Pseudo Selectors

- CSS selectors are typically document-related
  - Reflect an element's context in the page
    - Nesting, containing block, tag, class(es) ...
- What about selectors for other characteristics?
  - The first letter of some text
  - Links that have been visited
- Can use pseudo-classes and pseudo-elements
  - Pseudo characteristic preceded by : in selector

```
p:pseudo { ... :pseudo { ... } }
```
Pseudo Selectors

- Some pseudo-elements
  - `:first-letter` and `:first-line`
    - Applied to first letter/line of text in a block

- Some pseudo-classes
  - `:link` and `:visited`
    - Applied to anchors that have (not) been visited
  - `:hover`
    - Applied to elements the cursor is hovering over
  - `:before` and `:after`
    - Use with `content: text value`
    - Inserts `text value` before/after the element's content

Display

- Renders an element differently

- `display`
  - Specifies rendering category for an element
  - Value: `block`, `inline`, `inline-block`, `table-cell`, etc.
    - IE7 and earlier only support a few of these

- Does not actually make the element something else
  - Just displays it as such

- `inline-block` is interesting
  - “An inline-level block container”
  - The element itself is considered inline
  - The “inside” of the element is considered a block
Within <body>

- Block elements are arranged vertically
- Placed in order given
- Occupy width of page

<body>
  <h1>...</h1>
  <p>...</p>
  <ol>...</ol>
</body>

Within block elements

- Inline elements are arranged horizontally
- Left to right
- Wrap as lines fill

<p>
  Text
  <strong> ... </strong>
  Text
  <a> ... </a>
  Text
</p>
Floating Elements

- Removed from standard flow
  - Vertical position unchanged
  - Horizontal position specified
- **Oriented** in containing block
- `float`
  - Specifies a floating element
  - Value: left, right or none

```html
<p>
  Text <img class="floatRight"... />
  Text <a>... </a> Text
</p>
<p>More text</p>
```

Correct Use of Floats

- Not synonymous with alignment
  - Float relates to flow and placement
- Block elements are typically set as floats
  - HTML5 outline elements, *div*s, images
- Vertical position is unchanged
  - Same as with standard flow
  - Elements will wrap accordingly
Preventing Wrapping

- clear
  - Positions an element under floats
  - Value: left, right, both or none
    - Position of float to avoid

Fitting Floats

- Float may not fit in containing element
  - May be significantly larger
- Option: clearfix
  - Forces container to “clear after itself”
  - Series of CSS rules

```html
<p>
  Text <img class="floatRight"... />
  Text <a> ... </a> Text
</p>

<p class="clearfix">
  Text <img class="floatRight"... />
  Text <a> ... </a> Text
</p>
```
Some Clearfix Methods

“Micro-Clearfix”

```css
.clearfix:before, .clearfix:after {
  content: " ";
  display: table;
}
.clearfix:after {
  clear: both;
}
.clearfix {
  zoom: 1;
}
```

“Updated Classic”

```css
.clearfix:after {
  visibility: hidden;
  display: block;
  content: " ";
  clear: both;
  height: 0;
}
* html .clearfix { zoom: 1; }
*:first-child+html .clearfix {
  zoom: 1;
}
```

For IE 6 or 7

---

Some Clearfix Methods

“Clearfix Reloaded”

```css
.clearfix:after {
  content: " ";
  display: block;
  clear: both;
}
```
A Few Notes on Clearfix

- Not regarded as an ideal solution
- Try to design/apply styles so you can avoid it
- Alternatives
  - Apply `clear: both` to next block element
  - Try the `overflow` property
    - Specifies content overflow management
    - Values: `visible`, `hidden`, `scroll` or `auto`
    - `hidden` should resize block to fit (may not be guaranteed)

Creating Columns

- Use a `div` for each column and apply style(s)

```html
<div class="outerColumn">
  <div class="innerColumn"> ... </div>
</div>

* Repeat the above twice more

```
Creating Columns

- Use a div for each column and apply style(s)
  - Specify width and float direction
  - Columns will arrange horizontally
- May apply clear to following block
- Does not guarantee equal height
  - Can be done in CSS or with JavaScript
    - Decent support for display: flex in CSS3

```
outerColumn { innerColumn { .noWrap {
  float: left; margin-left: 5px; clear: both;
  width: 33.3%; margin-right: 5px; }
}
```