Advanced CSS

Slava Kim
CSS is simple! But is it?

You have some boxes on the screen

Boxes have text

Use CSS to change colors and move boxes slightly away from each other

... are we done yet?
Web in 1994

Most of you were not born yet

I was in my mom’s belly

HTML was only used to decorate academic papers published online

CSS is created to bring more richness to the web

Sometimes gifs of dancing babies inserted, too (jk that’s in 1996)
Web today
Web today - complicated layouts
Web today - pseudo 3D
Web today - animation

https://tympanus.net/Development/ModalWindowEffects/
Web today - Mobile Responsive Design
Today’s topics

Box model

Layouts ⇥ this one is big

Animations

Responsive Layouts
Box Model
Box model

```html
<div id="my-div">
    Hello World
</div>
```

```css
#my-div {
    margin: 2em;
    padding: 1em;
    border: solid black 1px;
    background: #eee;
    width: 250px;
}
```
Box model

```html
<div id="my-div">
  Hello World
</div>
```

```css
#my-div {
  margin: 2em;
  padding: 1em;
  border: solid black 1px;
  background: #eee;
  width: 250px;
}
```

https://codepen.io/anon/pen/rpKjPK
Box model

```html
<div id="my-div">
  Hello World
</div>
```

```css
#my-div {
  margin: 2em;
  padding: 1em;
  border: solid black 1px;
  background: #eee;
  width: 250px;
}
```

https://codepen.io/anon/pen/rpKjPK
Box model

```html
<div id="my-div">
  Hello World
</div>
```

```css
#my-div {
  margin: 2em;
  padding: 1em;
  border: solid black 1px;
  background: #eee;
  width: 250px;
}
```

https://codepen.io/anon/pen/rpKjPK,
Who will win?

1st grade math

1 CSS boi
box-sizing

```html
<div id="my-div">
  <div id="part-1">Part 1</div><div id="part-2">Part 2</div>
</div>
```

```css
#my-div {
  margin: 2em;
  background: #ccc;
  width: 300px;
  height: 50px;
}

#part-1, #part-2 {
  display: inline-block;
  width: 150px;
  margin: 0;
  padding: 1px;
}
```

https://codepen.io/anon/pen/goKmGW
Solution 1: remove padding

https://codepen.io/anon/pen/aEKJLe

```html
<div id="my-div">
  <div id="part-1">Part 1</div>
  <div id="part-2">Part 2</div>
</div>
```

```css
#my-div {
  margin: 2em;
  background: #ccc;
  width: 300px;
  height: 50px;
}

#part-1, #part-2 {
  display: inline-block;
  width: 150px;
  margin: 0;
  padding: 0px;
}
```
Solution 2: use box-sizing: border-box

https://codepen.io/anon/pen/LerWeV
Layouts
Layouts basics in CSS

http://learnlayout.com/
Layouts basics in CSS

http://learnlayoutout.com/
Layouts basics in CSS

http://learnlayout.com/
Display property

**block**

`<div>`

div is the standard block-level element. A block-level element starts on a new line and stretches out to the left and right as far as it can. Other common block-level elements are `p` and `form`, and new in HTML5 are `header`, `footer`, `section`, and more.

**inline**

`<span>`

span is the standard inline element. An inline element can wrap some text inside a paragraph `<span>` like this `</span>` without disrupting the flow of that paragraph. The `a` element is the most common inline element, since you use them for links.
Display property

Block - big block that stretches a lot, always has a new line

Inline - an element that is part of text, size is always proportional to text

Inline-block - stays within text, but can have its own size independent from text

None - hidden from the page

Flex - flexbox, special stuff

Grid - grid, special stuff

http://learnlayout.com/display.html
position: fixed

Always at the certain place relative to screen

Positioned with left, right, top, bottom

https://codepen.io/anon/pen/xpzqvZ
position: relative

Position the element relative to where it would be normally.

Positioned with left, right, top, bottom
position: absolute

Position the element absolutely relative to another element’s position.

Finds first ancestor that’s positioned non-statically.

Positioned with left, right, top, bottom
You can mark an element to “float” to a side

Usually is useful for images to be embedded into the text

Not great for layouts, tends to break a lot of things, needs a “clear”, etc

http://learnlayout.com/float.html
Common task: center DIV

There is a trick that a lot of people use:

Make left/right margin equal to “auto”

This will center the block element

https://codepen.io/anon/pen/jYKRwZ
Flexbox

A fairly new standard that is now available in most browsers

Simplifies a lot of layout tasks

Delegates a lot of layout computation to the browser as the DOM changes

https://css-tricks.com/snippets/css/a-guide-to-flexbox/

https://codepen.io/anon/pen/QaxPpO
Three things:

| Thing       | Thing another | Thing |

Many things:

| Thing       | Thing another | Another one | Woohoo | Things getting long | Thing |

Many many things:

| Thing       | Thing another | Another one | Woohoo | Things getting long | Thing | Thing | Thing |
Flexbox

You can let boxes grow more, or less, or not even grow at all.

The size will be recalculated as the containing div changes its size.

https://codepen.io/anon/pen/QaxPpO
<table>
<thead>
<tr>
<th>small</th>
<th>bigger</th>
<th>biggest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gets bigger</td>
<td>Gets even bigger</td>
<td>Stays the same</td>
</tr>
<tr>
<td>Gets bigger</td>
<td>Gets even bigger</td>
<td>Stays the same</td>
</tr>
<tr>
<td>Gets bigger</td>
<td>Gets even bigger</td>
<td>Stays the same</td>
</tr>
</tbody>
</table>
https://css-tricks.com/snippets/css/a-guide-to-flexbox/
Animations in CSS

Animate.css

Just-add-water CSS animations
Libraries for CSS Animations

Animate.CSS

Cssanimation.io

Hover.css

etc
Simple Transition

https://codepen.io/anon/pen/GyBjzW
Simple Transition

```html
<div id="container">
  <div class="appeared">comment</div>
</div>
<button id="add">
  add another comment
</button>

#container div {
  opacity: 0.4;
}

#container div.appeared {
  opacity: 1.0;
  transition: opacity 3s;
}
```
Simple Transition

Append a new element to DOM

Trigger the reflow

Add the transition class

Without reflow, browser will batch operations together and animation will omitted.

document.getElementById('add').addEventListener('click', function () {
    // create a new element
    var el = document.createElement('div');
    el.innerText = "new comment";

    // add it to the container
    document.getElementById('container').
        appendChild(el);

    // force browser to display it (force reflow)
    el.getBoundingClientRect();

    // add the class for animation
    el.className = "appeared";
});
CSS Animation - transition

The basic idea:

Define the start state

Define the end state

Define the transitions in between in “keyframes”

Or let the browser automatically decide how to transition

https://codepen.io/anon/pen/YYjpzN
CSS Animation - transition

```css
#comment {
  padding: 5px;
  border: solid black 1px;
  width: 100px;
  height: 100px;
  font-size: 20px;

  /* animated properties */
  background-color: red;
  color: yellow;
  left: 0px;
  top: 0px;
  position: absolute;

  /* transition details */
  transition-property: width height background-color font-size left top transform -webkit-transform color;
  transition-duration: 2s;
  transition-timing-function: ease-in-out;
}
```
CSS Animation - transition

```css
#comment {
  padding: 5px;
  border: solid black 1px;
  width: 100px;
  height: 100px;
  font-size: 20px;

  /* animated properties */
  background-color: red;
  color: yellow;
  left: 0px;
  top: 0px;
  position: absolute;

  /* transition details */
  transition-property: width height background-color font-size left top transform -webkit-transform color;
  transition-duration: 2s;
  transition-timing-function: ease-in-out;
}
```

```css
#comment.final {
  transform: rotate(270deg);
  width: 50px;
  height: 50px;
  background-color: blue;
  left: 150px;
  top: 25px;
  position: absolute;
  transition-property: width height background-color font-size left top transform color;
  transition-duration: 2s;
  transition-timing-function: ease-in-out;
}
```
CSS Animation - transition

```css
#comment {
  padding: 5px;
  border: solid black 1px;
  width: 100px;
  height: 100px;
  font-size: 20px;

  /* animated properties */
  background-color: red;
  color: yellow;
  left: 0px;
  top: 0px;
  position: absolute;

  /* transition details */
  transition-property: width height background-color font-size left top transform color;
  transition-duration: 2s;
  transition-timing-function: ease-in-out;
}
#comment.final {
  transform: rotate(270deg);
  width: 50px;
  height: 50px;
  background-color: blue;
  left: 150px;
  top: 25px;
  position: absolute;
  transition-property: width height background-color font-size left top transform color;
  transition-duration: 2s;
  transition-timing-function: ease-in-out;
```
CSS Animation - keyframes

```css
@keyframes identifier {
  0% { top: 0; left: 0; }
  30% { top: 50px; }
  68%, 72% { left: 50px; }
  100% { top: 100px; left: 100%; }
}
```
More about CSS Animations


Responsive Layouts
Media queries

Media queries allow you to run certain CSS only if:

- The device screen is at least certain size
- The device supports hovering (with a mouse)
- The device supports colors
- ... Speech input, aspect ratio, etc
Media queries in action

```html
1  <div id="container">
2    This is the container
3  </div>
```

```css
/*container */

/*container */

@media (max-width: 1000px) {
  #container {
    background: green;
  }
}

@media (max-width: 800px) {
  #container {
    background: red;
  }
}
```

https://codepen.io/anon/pen/wpxoPY
Responsive layout example

https://codepen.io/anon/pen/PEBbeB
Responsive layout example

On smaller screen sizes turn the columns into sections by changing the Flexbox direction from “row” to “column”.

You can imagine hiding certain columns, reordering them, separating unimportant content, etc.

https://codepen.io/anon/pen/PEBbeB
Live Example

http://jadlimcaco.com/

http://zbt.mit.edu/
Debugger in Chrome

Chrome has a special button to simulate different screen sizes, aspect ratios and presets of devices

Open devtools

Click on the small device icon
Recap
Recap

- **padding**
- **margin**
- **border**

**Hello World**
Recap

**HTML**

```html
<div id="my-div">
  <div id="part-1">Part 1</div>
  <div id="part-2">Part 2</div>
</div>
```

**CSS**

```css
#my-div {
  margin: 2em;
  background: #ccc;
  width: 300px;
  height: 50px;
}
#part-1, #part-2 {
  display: inline-block;
  width: 150px;
  margin: 1px;
  padding: 1px;
  box-sizing: border-box;
}
```
Recap

http://learnlayout.com/
Recap

https://codepen.io/anon/pen/GyBjzW
Media queries in action

```html
data-clipboard-attribute
1. <div id="container">
2.   This is the container
3. </div>
```

```css
#container {
  background: blue;
  color: white;
  padding: 1em;
}

@media (max-width: 1000px) {
  #container {
    background: green;
  }
}

@media (max-width: 800px) {
  #container {
    background: red;
  }
}
```

https://codepen.io/anon/pen/wpxoPY