



COMMON HTML TAGS

name attribute(s)

↓ ↓

```
<p class="my-class"> ← opening
    Some text         ← contents
</p> ← closing
```

Paragraph

```
<p>Paragraph of text</p>
```

Styling

```
<em>Italic</em> and
<strong>bold</strong>
```

Image

```

```

Link (aka "anchor")

```
<a href="http://google.com/">
    Link to Google
</a>
```

Headers

```
<h1>Huge</h1> <h2>Big</h2>
... <h6>Small</h6>
```

Freeform

```
<div></div>            (block)        and
<span></span>           (inline)
```

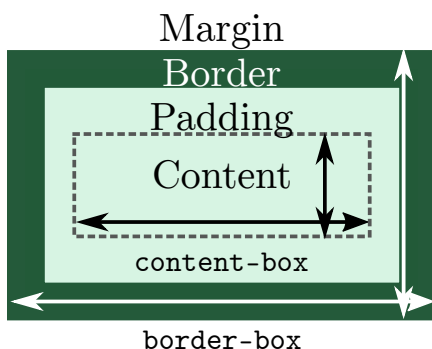
Semantic

```
<section>, <article>, and more
```

Comment

```
<!-- Ignored text -->
```

BOX MODEL



CSS EXAMPLES

selector

↓

```
p {
  font-size: 16pt;
}
```

↑ ↑

property value

Text

```
p {
  font-size: 16pt;
  text-align: center;
  color: green;
  font-family: "Arial";
  font-weight: bold;
  line-height: 20px;
}
```

Sizing

```
div {
  height: 30px;
  width: 100%;
  margin-left: 20px;
  padding-top: 10px;
}
```

Block styling

```
.highlighted-area {
  display: block;
  box-sizing: border-box;
  background: yellow;
  border: 2px solid green;
  border-radius: 10px;
}
```

Transition (animates properties)

```
.fading-link {
  transition: color 1s,
             padding-left 3s;
}

.fading-link:hover {
  color: green;
  padding-left: 20px;
}
```

CSS DISPLAY

block fill available width, word-wrap inline children (e.g. paragraph)

inline word-wrap (e.g. strong)

inline-block square block within word-wrapped text (e.g. emoji or icon)

flex line up children, control spacing

grid position children in grid

CSS SELECTORS

Tag `div { color: blue; }`

Class `.class-name { color: blue; }`

ID `#id_name { color: blue; }`

Containment (match children)

```
/* Match all divs in #a */
#a div { color: blue; }
/* Divs immediately in #b */
#b > div { color: blue; }
/* Wildcard: anything in #c */
#c * { color: blue; }
```

Pseudo-elements (add content)

```
h1::before { content: "-"; }
```

Pseudo-classes (match by position)

```
/* First p margin */
p:nth-child(1) { margin: 10px; }
/* Striped table rows */
tr:nth-child(odd) { color: gray; }
```

GRID CONTAINER

grid-template-columns grid-template-rows

	1fr 1fr 1fr	1fr 1fr 1fr
	50px auto 50px	50px auto 90px
	20% 20% auto	100px auto 20%

```
.container-element {
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
  grid-template-rows: 50px auto 90px;
}
```

GRID CHILD POSITIONING

	grid-column	grid-row
	2 / 3	
	2 / span 2	
		1 / 3
	1 / span 2	1 / span 2

```
.child-element {
  grid-column: 1 / span 2;
  grid-row: 1 / span 2;
}
```