

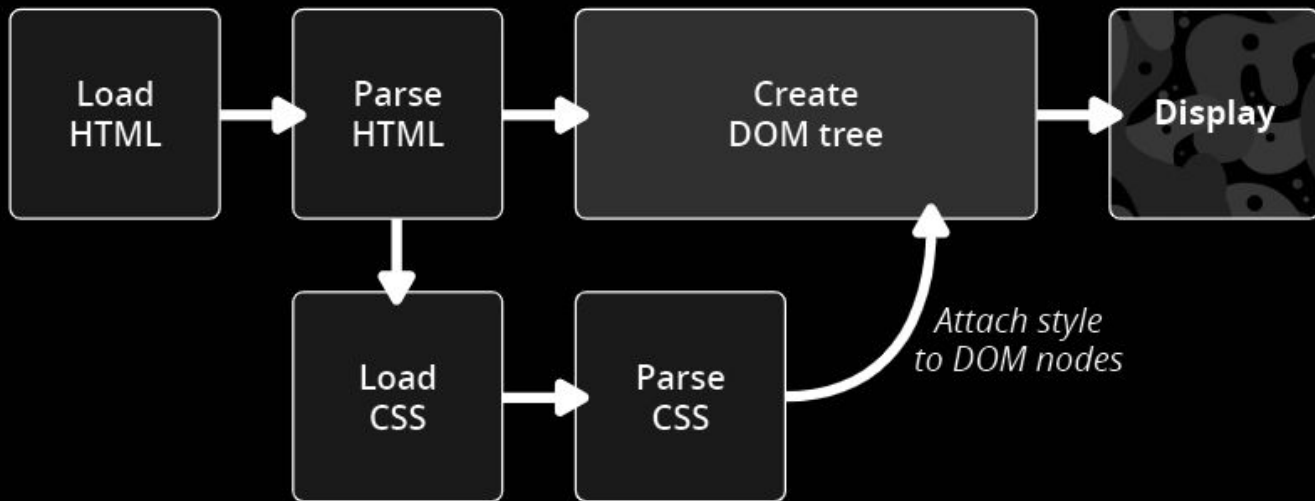
{ CSS }

Cascading Style Sheets

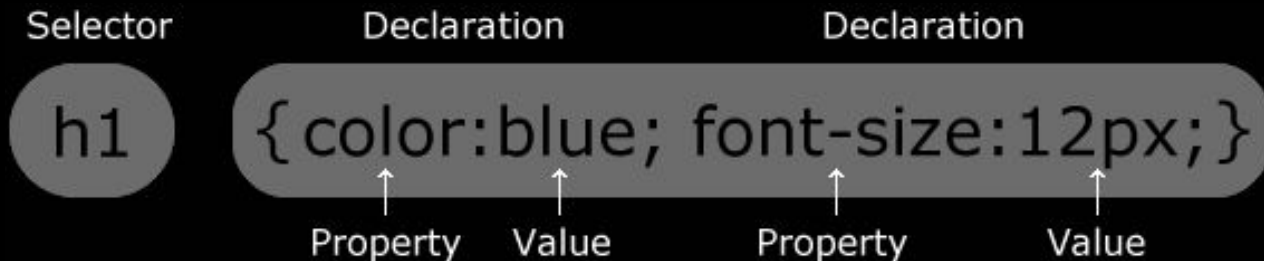
language to specify how documents are **presented**

csszengarden.com

How it works?



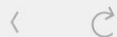
Syntax



Inline styles

index.html

```
1 <p style="color: red;">I'm learning to code!</p>
```



https://localhost/



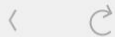
I'm learning to code!

<style> tag

index.html

style.css

```
1 <head>
2 <style>
3   p {
4     color: red;
5     font-size: 20px;
6   }
7 </style>
8 </head>
9 <body>
10 <p>I'm bigger and red too.</p>
11 </body>
```

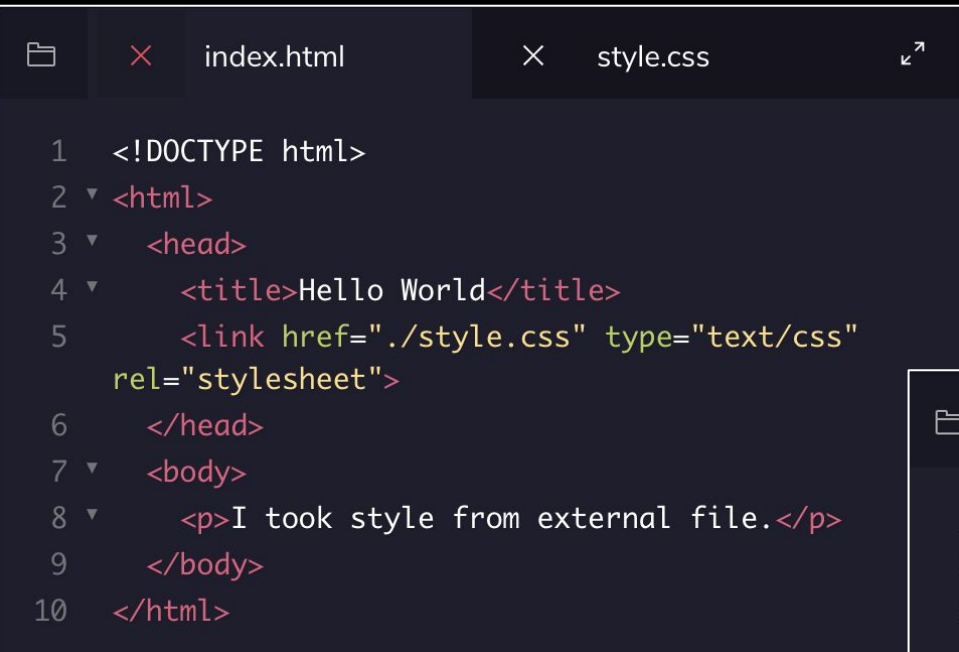


https://localhost/



I'm bigger and red too.

The .css file



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Hello World</title>
5 <link href="./style.css" type="text/css"
  rel="stylesheet">
6 </head>
7 <body>
8 <p>I took style from external file.</p>
9 </body>
10 </html>
```



I took style from external file.



```
1 p {
2   font-family: Arial;
3   color: red;
4 }
```


Tag name

```
p {  
  
}
```

Class name

```
<p class="brand">Sole Shoe Company</p>
```

```
<h1 class="green bold"> ... </h1>
```

```
.green {  
  color: green;  
}  
  
.bold {  
  font-weight: bold;  
}
```

ID name

```
<h1 id="large-title"> ... </h1>
```

```
#large-title {
```

IDs and Classes

- IDs **override** the styles of tags and classes.
- IDs are **UNIQUE**, but classes not
- There are no browser defaults for any ID or Class
- <http://yourdomain.com#comments>
- Elements can have BOTH

Chaining selectors

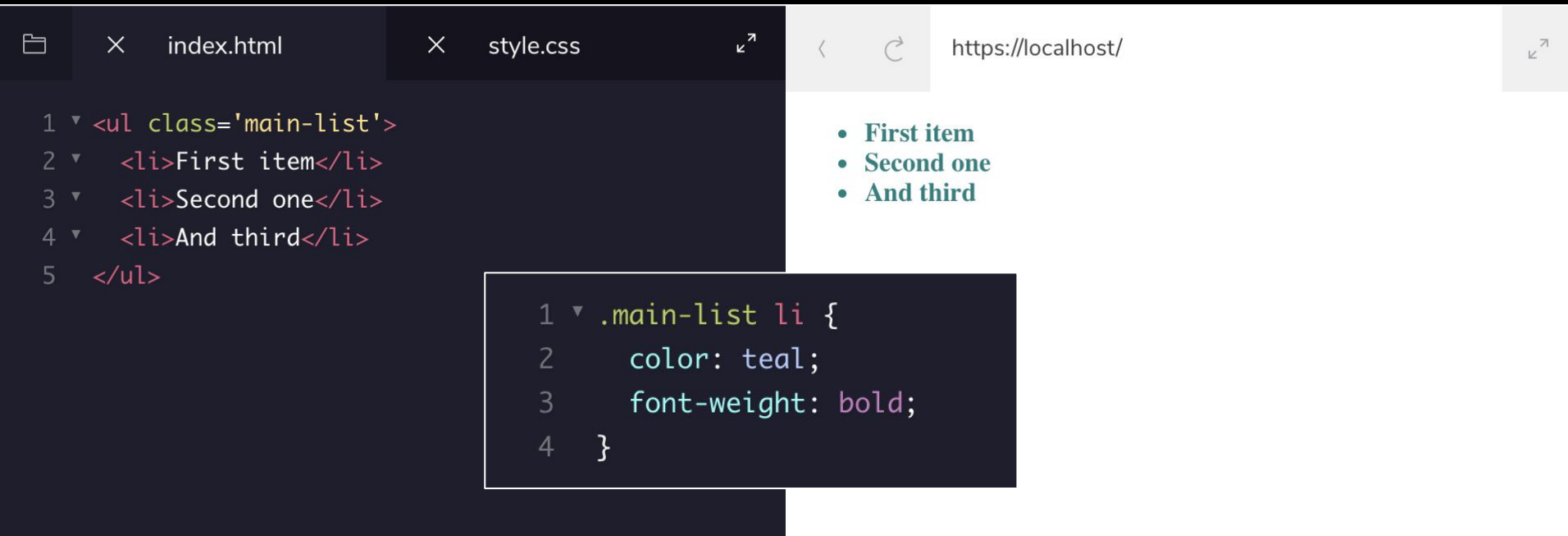
```
index.html style.css ↗
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="UTF-8" />
5   <title>My web page title</title>
6   <link href="./style.css" type="text/css"
    rel="stylesheet" />
7 </head>
8 <body>
9   <h1 class="special">My header</h1>
10 </body>
11 </html>
```



My header

```
h1.special {
}
```

Nested elements



The screenshot shows a web browser interface with two tabs: 'index.html' and 'style.css'. The 'index.html' tab is active, displaying the following HTML code:

```
1 <ul class='main-list'>
2   <li>First item</li>
3   <li>Second one</li>
4   <li>And third</li>
5 </ul>
```

The 'style.css' tab is also visible, showing the following CSS code:

```
1 .main-list li {
2   color: teal;
3   font-weight: bold;
4 }
```

The browser's address bar shows the URL 'https://localhost/'. The rendered page displays a bulleted list with three items, all in teal and bold font:

- First item
- Second one
- And third

Specificity

Specificity is the order by which the browser decides which CSS styles will be displayed.

```
<h1 class="headline">Breaking News</h1>
```

```
h1 {  
  color: red;  
}  
  
.headline {  
  color: firebrick;  
}
```

What about IDs?

Chaining and specificity

Adding more than one tag, class, or ID to a CSS selector increases the specificity of the CSS selector.

The screenshot shows a web browser window with the address bar displaying `https://localhost/`. The browser's developer tools are open, showing the HTML and CSS. The HTML pane on the left displays the following code:

```
1 <body>
2   <span class="description">
3     <h5>My header</h5>
4   </span>
5 </body>
```

The CSS pane on the right shows the following styles:

```
1 .description h5 {
2   color: red;
3 }
4
5 h5 {
6   color: rebeccapurple;
7 }
```

The browser's rendered output shows the text "My header" in red, demonstrating that the chained selector `.description h5` has higher specificity than the simple `h5` selector.

!Important

The image shows a web browser window with a dark-themed code editor on the left and a light-themed browser window on the right. The code editor has two tabs: 'index.html' and 'style.css'. The 'index.html' tab is active, showing the following HTML code:

```
1 <body>
2   <span class="description">
3     <h5>My header</h5>
4   </span>
5 </body>
```

The 'style.css' tab is also visible, showing the following CSS code:

```
1 .description h5 {
2   color: red;
3 }
4
5 h5 {
6   color: rebeccapurple!important;
7 }
```

The browser window shows the URL 'https://localhost/' and the rendered page content 'My header' in a purple color, demonstrating the effect of the 'rebeccapurple!important' rule.

Multiple selectors

```
h1 {  
  font-family: Georgia;  
}  
  
.menu {  
  font-family: Georgia;  
}
```

=

```
h1,  
.menu {  
  font-family: Georgia;  
}
```

Basic properties

color: white, #FFF, rgb(255, 255, 255)

opacity: 0.5

background-color: black

background-image: url("../image.png")

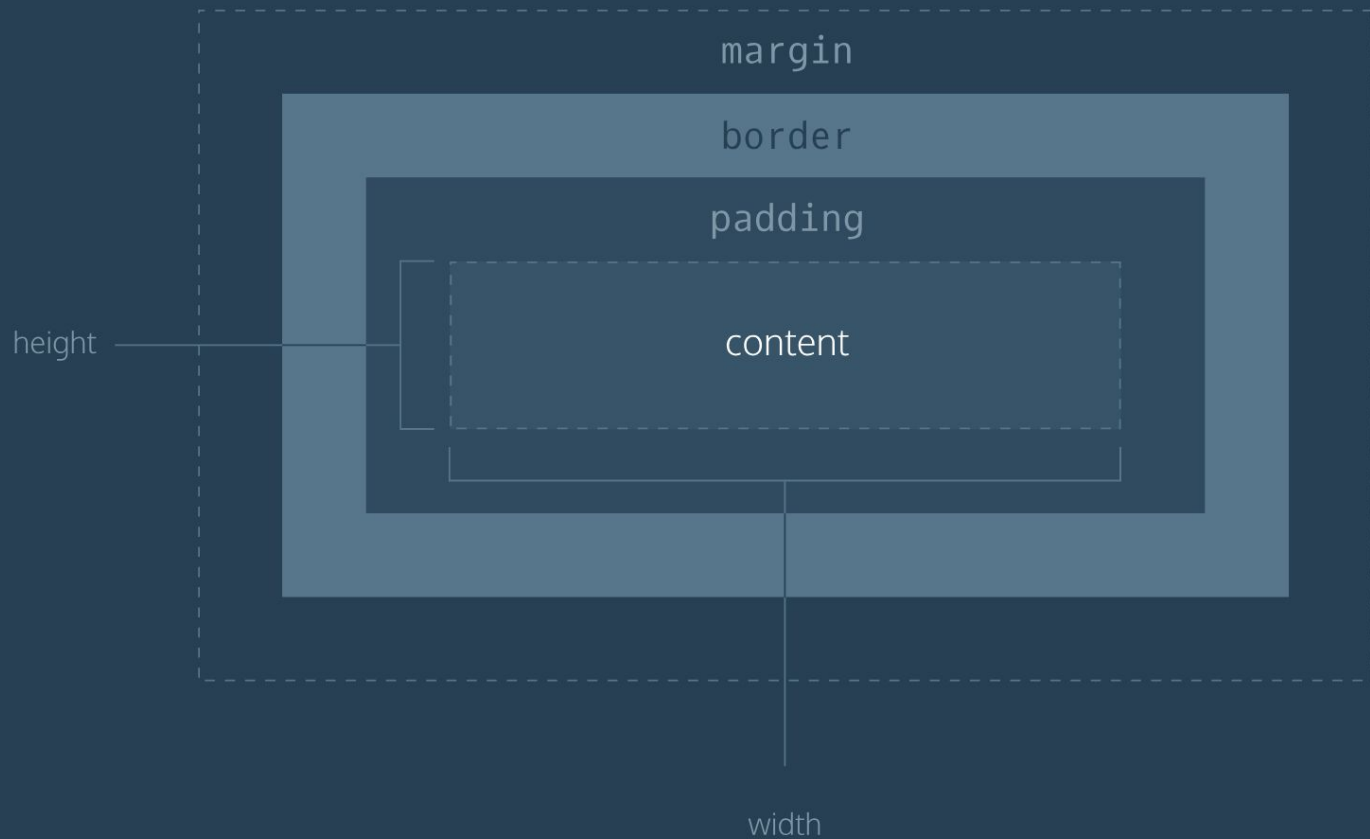
font-family: Times New Roman

font-weight: bold, normal

font-size: 18px, 2em, 3rem

text-align: right, center, left

Box model



Box model

✕ style.css

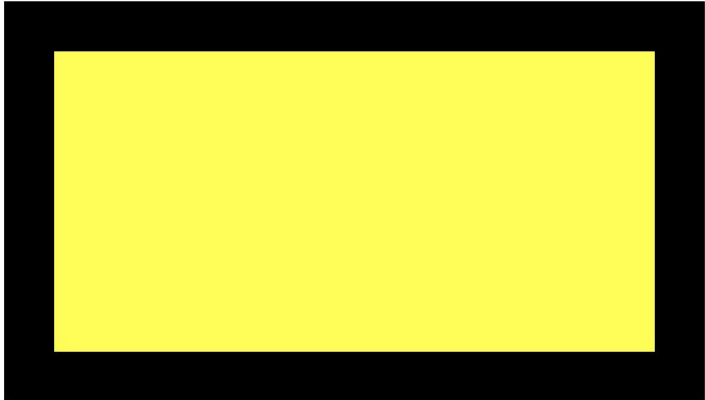
✕ index.html ↗

< ↻ https://localhost/ ↗ ↶

```
1 ▾ div {  
2   background-color: yellow;  
3   width: 250px;  
4   height: 100px;  
5   border: 25px solid black;  
6   padding: 25px;  
7   margin: 50px;  
8 }
```

padding: 6px 11px 4px 9px;

margin: 0 auto;



Box model

```
p {  
  min-width: 300px;  
  max-width: 600px;  
}
```

The width of all paragraphs will not shrink below 300 pixels, nor will the width exceed 600 pixels.

Overflow

Lorem ipsum dolor amet
next level banh mi actually
etsy craft beer. Portland
meh palo santo pitchfork
wayfarers raclette kinfolk
try-hard YOLO. Lo-fi cred
pork belly, cloud bread
artisan heirloom raw
denim kombucha. Godard
etsy ugh, letterpress roof

party fingerstache
succulents edison bulb.
Iceland disrupt palo santo
fixie hella taiyaki celiac
green juice

visible

Lorem ipsum dolor amet
next level banh mi actually
etsy craft beer. Portland
meh palo santo pitchfork
wayfarers raclette kinfolk
try-hard YOLO. Lo-fi cred
pork belly, cloud bread
artisan heirloom raw
denim kombucha. Godard
etsy ugh, letterpress roof

hidden

Lorem ipsum dolor amet
next level banh mi
actually etsy craft beer.
Portland meh palo santo
pitchfork wayfarers
raclette kinfolk try-hard
YOLO. Lo-fi cred pork
belly, cloud bread
artisan heirloom raw

scroll

Lorem ipsum dolor amet
next level banh mi
actually etsy craft beer.
Portland meh palo santo
pitchfork wayfarers
raclette kinfolk try-hard
YOLO. Lo-fi cred pork
belly, cloud bread
artisan heirloom raw
denim kombucha.

overflow-y

Pseudo-classes

`:link`

Selects all unvisited links

`:hover`

Selects elements on mouse hover

`:visited`

Selects all visited links

`:active`

Selects an element whilst it is being activated by the user, for example, when the user is mid-click

`:focus`

Selects elements (typically form elements) that have been focused on via a click or keyboard event

Pseudo-elements

```
div:before {  
  content: "";  
}  
  
div:after {  
  content: "";  
}
```

The **:before** and **:after** pseudo-elements allow you to insert content before or after any HTML element that isn't self closing (like `` and `<input>`).

The content property is required but can be left blank.

These pseudo-elements can be treated and styled like any other element.



Combinators

*

The Universal Selector - select all elements

h1

p

p

ul

p

h1 + p

The Adjacent Sibling Combinator - select all <p> elements that immediately follow a <h1>

h1

p

p

ul

p

h1 ~ p

The General Sibling Combinator - select all <p> elements that follow (and are siblings of) a <h1>

h1

p

p

ul

p

.list > li

The Child Combinator - select all elements that are direct children of .list

```
<ul class="list">
  <li>One ✓
  <li>Two ✓
  <ul>
    <li>Sub Item ✗
```

Categories of HTML elements

Display: inline

<code>display: inline</code>	<code>display:</code>
<code>inline</code>	<code>display: inline</code>

Display: block

<code>display: block</code>
<code>display: block</code>
<code>display: block</code>

Display: inline-block

<code>display: inline-block</code>	<code>display: inline-block</code>	<code>display: inline-block</code>
------------------------------------	------------------------------------	------------------------------------

Media queries



style.css



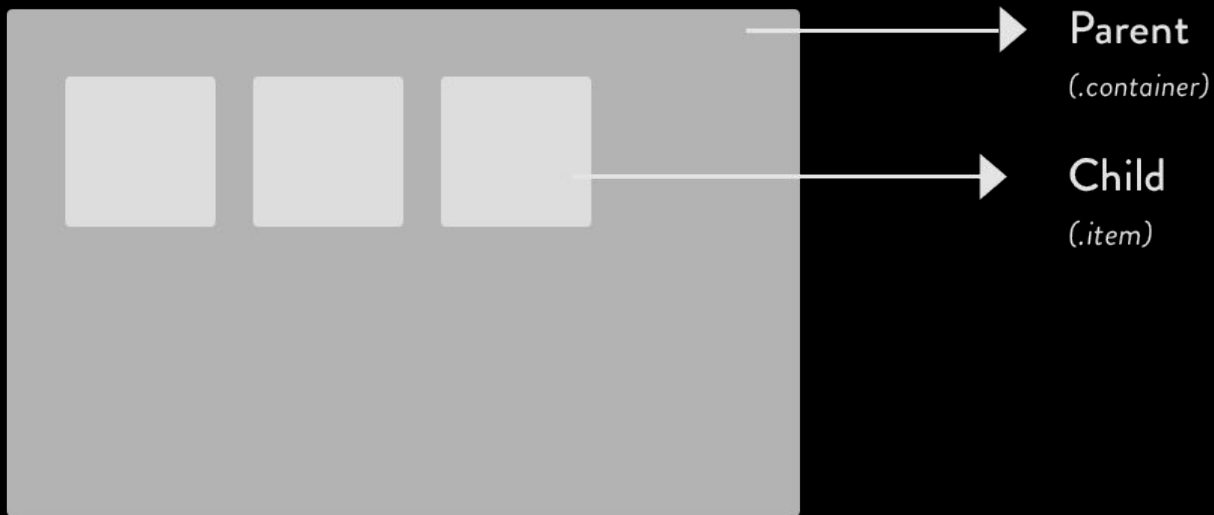
```
1  /* Large desktop */
2  ▼ @media (min-width: 1200px) {
3      /* CSS Classes */
4  }
5  /* Portrait tablet to landscape and desktop */
6  ▼ @media (min-width: 768px) and (max-width: 979px)
7      {
8      /* CSS Classes */
9  }
10 /* Landscape phone to portrait tablet */
11 ▼ @media (max-width: 767px) {
12     /* CSS Classes */
13 }
14 /* Landscape phones and down */
15 ▼ @media (max-width: 480px) {
16     /* CSS Classes */
17 }
```

Flexible Box Layout Module

Flexbox - makes easier to design flexible responsive layout structure



Flexbox



Flexbox

Flex container properties:

`flex-direction`

`flex-wrap`

`flex-flow`

`justify-content`

`align-items`

`align-content`

Flex item properties:

`order`

`flex-grow`

`flex-shrink`

`flex-basis`

`flex`

`align-self`

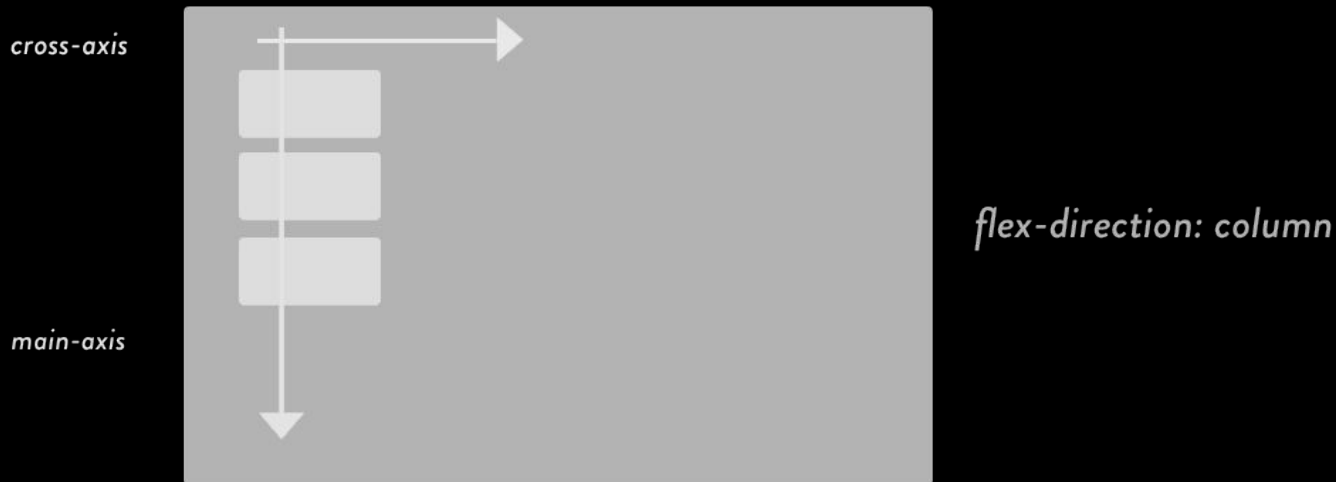
Flexbox

```
1 ▾ <div class="flex-container">
2 ▾   <div>1</div>
3 ▾   <div>2</div>
4 ▾   <div>3</div>
5   </div>
```

```
1 ▾ .flex-container {
2   display: flex;
3 }
```



Flexbox **flex-direction**



Flexbox **flex-direction**

```
1 ▾ <div class="flex-container">
2 ▾   <div>1</div>
3 ▾   <div>2</div>
4 ▾   <div>3</div>
5   </div>
```

```
1 ▾ .flex-container {
2   display: flex;
3   flex-direction: column;
4 }
```



flexboxfroggy.com

Workshop #1

First page

postcards list polishing

Workshop #2

Second page

postcards share form polishing