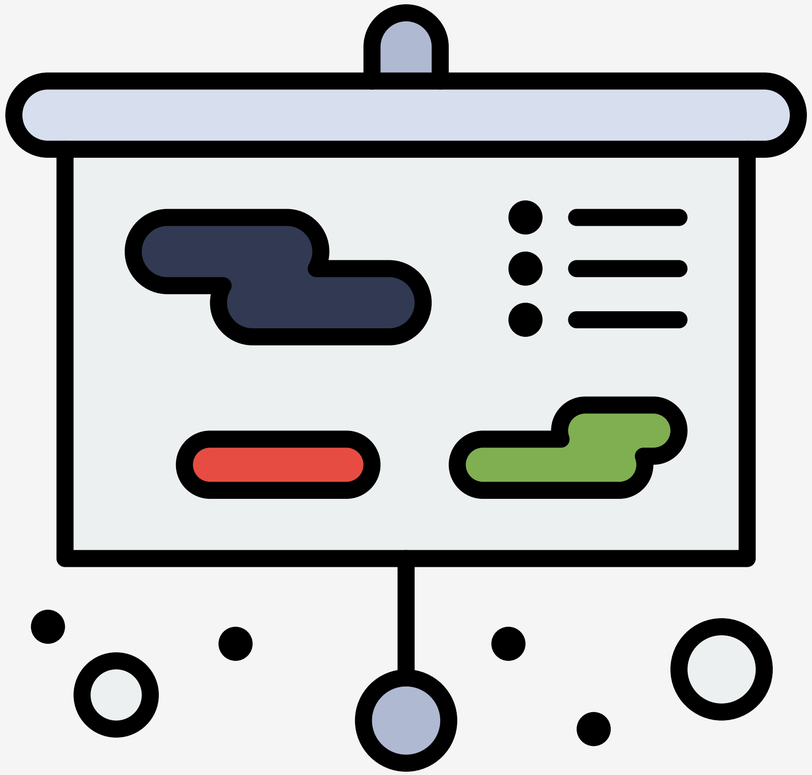
RESPONSIVE
WEB DESIGN II

Lecture 12

TODAY'S TOPICS



- CSS Complex Selectors
- CSS Pseudo-elements
- CSS Custom Properties
- CSS `calc()` Function

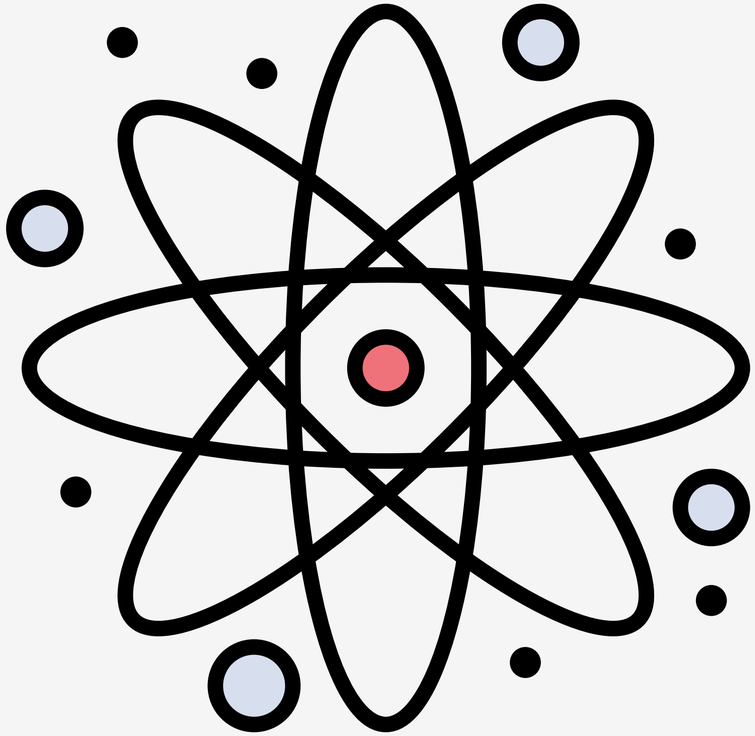
ANNOUNCEMENTS

- Sign-in Sheet

QUESTIONS

CSS COMPLEX SELECTORS

COMBINATORS

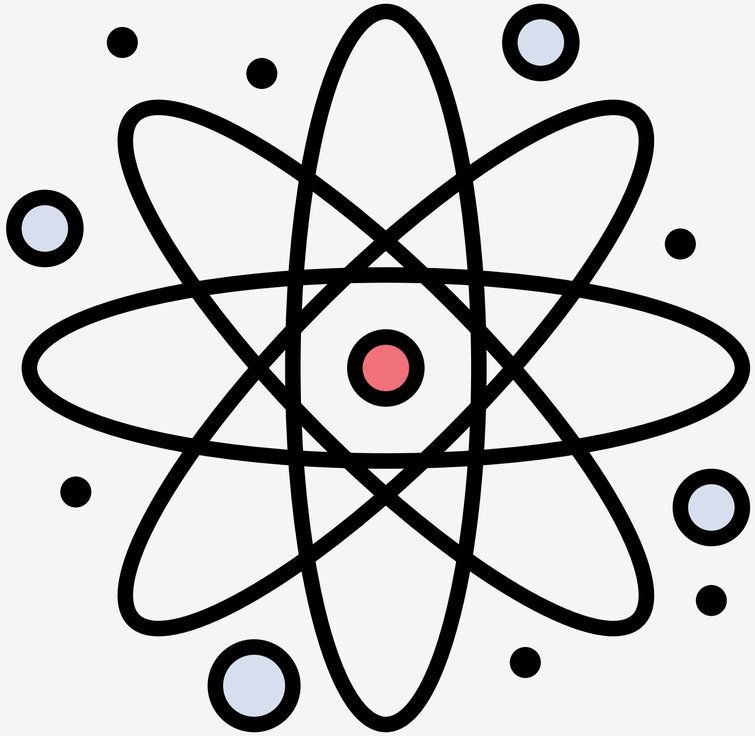


- Combinators are special type of selectors combine other selectors
- There are four types of combinators:
  - Descendant (space)
  - Child (>)
  - Adjacent Sibling (+)
  - Siblings (~)

COMBINATORS

```css
/* Any matching child */
.parent .child {
  color: indianred;
}

/* Any matching direct child */
.parent > .child {
  color: indianred;
}

/* The very next sibling */
.child + .child {
  color: cadetblue;
}

/* All following siblings */
.child ~ .child {
  color: cadetblue;
}
```

ATTRIBUTE SELECTORS

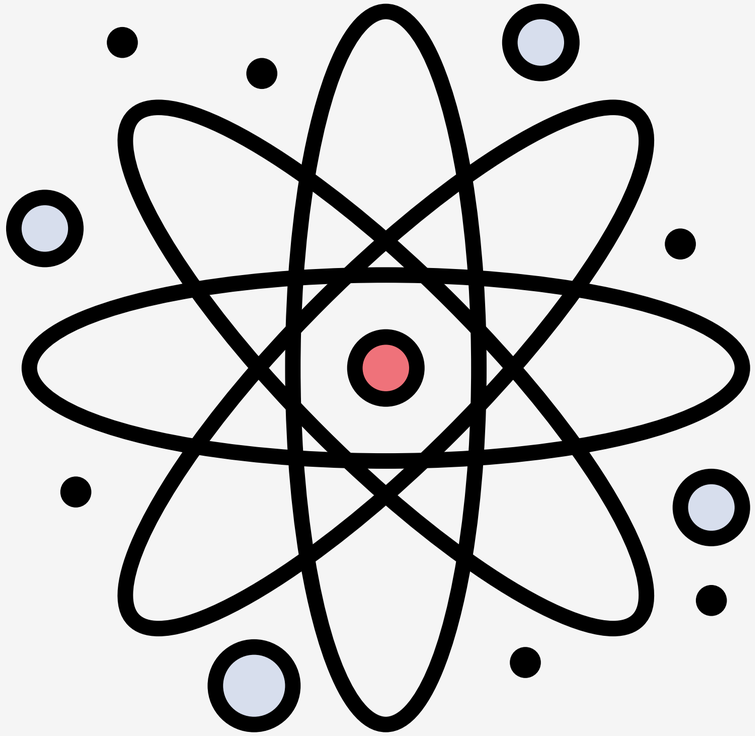


- Attribute selectors select an element based on the presence or value of an attribute
- Different flavours of the attribute selectors:
  - `[attr]`
  - `[attr="value"]`
  - `[attr^="value"]`
  - `[attr$="value"]`
  - `[attr*="value"]`

ATTRIBUTE SELECTORS

```css
/* <a> tag with an href */
a[href] {
  color: cadetblue;
}

/* <a> that links to https://google.ca */
a[href="https://google.ca"] {
  font-weight: bold;
}

/* external links */
a[href^="http"] {
  font-style: italic;
}

/* links to a PDF */
a[href$=".pdf"] {
  color: indianred;
}
```

PSEUDO-CLASSES



- Target elements based on their current state.
- `:link`, `:visited`
- `:hover`, `:active`
- `:first-child`, `:last-child`, `:nth-child`
- `:focus`, `:required`, `:optional`, `:valid`, `:invalid`, `:read-only`

PSEUDO CLASSES

```css
/* <a> with href */
a:link {
  color: cadetblue;
}

/* <a> that has been visited */
a:visited {
  color: indianred;
}

/* mouse cursor over <a> */
a:hover {
  background-color: goldenrod;
  color: white;
}

/* clicking down on <a> */
a:active {
  text-shadow: 0 0 3px rgba(0,0,0,0.5);
}
```

PSEUDO CLASSES

```css
/* matches the first child if a <p> tag */
p:first-child {
  color: cadetblue;
}

/* matches the last child if a <p> tag */
p:last-child {
  color: cadetblue;
}

/* matches the second child if a <p> tag */
p:nth-child(2) {
  color: cadetblue;
}
```

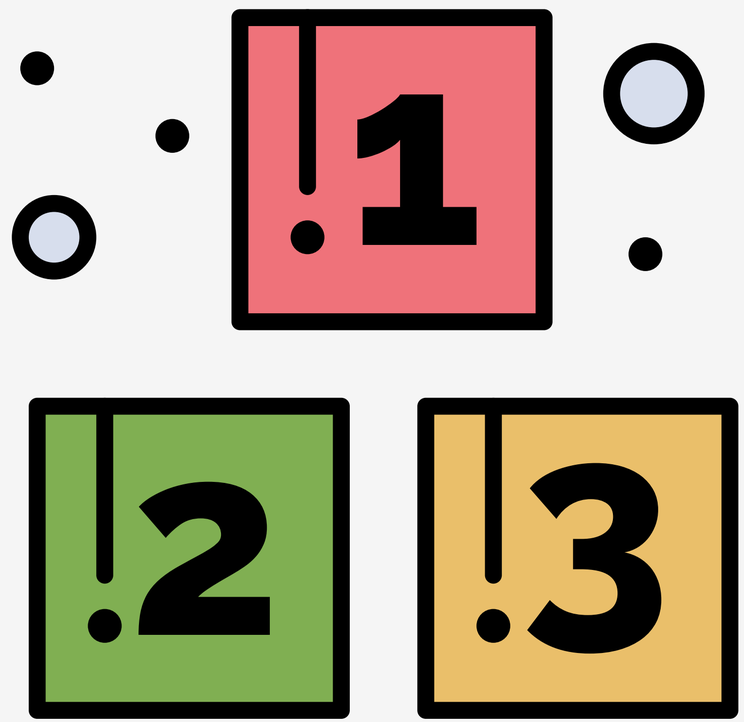
PSEUDO CLASSES

```css
/* matches every other row */
tr:nth-child(even) {
  background-color: #e7e7e7;
}

/* selects every 4th box starting with the first one */
.box:nth-child(4n + 1) {
  background-color: cadetblue;
}
```

HANDS-ON

CSS PSEUDO-ELEMENTS

::BEFORE AND ::AFTER



- Used to insert new content into an existing HTML element
- The `::before` element becomes the first child
- The `::after` element becomes the last child
- The `content` property specifies what content to insert - *REQUIRED*

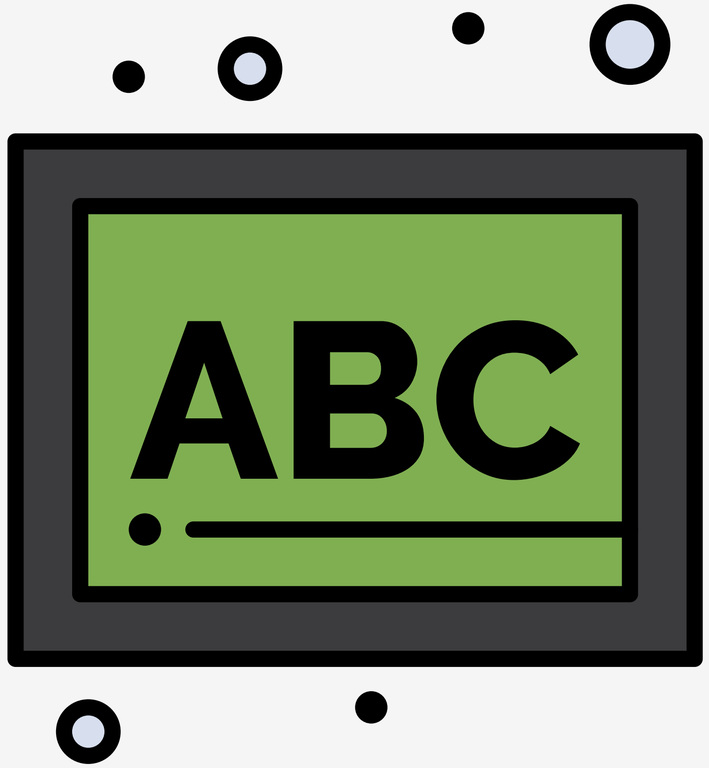
PSEUDO CLASSES

```
blockquote p::before {
  content: '\201C'; /* left quote */
}

blockquote p::after {
  content: '\201D'; /* right quote */
}
```

HANDS-ON

CSS CUSTOM PROPERTIES

CUSTOM PROPERTIES



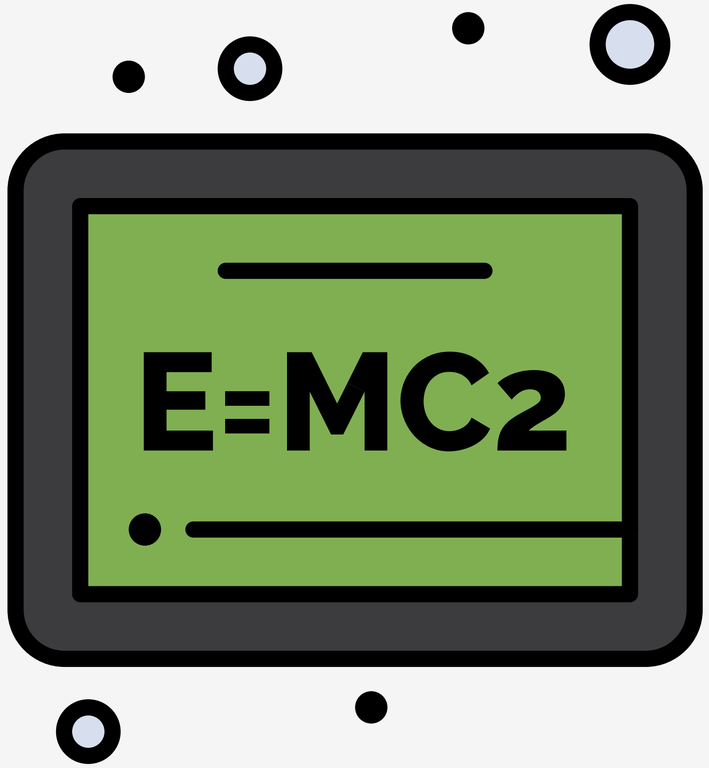
- Custom properties (CSS variables) can be used to store values to be referenced later
- Can be used to store colors, font families, breakpoints, etc
- Custom properties begin with `--`
- Custom properties are accessed using the `var()` function
- Custom properties are scooped to the elements in which they are declared

CUSTOM PROPERTIES

```
:root {
  --primary-color: #970cf9;
  --secondary-color: #ededе2;
  --serif: Georgia, 'Times New Roman', Times, serif;
  --sans-serif: Arial, Helvetica, sans-serif;
}

p { font-family: var(--serif); }

h2 { font-family: var(--sans-serif); }

.about {
  background-color: var(--primary-color);
  color: var(--secondary-color);
}
```

CSS CALC FUNCTION

CALC FUNCTION



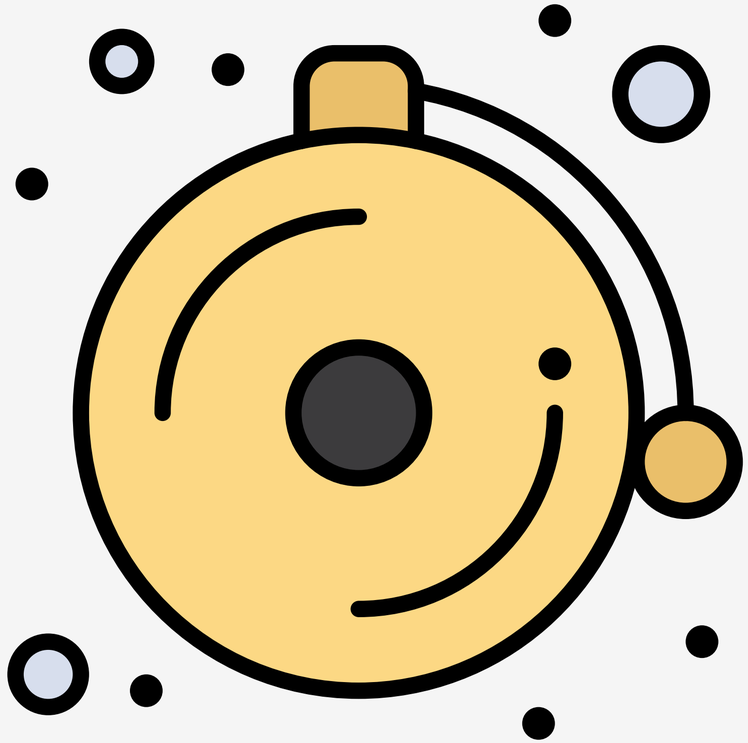
- The `calc()` function performs calculations when specifying CSS property values
- The `calc()` function takes a single expression and returns the resulting value
- The `calc()` function can be used to calculate length using different units

CALC FUNCTION

```css
.parent {
  display: flex;
  flex-wrap: wrap;
}

.child {
  width: calc(50% - 6px);
  border: 3px solid white;
}
```

HANDS-ON

NEXT TIME...



- Web Design Patterns