CSS

CSS FUNDAMENTALS

# Combinators

**IN A ROCKET**
Learn front-end development at *rocket speed*

**DESCENDANT**

Combinator

**CHILD**

**SIBLING**

# DESCENDANT COMBINATOR

Describe an element that is the descendant of another element in the document tree.

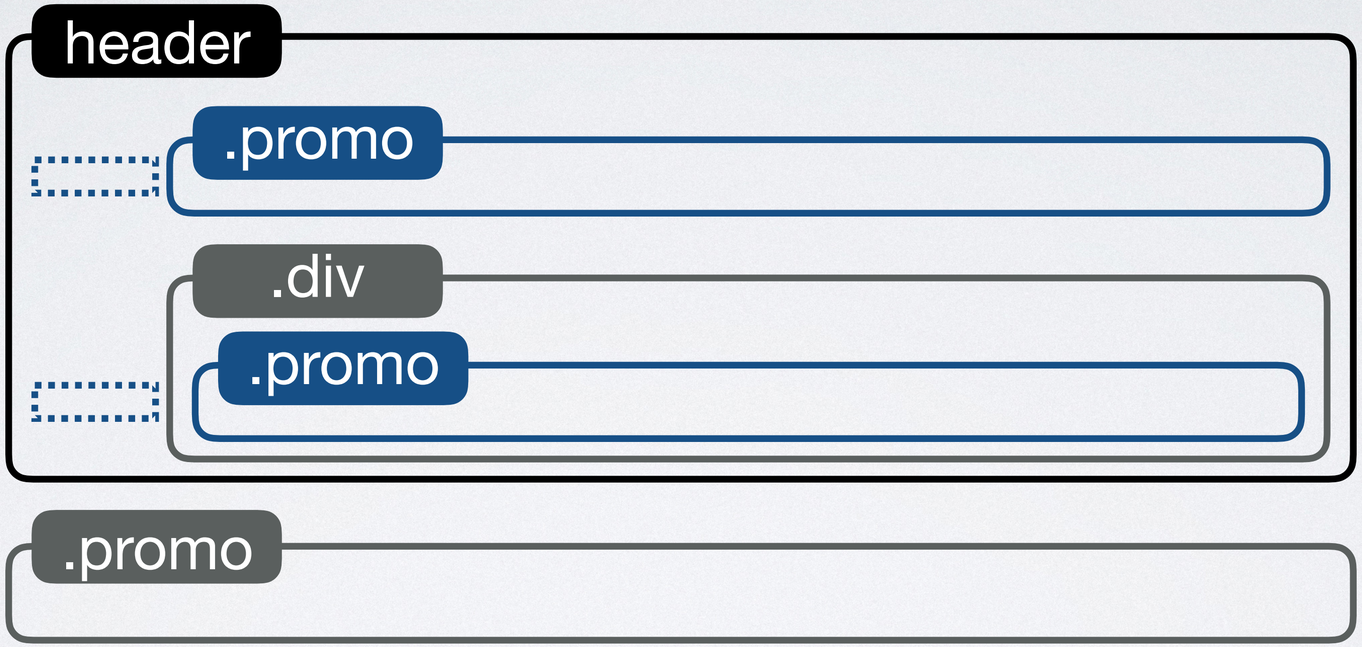
**Syntax** `selectorOutsite selectorInside {style properties}`

header .promo {color: green}

With this code all elements that have a class `promo` inside a `header` element are shown in green.

# DESCENDANT COMBINATOR

header .promo



# DESCENDANT COMBINATOR

READY TO USE CODE

## HTML

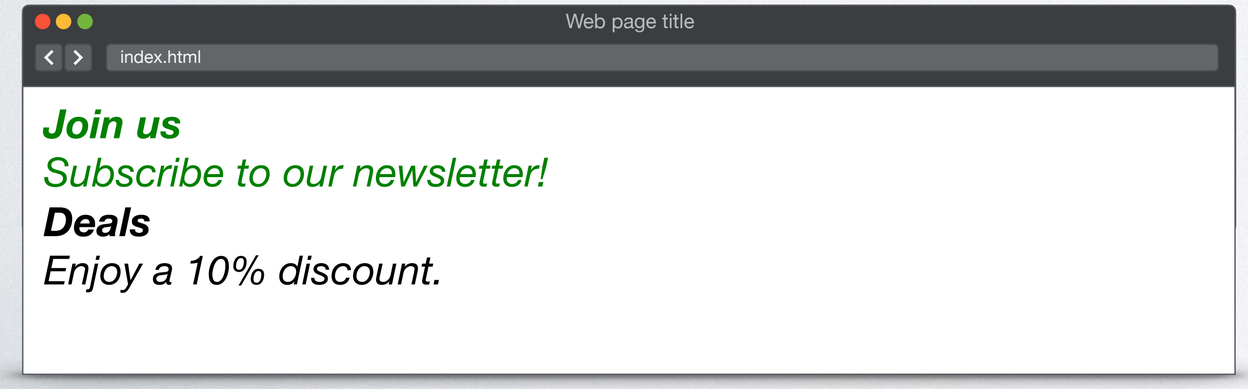
```
<body>
<header>
  <h1 class="promo">Join us</h1>
  <div>
    <p class="promo">Subscribe to our newsletter!</p>
  </div>
</header>
<h1 class="promo">Deals</h1>
<p class="promo">Enjoy a 10% discount.</p>
</body>
```

## CSS

```
.promo { font-style: italic; }

header .promo { color: green; }
```

## Browser



DESCENDANT

**CHILD**

Combinator

SIBLING

# CHILD COMBINATOR

Only targets immediate child elements.

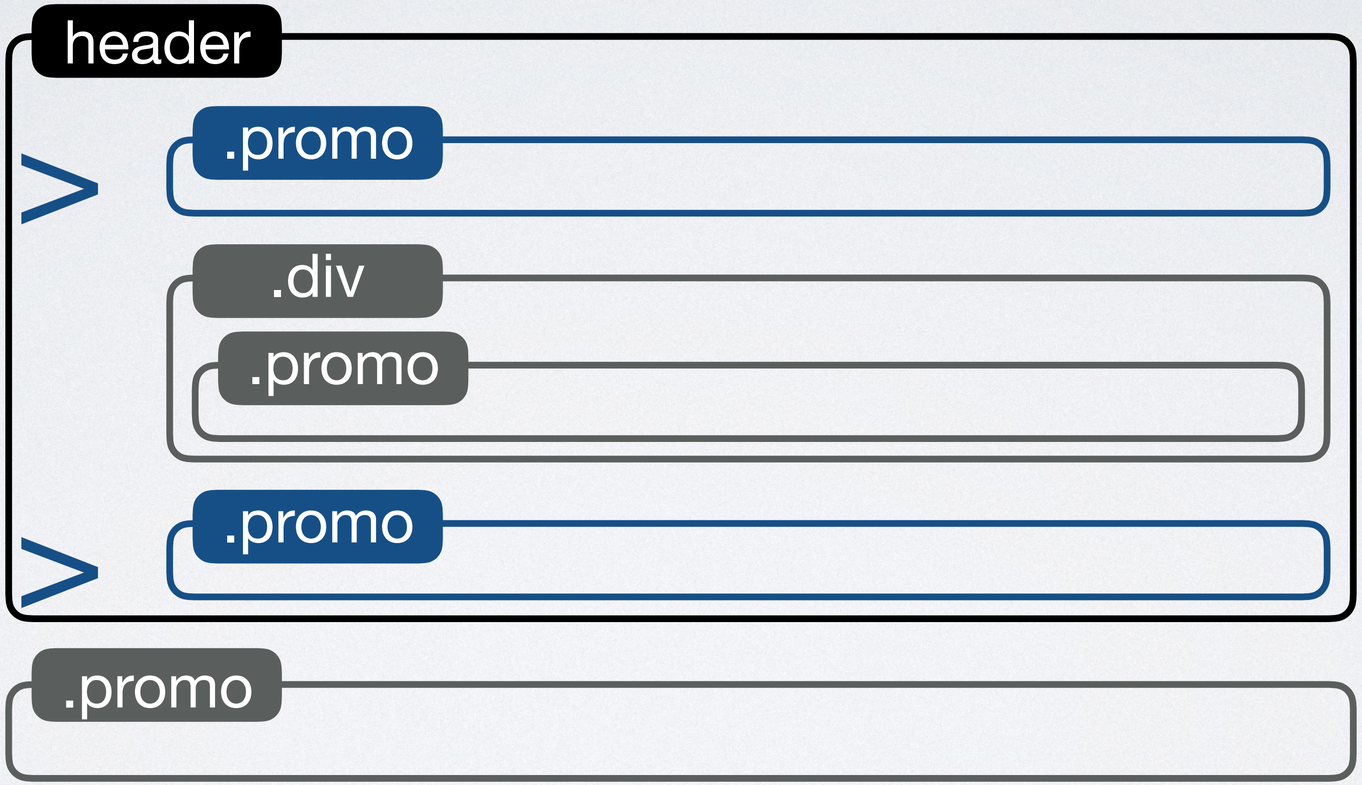
**Syntax** `selectorOutsite > selectorInside {style properties}`

```
header > .promo  {color: green}
```

With this code only the immediate child elements of `header` that have a class `promo` are shown in green.

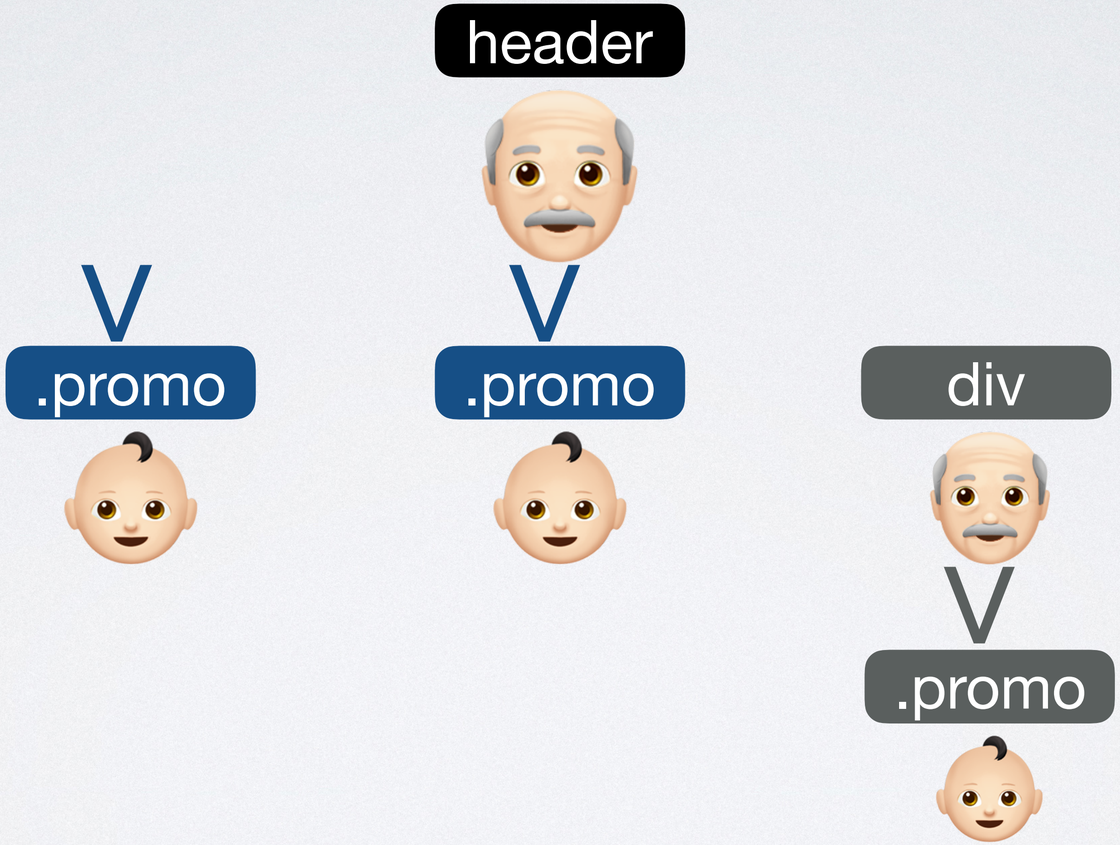
# CHILD COMBINATOR

## header > .promo



# CHILD COMBINATOR

## header > .promo



# CHILD COMBINATOR

READY TO USE CODE

## HTML

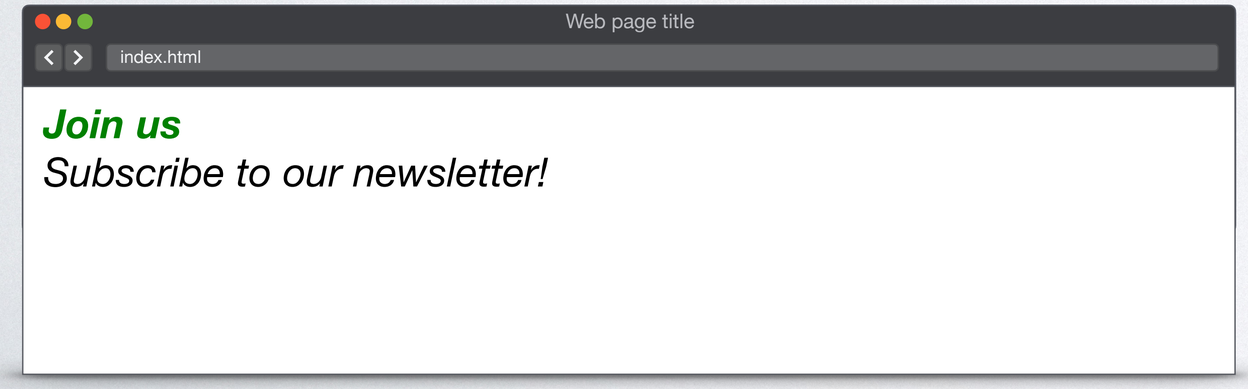
```
<body>
<header>
  <h1 class="promo">Join us</h1>
  <div>
    <p class="promo">Subscribe to our
    newsletter!</p>
  </div>
</header>
</body>
```

## CSS

```
.promo { font-style: italic; }

header > .promo { color: green; }
```

## Browser



DESCENDANT

CHILD

**SIBLING**

Combinator

## Sibling combinators

**General**

Adjacent

# SIBLING COMBINATOR: GENERAL

Select elements based of sibling relationships.

Syntax

```
selectorA ~ selectorB {style properties}
```

h2 ~ p {color: green}

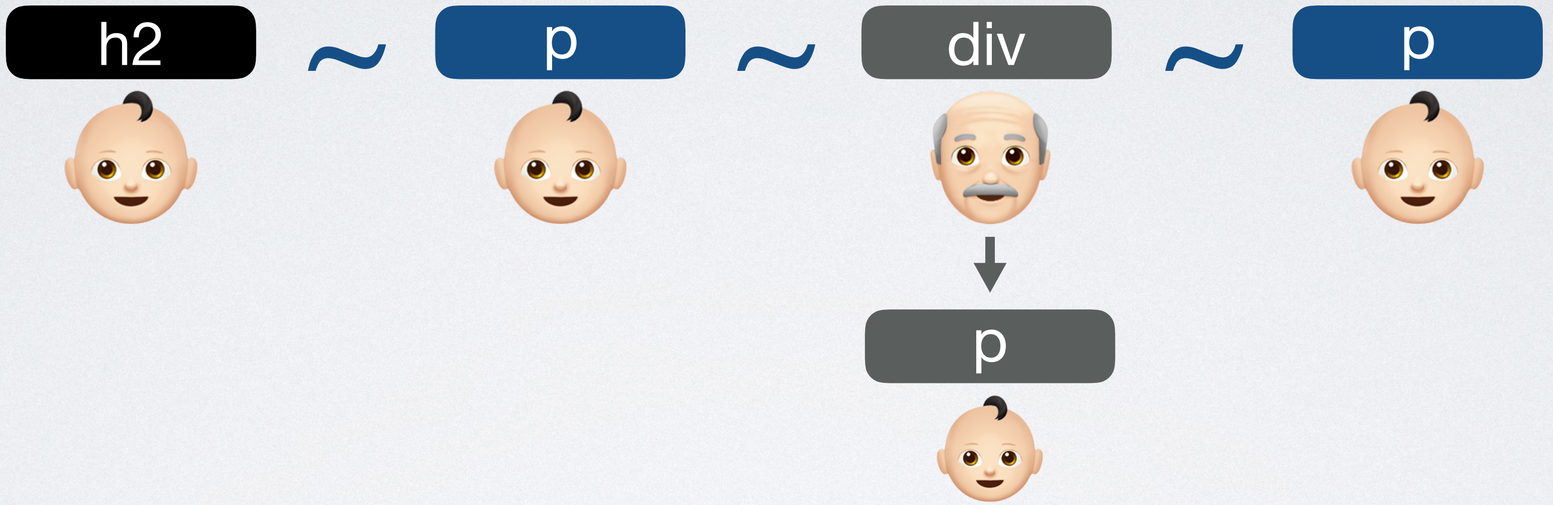
With this code only the paragraphs that are siblings of h2 are shown in green.

# SIBLING COMBINATOR: GENERAL

h2 ~ p

# SIBLING COMBINATOR: GENERAL

## h2 ~ p



# SIBLING COMBINATOR: GENERAL

READY TO USE CODE

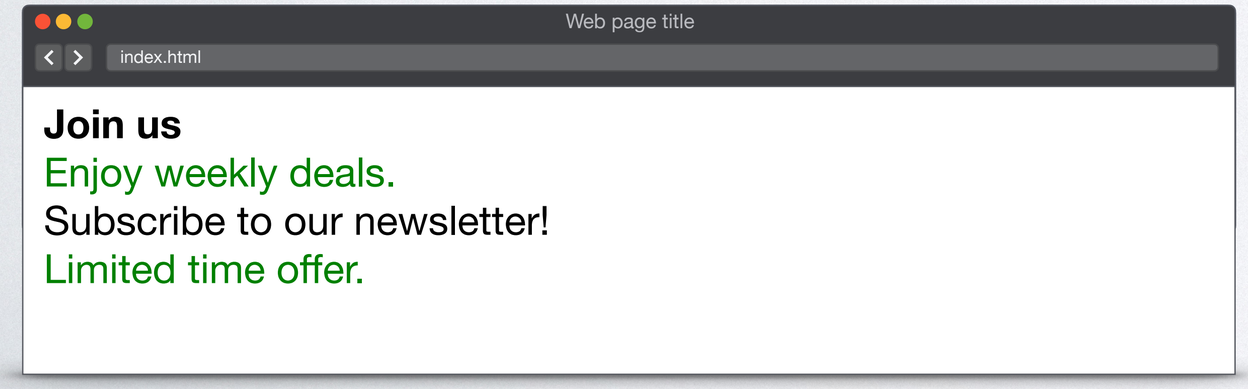
## HTML

```
<body>
<h2>Join us</h2>
<p>Enjoy weekly deals.</p>
<div>
  <p>Subscribe to our newsletter!</p>
</div>
<p>Limited time offer.</p>
</body>
```

## CSS

```
h2 ~ p { color: green; }
```

## Browser



## Sibling combinators

General

**Adjacent**

# SIBLING COMBINATOR: ADJACENT

Select elements that are immediate siblings, not just general.

Syntax

```
selectorA + selectorB {style properties}
```

h2 + p {color: green}

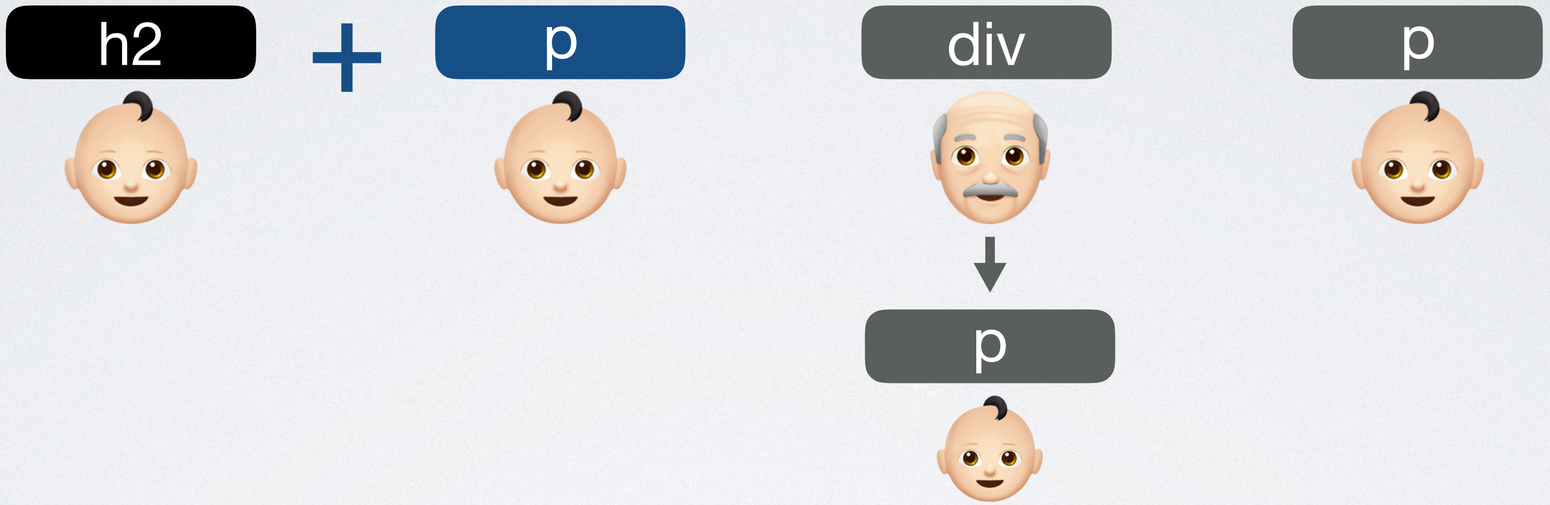
With this code only the paragraphs that are immediate siblings of h2 are shown in green.

# SIBLING COMBINATOR: ADJACENT

## h2 + p

# SIBLING COMBINATOR: ADJACENT

## h2 + p



# SIBLING COMBINATOR: ADJACENT

READY TO USE CODE

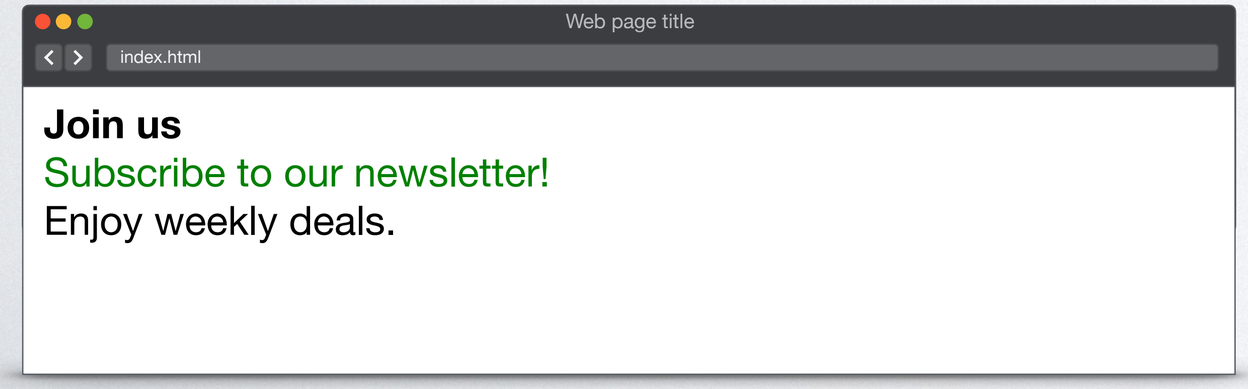
## HTML

```
<body>
<h2>Join us</h2>
<p>Subscribe to our newsletter!</p>
<p>Enjoy weekly deals.</p>
</body>
```

## CSS

```
h2 + p { color: green; }
```

## Browser



# SIBLING COMBINATOR: ADJACENT

READY TO USE CODE

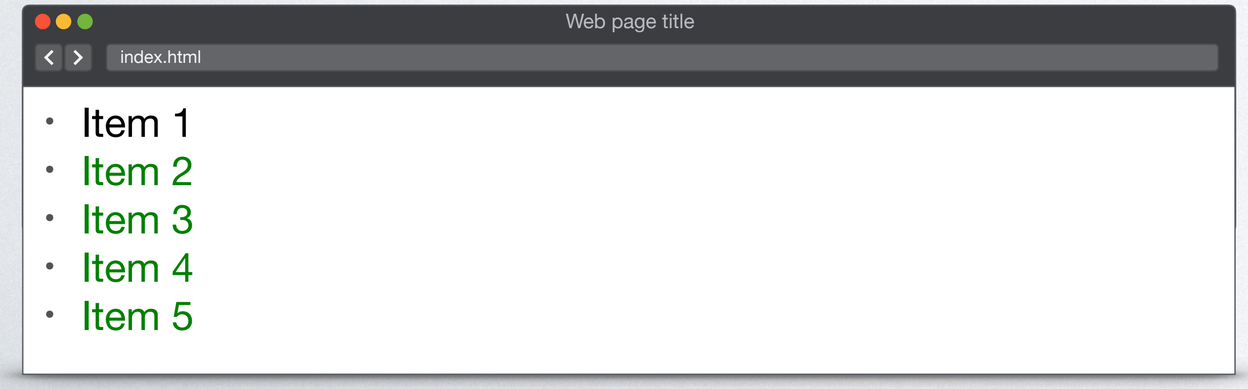
## HTML

```
<body>
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
  <li>Item 4</li>
  <li>Item 5</li>
</ul>
</body>
```

## CSS

```
li + li { color: green; }
```

## Browser



**DESCENDANT**
Combinator

**CHILD**
Combinator

**SIBLING**
Combinator

# AVOID EXTRA SELECTORS

```
body #container .myclass ul li {....}
```

```
.someclass li {...}
```

YOU CAN CONTINUE THIS COURSE FOR FREE ON

# inarocket.com

Learn front-end development at *rocket speed*

+ READY TO USE CODE

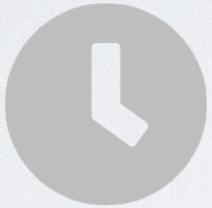
+ QUIZZES

+ FREE UPDATES

by miguelsanchez.com

# inarocket.com

Learn front-end development at *rocket speed*

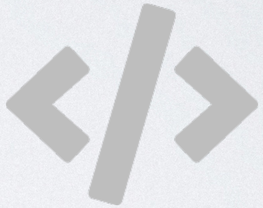


### We respect your time

No more blah blah videos. Just straight to the point slides with relevant information.

### Step by step guides

Clear and concise steps to build real use solutions. No missed points.



### Ready to use code

Real code you can just copy and paste into your real projects.

CSS

CSS FUNDAMENTALS

# Combinators

**IN A ROCKET**
Learn front-end development at *rocket speed*