00;">change</span>
```

- It would still be a part of the `<h3>` tag, just red.