

Lesson 5: Positioning Content

Positioning with Floats

- **float property** = allows us to take an element and position it to the left or right of its parent element
 - all other elements will flow around the floated element (eg. floated img left, paragraphs wrap around img)
 - alters element's default display value to block
 - multiple floats → create layout (eg. multi-column)
- values: left
right
- **clear property** = prevent content from wrapping around floated elements
 - values: left
right
both
- **containing floats** = like clearing, but ensures styles will be rendered properly
 - floated element must be inside a parent element as a container
 - everything outside of container flows normally
- common class name: clearfix
cf
- format:

<pre>class name: before, .class name: after { content: ""; display: table; } .class name: after { clear: both; } .class name { clear: both; *zoom: 1; } element { float: left/right; margin: #; width: #; }</pre>	→	<pre>group: before, .group: after { content: ""; display: table; } .group: after { clear: both; } .group { clear: both; *zoom: 1; } section { float: left, margin: 0 1.5%; width: 63%; } aside { float: right; margin: 0 1.5%; width: 30%; }</pre>
---	---	--

Positioning with Inline-Block

- primarily helpful for laying out pages or placing elements next to another w/in a line
- inline-block elements include a single space between them
- remove space between inline-block elements: put new opening tag on same line as previous closing tag
open an HTML comment directly after a closing tag & close it right before the next opening tag

→ (eg. same line): <section>

```
...  
</section> <section>  
...  
</section>
```

→ (eg. comment): <section>

```
...  
</section><!--  
--><section>  
...  
</section>
```

Creating Reusable Layouts

- inline-block elements → create grid/layout of page
- float → wrap content around an element
- other helpful tools in CSS: flex-based properties
grid-based properties

Uniquely Positioning Elements

- **position property** = identifies how an element is positioned on a page
used w/ box offset properties: t, r, b, l
default value: static
 - values: relative
absolute
 - **box offset properties** = identifies exactly where an element will be positioned
- **relative positioning** = allows elements to appear w/in the normal flow, leaving space for an element as intended w/o allowing other elements to flow around it
allows an element's display position to be modified w/ box offset properties
 - element overlaps
- **absolute positioning** = element won't appear in normal flow, space & position of element won't be preserved
 - won't overlap surrounding elements & other elements can occupy the non-preserved space
 - moved in relation to closest relatively positioned parent element
 - if DNE → positioned in relation to <body> element