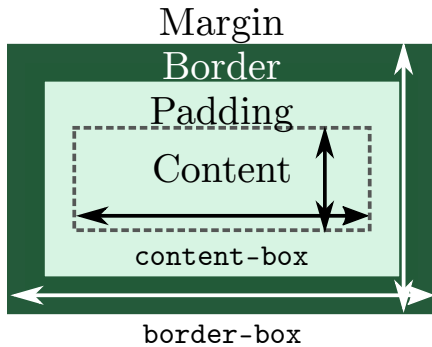


BOX MODEL



CSS TERMINOLOGY

selector
 ↓
 p {
 ↙ declaration block
 font-size: 16pt;
 }
 ↑
 property value

Selector “Targets” a CSS rule

Specificity The more specific the selector it is, the more likely it is to win against other selectors targeting the same element

Responsive design Looking good on mobile *and* desktop

Media queries If statements for CSS, apply CSS to browser sizes.

DOM TREE TERMINOLOGY

- Parent: `<div class="Main">`
 - Sibling: `<div>Sib</div>`
 - **Element:** `<div>`
 - * Child: `<p>a</p>`
 - * Child: `<hr />`
 - * Child: `<p>b</p>`
- Sibling: ``

CSS VARIABLES

```
:root { /* Definition */
  --bg: #FEAAFE;
  --gap: 20px;
}
.MainContent { /* Use */
  background: var(--bg);
  margin: calc(var(--gap) + 5px);
}
```

GRID CONTAINER

Use `display: grid` and specify columns/rows.

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 {
  display: grid;
  grid-template-columns: 50px auto 50px;
  grid-template-rows: 50px auto 90px;
}
```

GRID CHILD POSITIONING

Optional: Child elements can be custom positioned and sized.

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

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

CSS “PSEUDOS”

Pseudo-elements

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

Pseudo-selectors

```
a:hover { color: blue; }

/* First p margin */
p:nth-child(1) {
  margin-left: 10px;
}

/* Striped table rows */
tr:nth-child(odd) {
  background: gray;
}
```

CSS ANIMATION

```
.Link {
  color: blue;
  padding-left: 10px;
  transition: color 1s,
              padding-left 3s;
}
.Link:hover {
  color: green;
  padding-left: 20px;
}
```

ADDING JAVASCRIPT TO HTML

Responding to clicks

```
<div onclick="alert('Hello')">
  Click for annoying pop-up
</div>
```

Modifying DOM

```
<div id="a" onclick="
  document.querySelector('#a')
  .innerHTML += '+'
">Plus</div>
```

Running on page load

```
<script src="http://ex.amp/le.js">
</script>
<script> // Print to dev
  console.log("hi"); // tools console
</script>
```

BEM

Block-Element-Modifier Naming convention for CSS

Block Standalone entity meaningful on its own. Use camel-case: `ComponentName`.

Element A single part of a block that has no standalone meaning. Use a dash: `ComponentName-descendentName`.

Modifier A variant of a block (or element). Use two dashes: `ComponentName--modifier`.

```
.OkBtn { background: orange; }
.OkBtn-icon { width: 12px; }
.OkBtn--disabled { color: gray; }

<div class="OkBtn--disabled">
  
  Okay
</div>
```