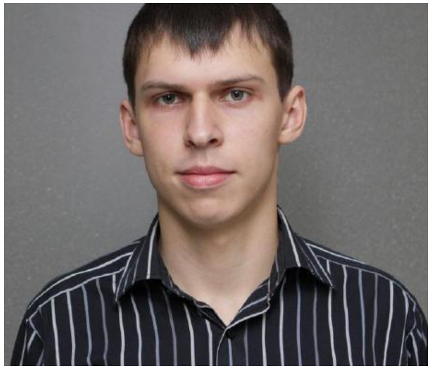
HTML & CSS

YURY RYBAK

MAY 17, 2019

iTechArt



Hi there. I’m Yury Rybak

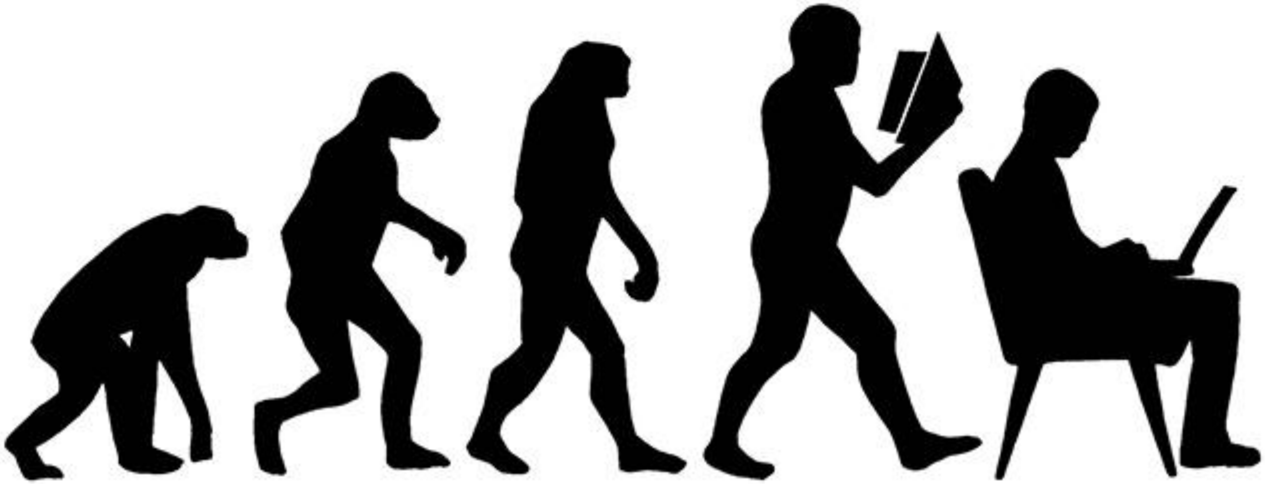
Department: D5
Technologies: Software Engineer

Contact Information:

Email: yury.rybak@itechart-group.com
Skype: live:yury.rybak.itechart

OUR GOALS

- Revise HTML & CSS knowledge;
- Learn basic and intermediate level of HTML & CSS;
- Find out what best and worst practices are;
- Apply skills practically.

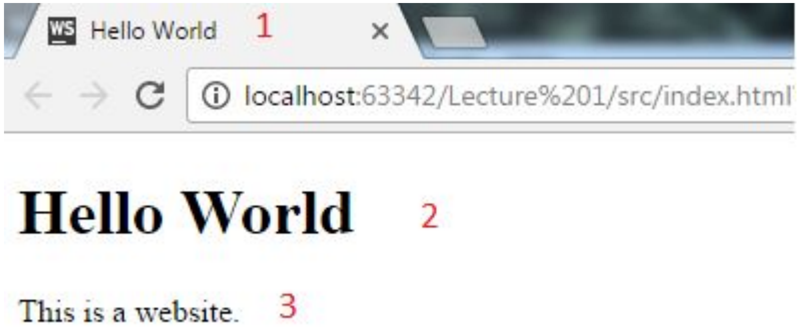


HTML

→ HyperText Markup Language

HTML PAGE STRUCTURE

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <title>Hello World</title>
        <link
            rel="stylesheet"
            href="style.css“
        >
    </head>
    <body>
        <h1>Hello World</h1>
        <p>This is a website.</p>
    </body>
</html>
```



WS Hello World 1 ×

localhost:63342/Lecture%201/src/index.html

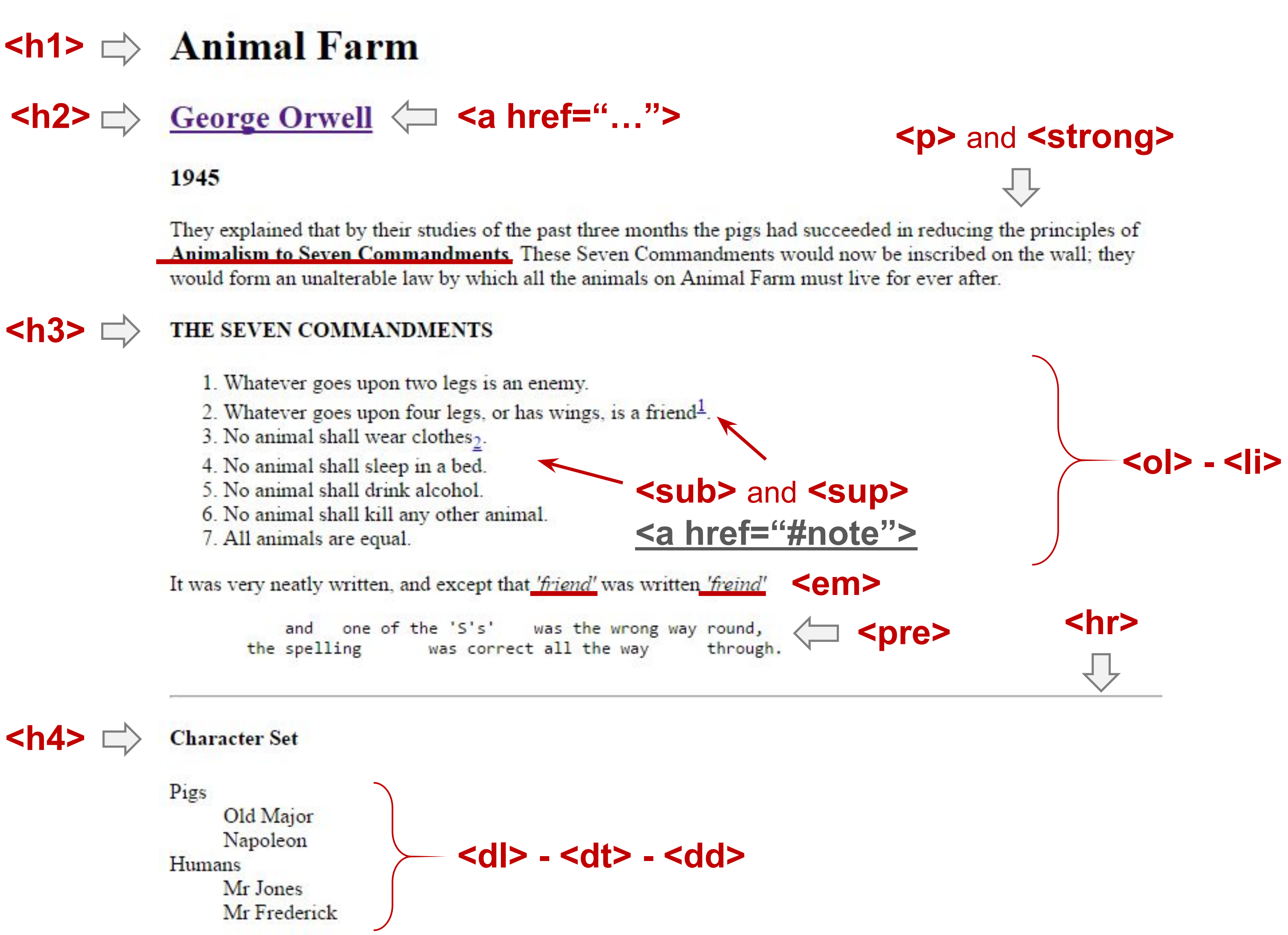
Hello World 2

This is a website. 3

HTML ELEMENTS

- Text formatting
- Forms and Input
- Images and Iframe
- Audio and Video
- Links
- Lists
- Tables
- Styles and Semantics
- Programming
- Meta Info

BASIC HTML TAGS - 1



<h1>

Animal Farm

<h2>

George Orwell

<a href=“…”>

<p> and <strong>

1945

They explained that by their studies of the past three months the pigs had succeeded in reducing the principles of **Animalism to Seven Commandments**. These Seven Commandments would now be inscribed on the wall; they would form an unalterable law by which all the animals on Animal Farm must live for ever after.

<h3>

THE SEVEN COMMANDMENTS

1. Whatever goes upon two legs is an enemy.
2. Whatever goes upon four legs, or has wings, is a friend[1].
3. No animal shall wear clothes$_2$.
4. No animal shall sleep in a bed.
5. No animal shall drink alcohol.
6. No animal shall kill any other animal.
7. All animals are equal.

<sub> and <sup>

<a href=“#note”>

<ol> - <li>

It was very neatly written, and except that *'friend'* was written *'freind'*

<em>

```
        and    one of the 'S's'    was the wrong way round,
     the spelling        was correct all the way      through.
```

<pre>

<hr>

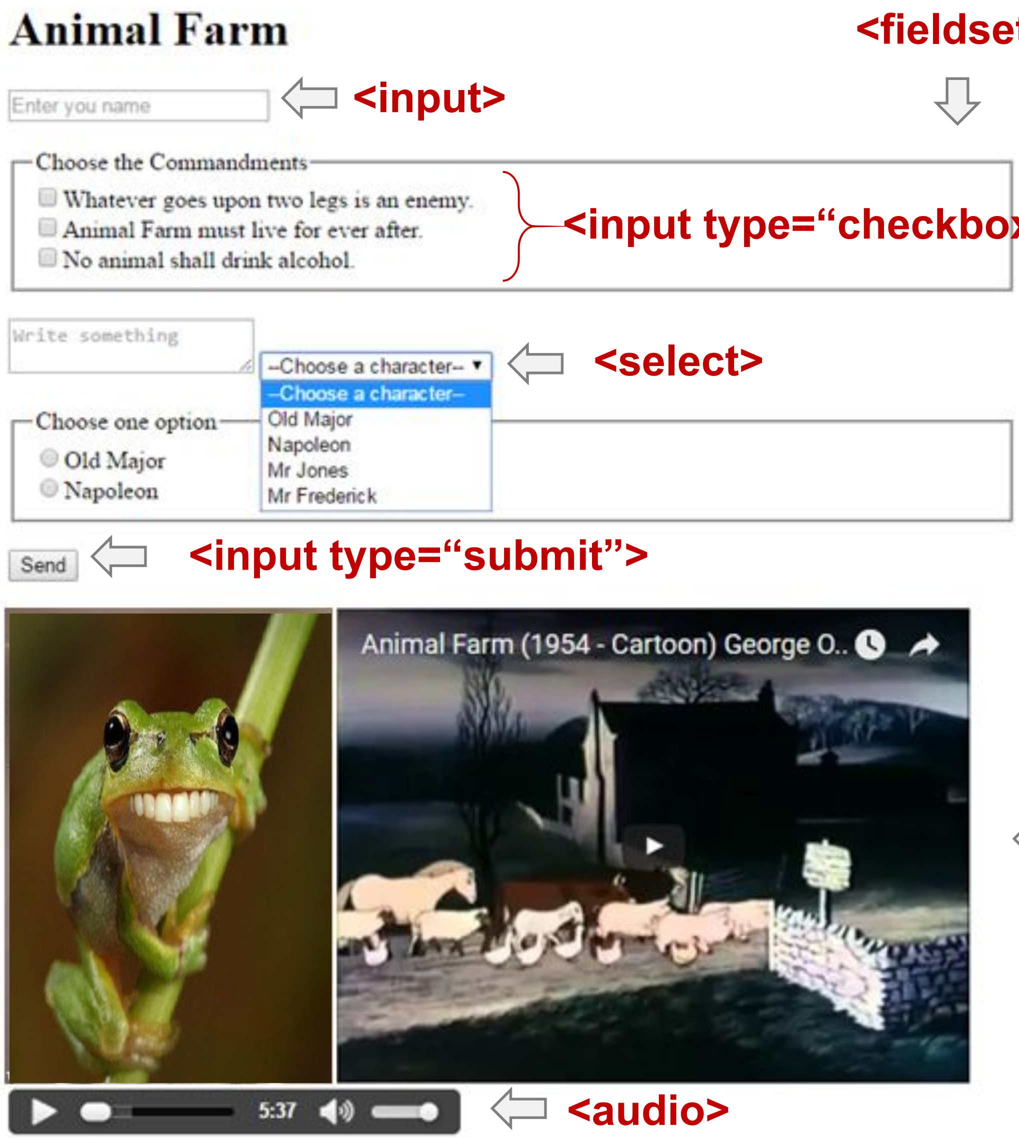
<h4>

Character Set

Pigs
    Old Major
    Napoleon
Humans
    Mr Jones
    Mr Frederick

<dl> - <dt> - <dd>

BASIC HTML TAGS - 2



Animal Farm

Enter you name

Choose the Commandments

□ Whatever goes upon two legs is an enemy.
□ Animal Farm must live for ever after.
□ No animal shall drink alcohol.

Write something

--Choose a character-- ▾

--Choose a character--
Old Major
Napoleon
Mr Jones
Mr Frederick

Choose one option

○ Old Major
○ Napoleon

Send

Animal Farm (1954 - Cartoon) George O..

5:37

<fieldset>

<input>

<input type=“checkbox”>

<textarea>

<select>

<input type=“radio”>

<input type=“submit”>

<img>

<iframe>
&
<video>

<audio>

STRUCTURAL VS. PHYSICAL FORMATTING

<b> vs. <strong>

&

<i> vs. <em>

VALID HTML – IS IT IMPORTANT?

Validation is a process of checking your documents against a formal Standard, such as those published by the World Wide Web Consortium (W3C) for HTML

WHY VALIDATE, WHAT ARE THE BENEFITS?

- Non uniform browser correction
- Rendering time
- Future proofing
- Google prefers valid code
- Accessibility

HTML5 validator: https://validator.w3.org/

- Self satisfaction

SEMANTIC

➔ Be understandable

CONTAINERS AND SEMANTIC HTML

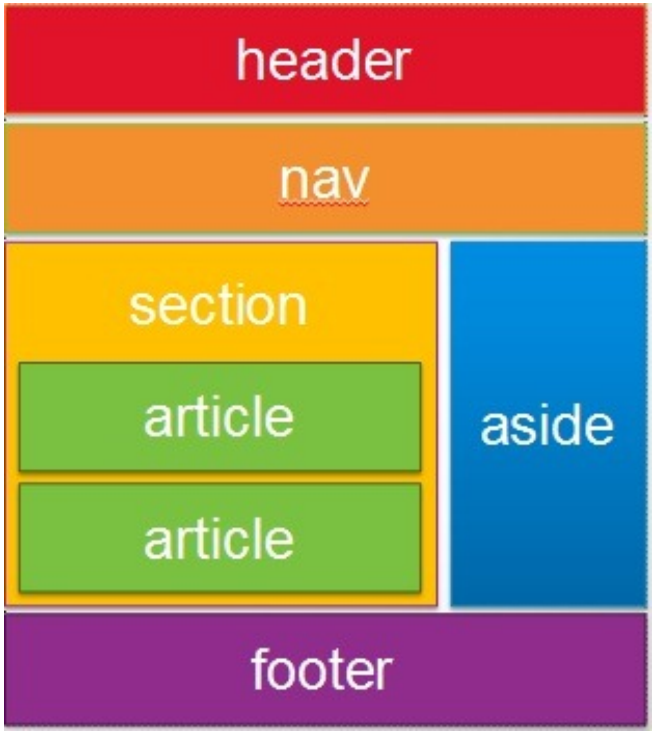
```
<div id="main">
  <div class="sidebar"></div>
  <div class="article"></div>

  <div class="article"></div>

</div>
```

→

```
<main>
  <aside></aside>
  <section>

     <article></article>

     <article></article>

  </section>

</main>
```

SEMANTIC HTML

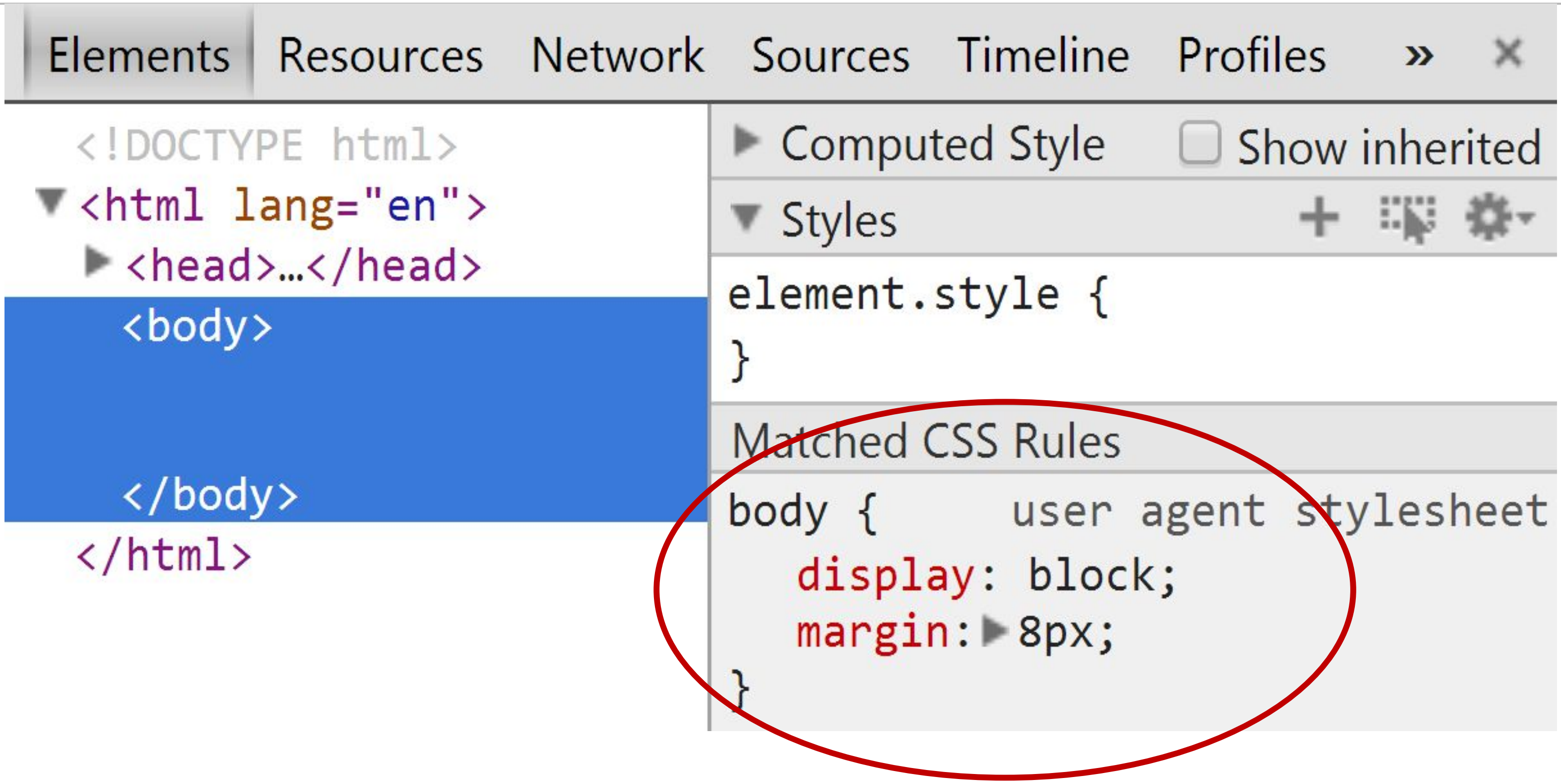
- Clearly defined regions for the developer
- Hooks for CSS without extra HTML
- Clearly defined regions for search engines and screen readers



CSS

→ Cascading Style Sheets

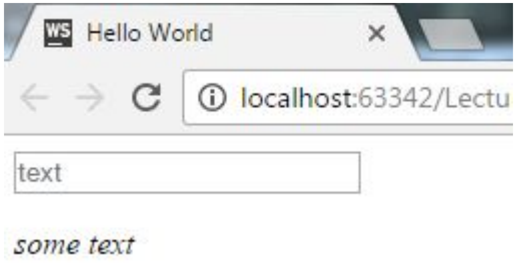
BROWSER STYLES



```
<!DOCTYPE html>
▼<html lang="en">
  ▶<head>…</head>
   <body>


   </body>
</html>
```

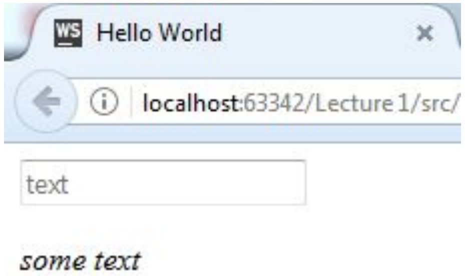
```
element.style {
}
```

```
body {        user agent stylesheet
   display: block;
   margin:▶8px;
}
```



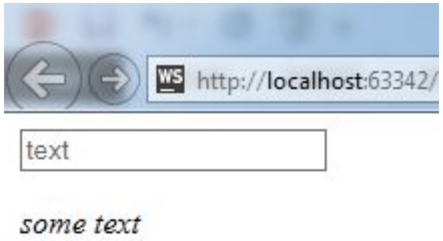
text

some text



text

some text

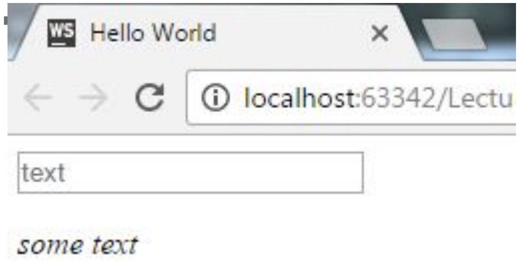


text

some text

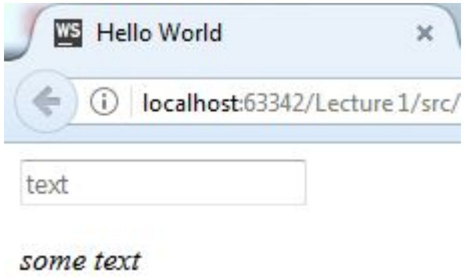
RESET.CSS

DEFAUL



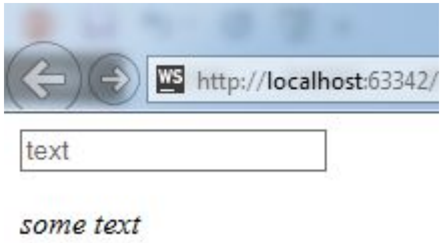
text

some text



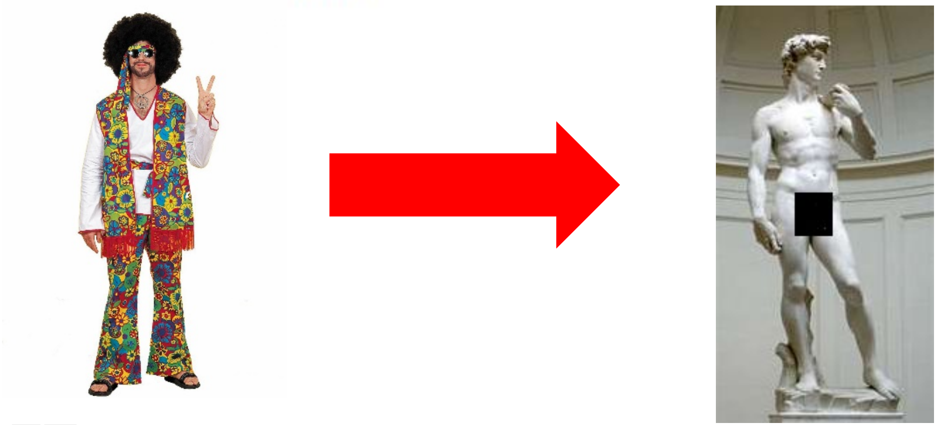
text

some text

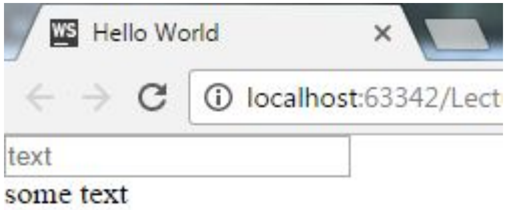


text

some text

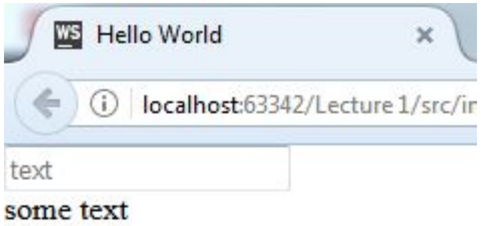


AFTER RESET



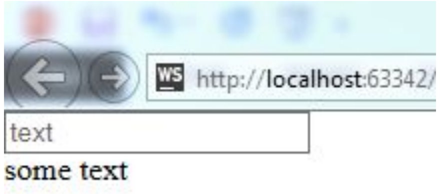
text

some text



text

some text

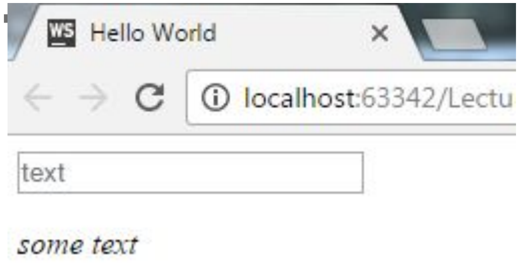


text

some text

NORMALIZE.CSS

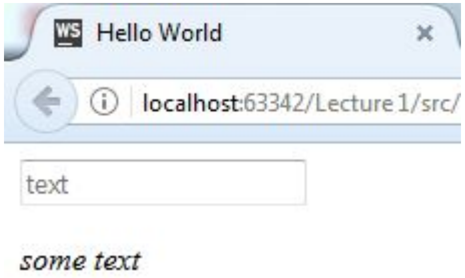
DEFAUL


Hello World

localhost:63342/Lectu

text

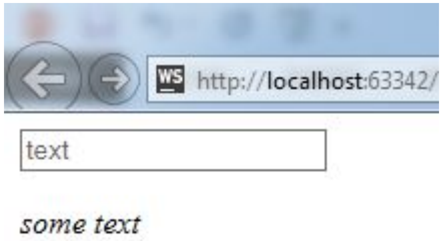
some text


Hello World

localhost:63342/Lecture 1/src/

text

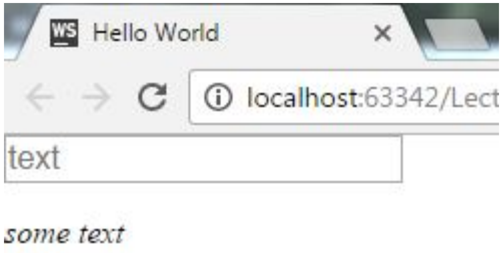
some text


http://localhost:63342/

text

some text

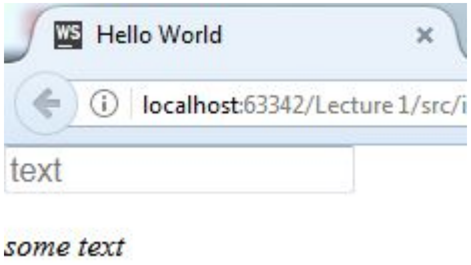
AFTER NORMALIZE


Hello World

localhost:63342/Lect

text

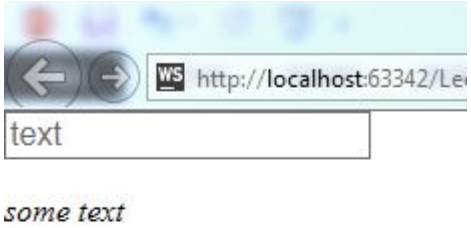
some text


Hello World

localhost:63342/Lecture 1/src/i

text

some text

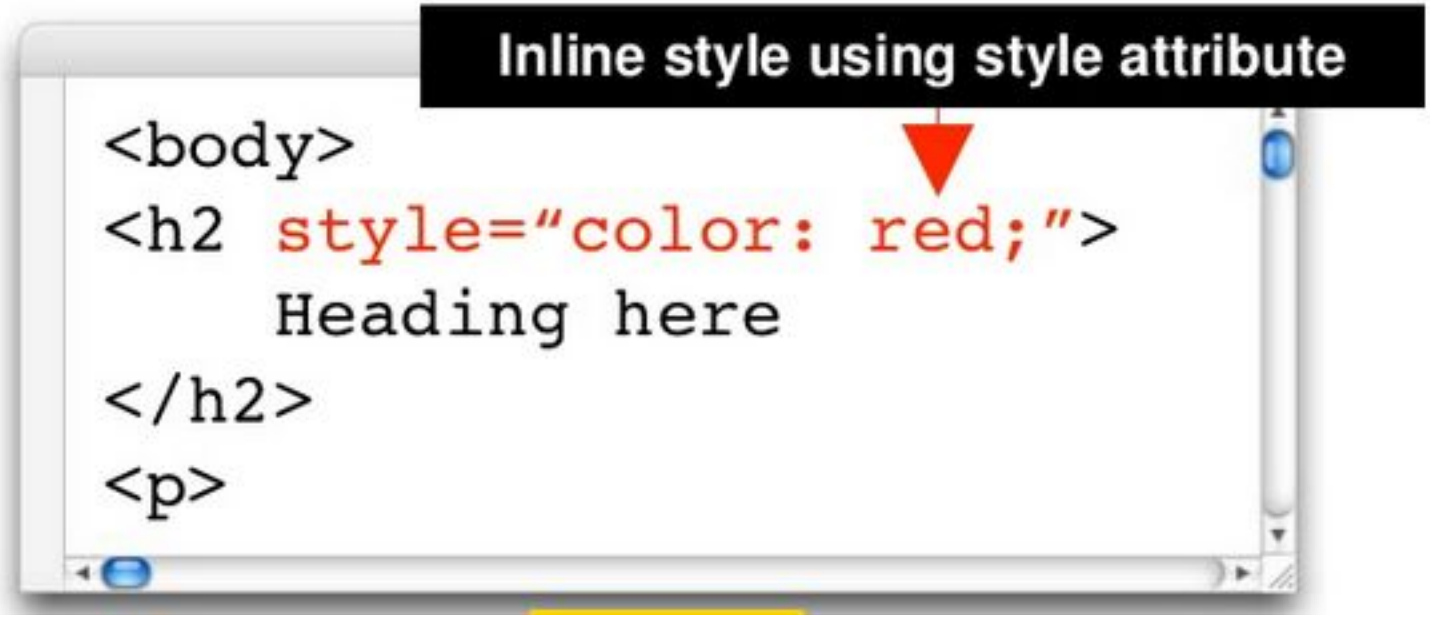

http://localhost:63342/Le

text

some text

HOW TO ADD CSS TO AN HTML PAGE?

1. INLINE

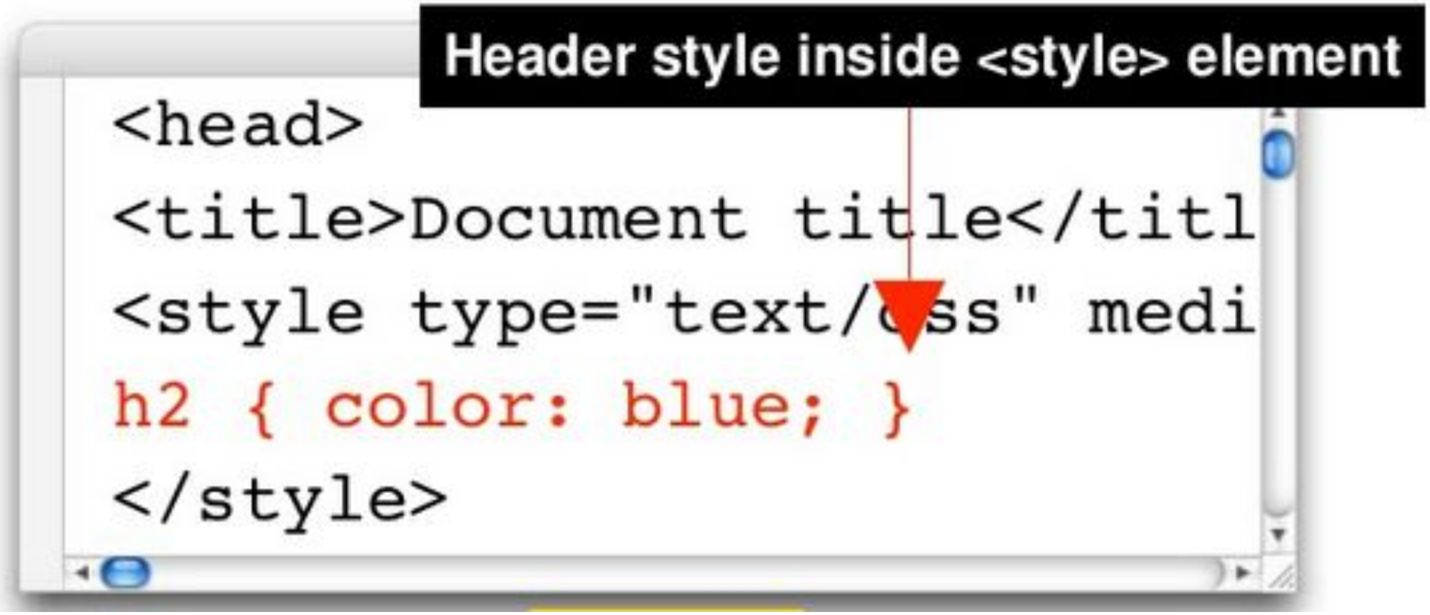


Inline style using style attribute

```
<body>
<h2 style="color: red;">
    Heading here
</h2>
<p>
```

HOW TO ADD CSS TO AN HTML PAGE?

2. EMBEDDED

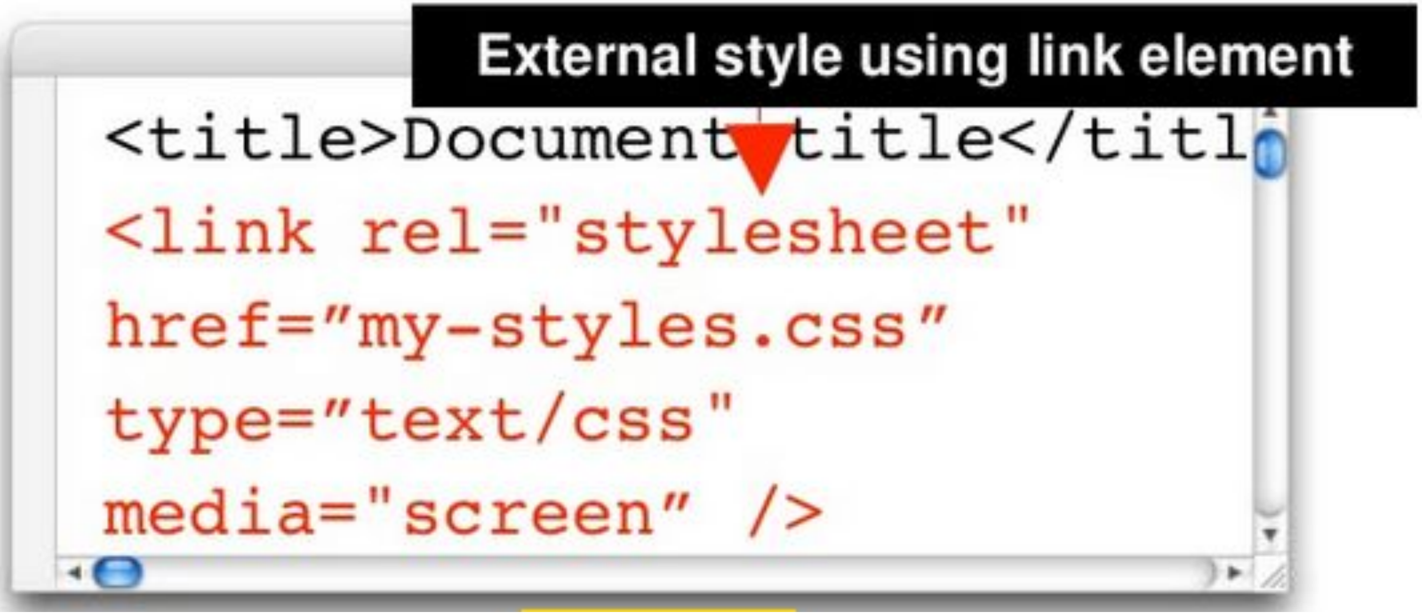


Header style inside <style> element

```
<head>
<title>Document title</titl
<style type="text/css" medi
h2 { color: blue; }
</style>
```

HOW TO ADD CSS TO AN HTML PAGE?

3. LINKED



External style using link element

```
<title>Document title</titl
<link rel="stylesheet"
href=”my-styles.css”
type=”text/css"
media="screen” />
```

HTML + CSS: ATTRIBUTES

HTML:

```
<div id="content" class="box content">...</div>

<div data-value=“true”>...</div>
```

CSS:

```
#content, .content {
    font-size: 14px;
}

div[data-value], div[data-value=‘true’] {
    color: red;
}
```

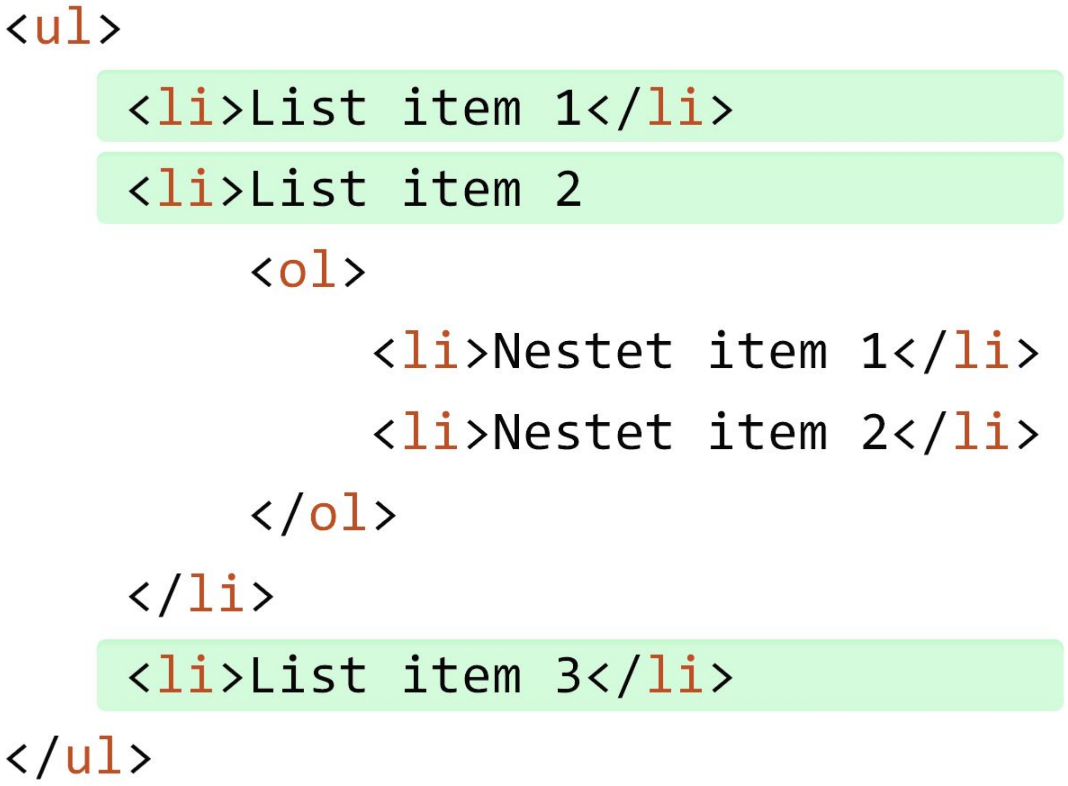
DESCENDANT COMBINATOR

```
ul li {
        …
}
```

```
<ul>
    <li>List item 1</li>
    <li>List item 2
        <ol>
            <li>Nestet item 1</li>
            <li>Nestet item 2</li>
        </ol>
    </li>
    <li>List item 3</li>
</ul>
```

CHILD COMBINATOR

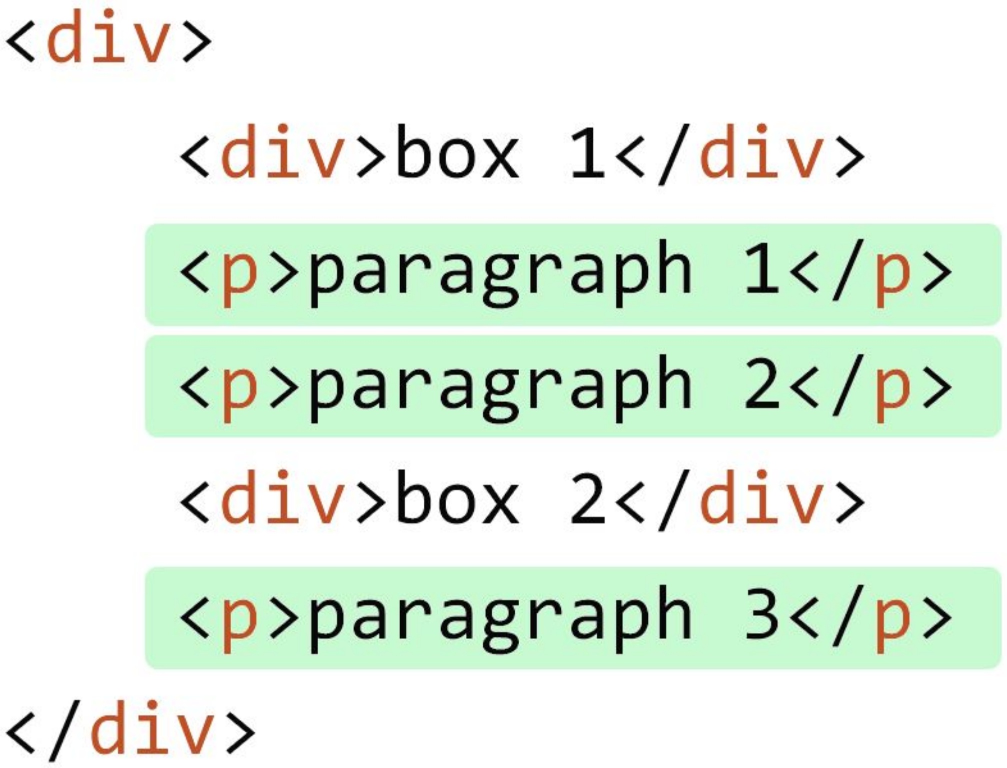
```
ul > li {
        …
}
```



```
<ul>
    <li>List item 1</li>
    <li>List item 2
        <ol>
            <li>Nestet item 1</li>
            <li>Nestet item 2</li>
        </ol>
    </li>
    <li>List item 3</li>
</ul>
```

GENERAL SIBLING COMBINATOR

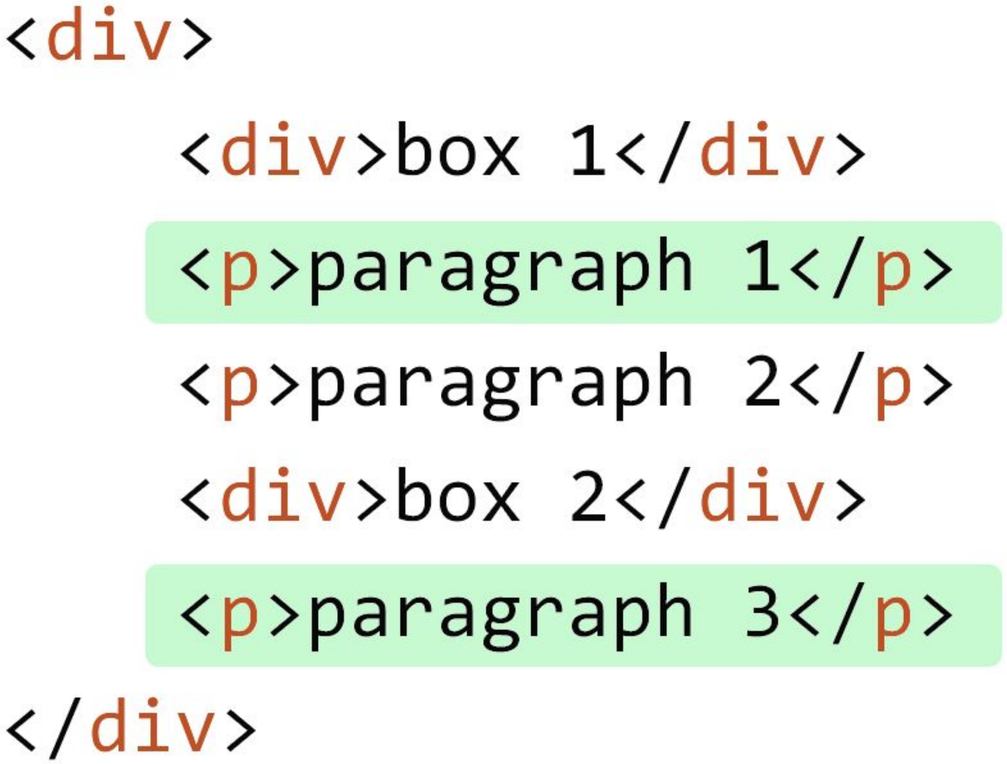
```
div ~ p {
    …
}
```



```
<div>
    <div>box 1</div>
    <p>paragraph 1</p>
    <p>paragraph 2</p>
    <div>box 2</div>
    <p>paragraph 3</p>
</div>
```

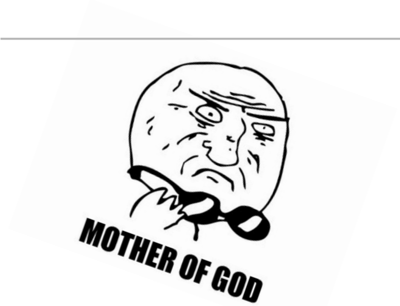
ADJACENT SIBLING COMBINATOR

```
div + p {
    …
}
```



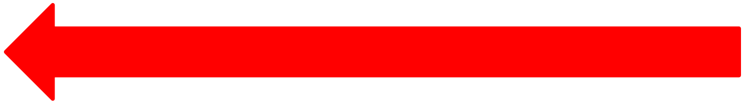
```
<div>
    <div>box 1</div>
    <p>paragraph 1</p>
    <p>paragraph 2</p>
    <div>box 2</div>
    <p>paragraph 3</p>
</div>
```

WHAT DOES THIS LINE MEAN?



```
#search-results > div.search-content ~ div + ul li * {
    margin: 0;
    list-style-type: square;
}
```

HOW DOES A BROWSER PARSE STYLES?



```
#search-results > div.search-content ~ div + ul li * {
    margin: 0;
    list-style-type: square;
}
```

CSS RULES PRIORITIES

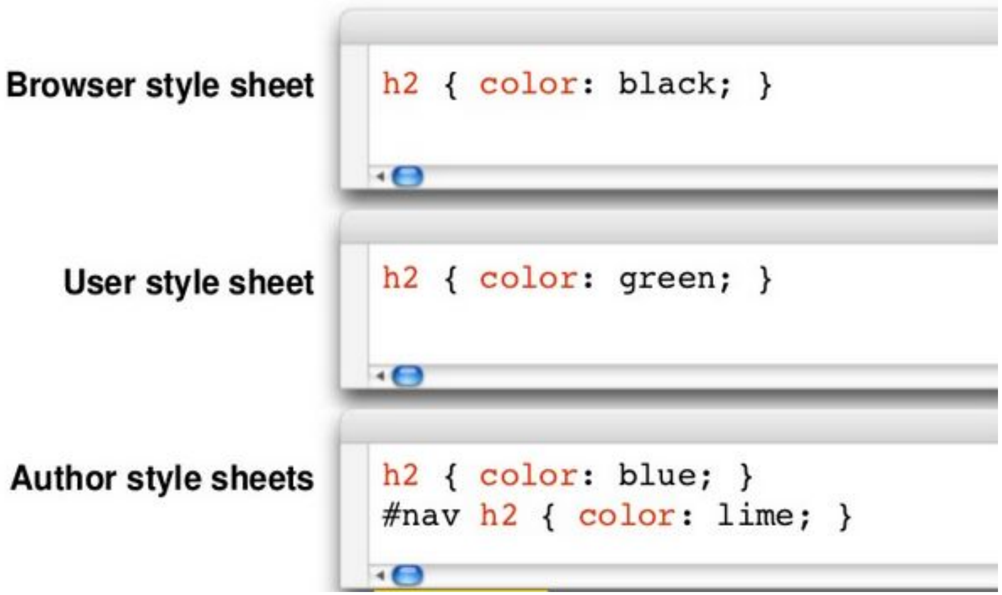
⟶ Or why aren’t my styles applied?

PRIORITY OF DIFFERENT STYLES

Low

Middle

High



Browser style sheet

```
h2 { color: black; }
```

User style sheet

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

Author style sheets

```
h2 { color: blue; }
#nav h2 { color: lime; }
```

!IMPORTANT MATTERS

Ascending order of style priorities

Browser styles

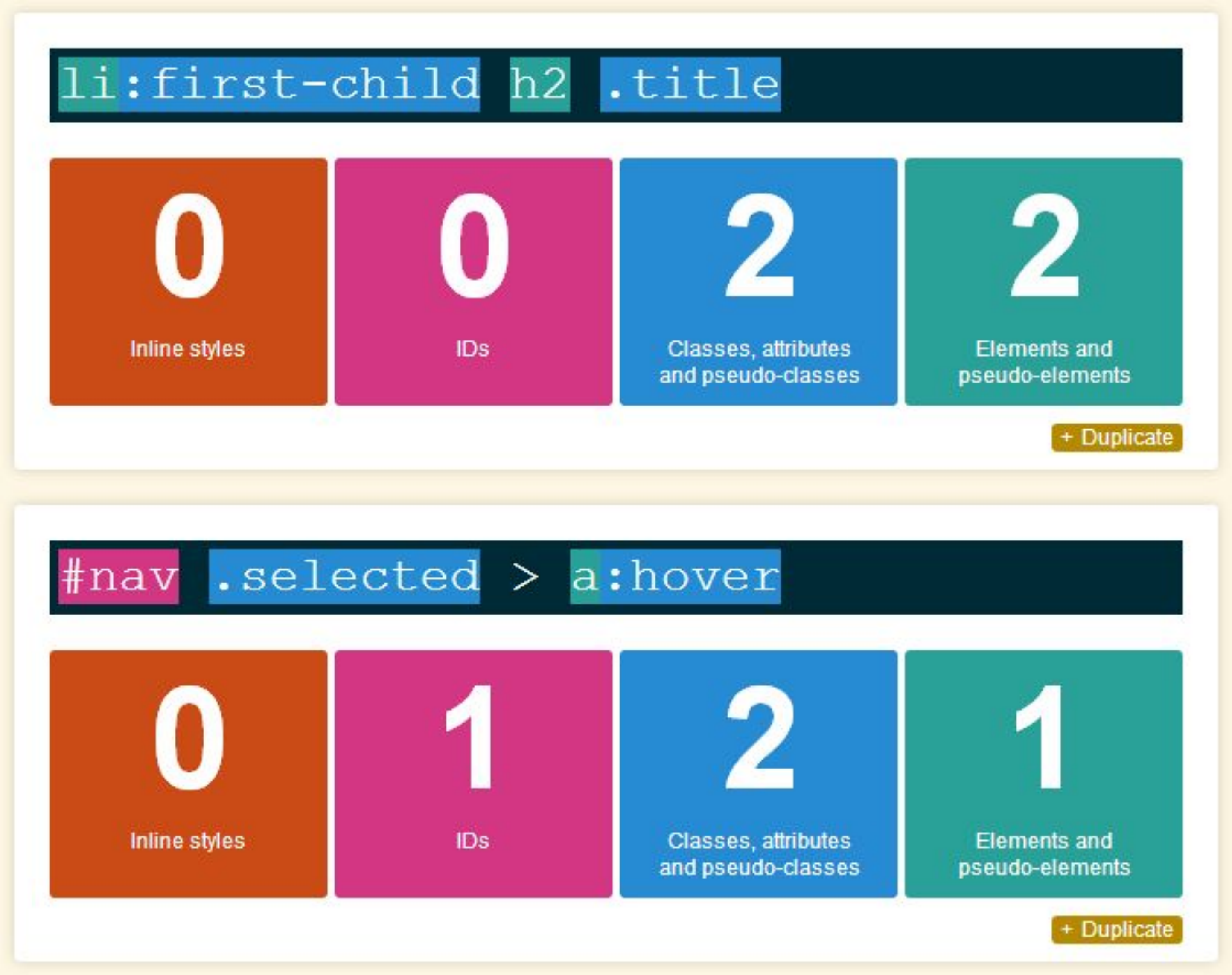
User style sheet - normal declarations

Author style sheet - normal declarations

Author style sheet - !important declarations

User style sheet - !important declaration

STYLE CALCULATOR



```
li:first-child h2 .title
```

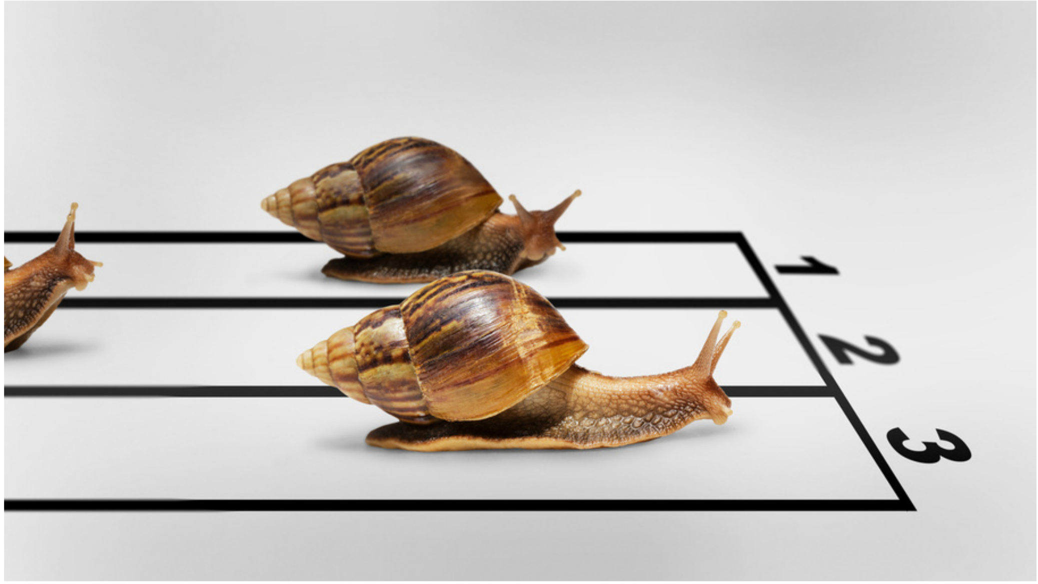
Inline styles	IDs	Classes, attributes and pseudo-classes	Elements and pseudo-elements
0	0	2	2

```
#nav .selected > a:hover
```

Inline styles	IDs	Classes, attributes and pseudo-classes	Elements and pseudo-elements
0	1	2	1

LAST DECLARATION

If two declarations have
the same priority, origin and specificity,
the last one “wins”.



WHAT DECLARATION WINS?

131

```
#search-results .search-content .search-list li:hover {
    margin: 10px;
    list-style-type: none;
}
```

114

```
#search-results > div.search-content ~ div + ul li {
    margin: 0;
    list-style-type: square;
}
```

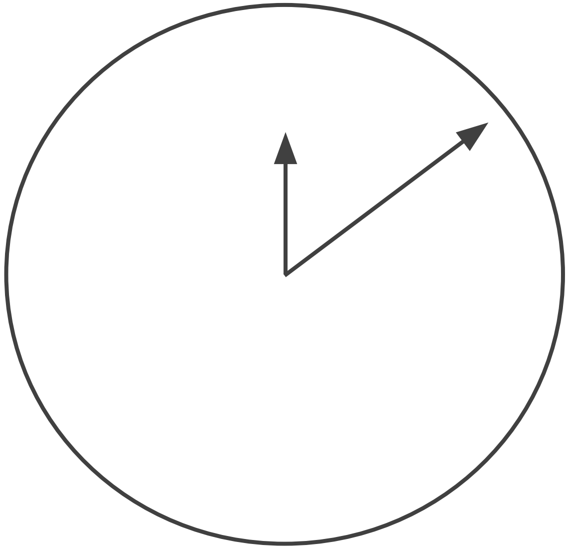
PROPERTIES

PROPERTIES

“trouble”

Top, **R**ight, **B**ottom, **L**eft

```
margin: 10px;

margin: 10px auto;

margin: 10px 20px 30%;

margin: 10px 2em 30px 40px;
```



COMPOUND PROPERTIES

```
                           border-width: 1em;
border: 1em dashed red; ⟺  border-style: dashed;
                           border-color: red;
```

VENDOR PREFIXES

```
.example {
    -webkit-transform: rotate(30deg);
        -ms-transform: rotate(30deg);
            transform: rotate(30deg);
}
```

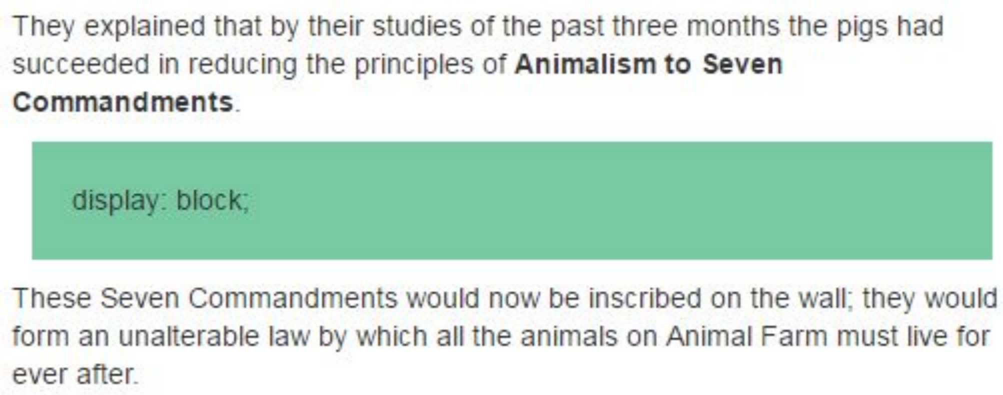
https://css-tricks.com/autoprefixer/

DISPLAY

→ https://htmlacademy.ru/courses/44

DISPLAY: NONE, BLOCK, INLINE

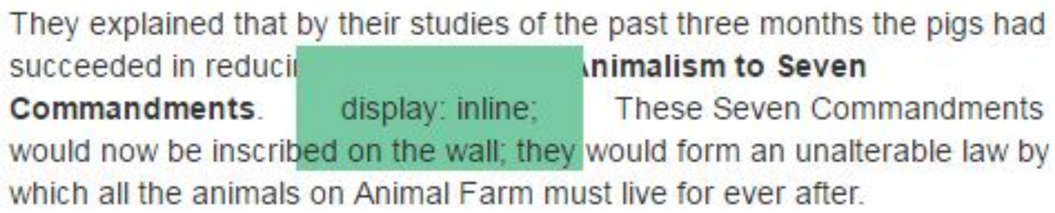
- None: <head>, <title>
- Block: <p>, <h2>
- Inline: <span>, <em>



They explained that by their studies of the past three months the pigs had succeeded in reducing the principles of **Animalism to Seven Commandments**.

display: block;

These Seven Commandments would now be inscribed on the wall; they would form an unalterable law by which all the animals on Animal Farm must live for ever after.

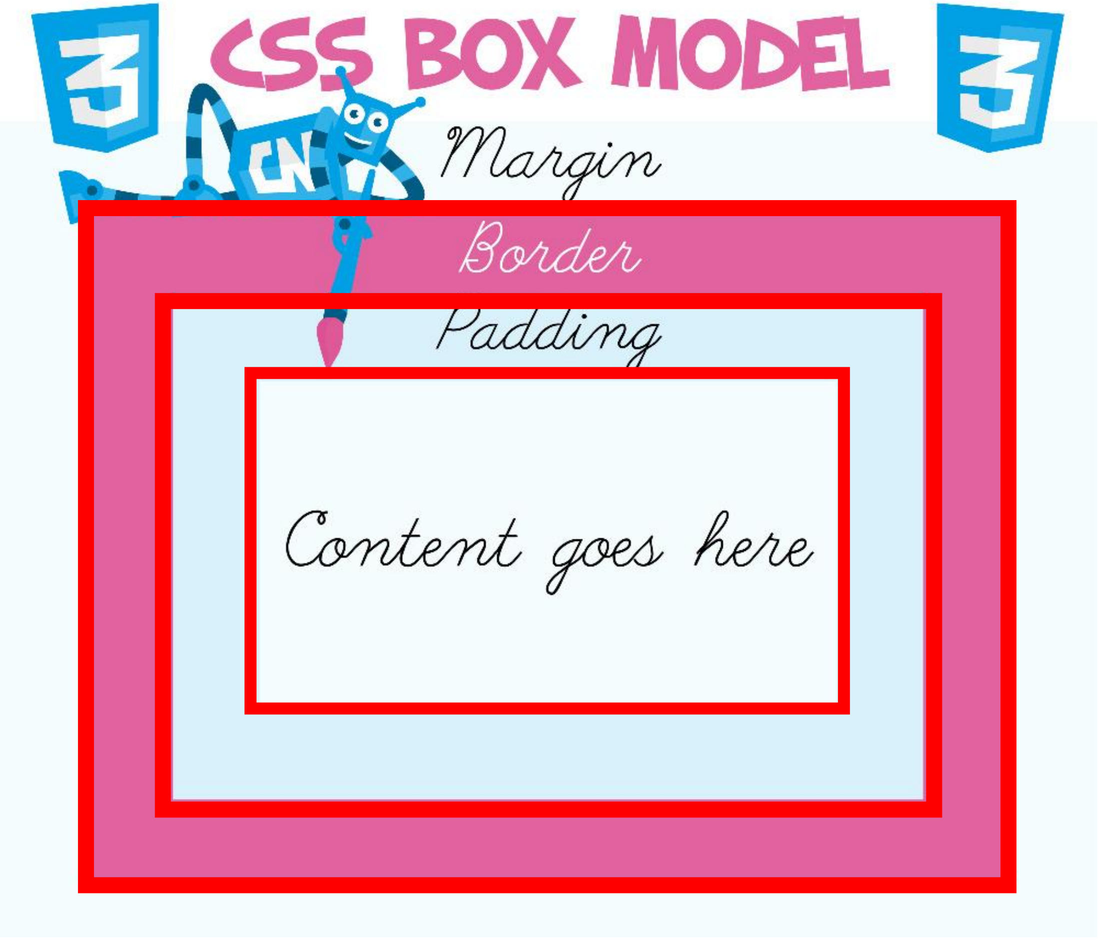


They explained that by their studies of the past three months the pigs had succeeded in reduci **nimalism to Seven Commandments**. display: inline; These Seven Commandments would now be inscribed on the wall; they would form an unalterable law by which all the animals on Animal Farm must live for ever after.

BOX MODEL

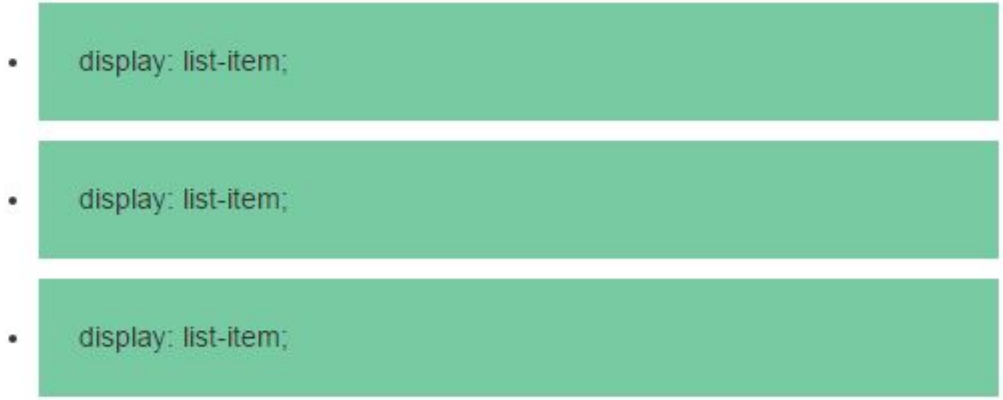
Box-sizing:

- content-box
- border-box



Affects width and height of an element

DISPLAY: LIST-ITEM



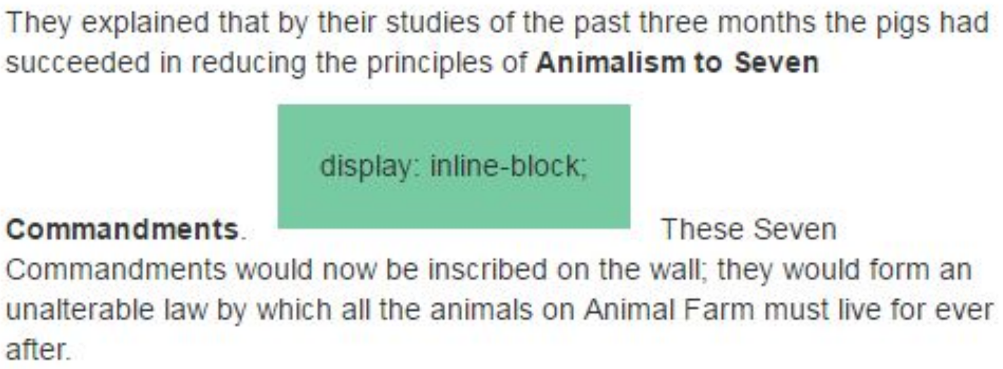
- display: list-item;
- display: list-item;
- display: list-item;

DISPLAY: TABLE

```
table    { display: table }
tr       { display: table-row }

thead    { display: table-header-group }
tbody    { display: table-row-group }
tfoot    { display: table-footer-group }

col      { display: table-column }
colgroup { display: table-column-group }
td, th   { display: table-cell }
caption  { display: table-caption }
```

DISPLAY: INLINE-BLOCK

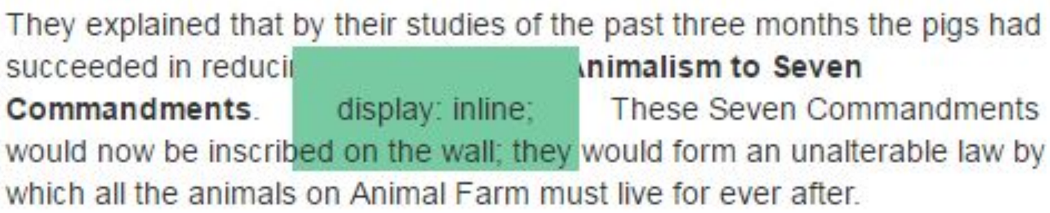
- Is used to display elements side-by-side*
- Has vertical alignment (default is baseline)
- Can be centered
- Can be justified
- * Similar to “float: left/right;”



They explained that by their studies of the past three months the pigs had succeeded in reducing the principles of **Animalism to Seven**

display: inline-block;

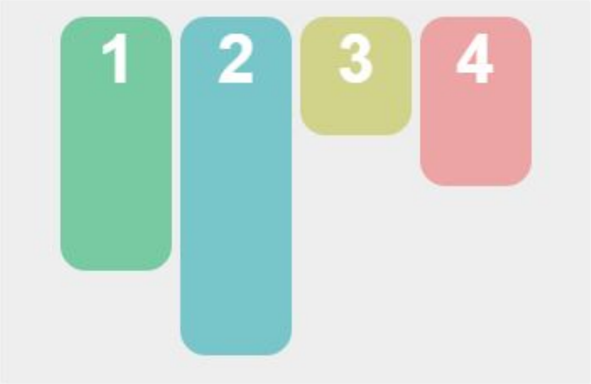
Commandments. These Seven Commandments would now be inscribed on the wall; they would form an unalterable law by which all the animals on Animal Farm must live for ever after.

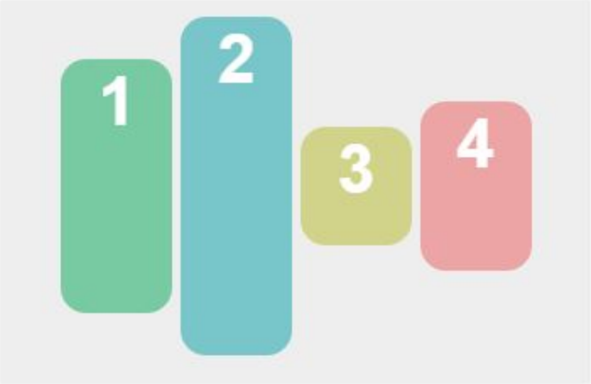


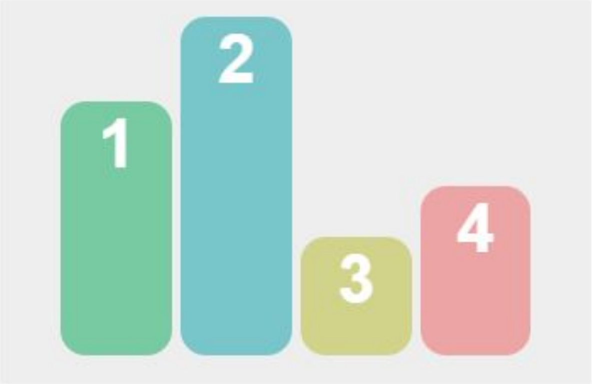
They explained that by their studies of the past three months the pigs had succeeded in reduci **nimalism to Seven** **Commandments**. display: inline; These Seven Commandments would now be inscribed on the wall; they would form an unalterable law by which all the animals on Animal Farm must live for ever after.

DISPLAY: INLINE-BLOCK

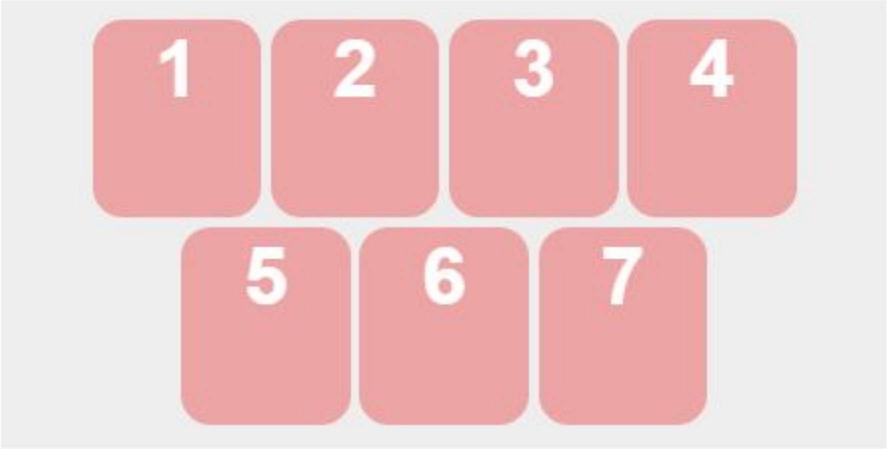
Vertical-align: **top** **middle** **bottom**

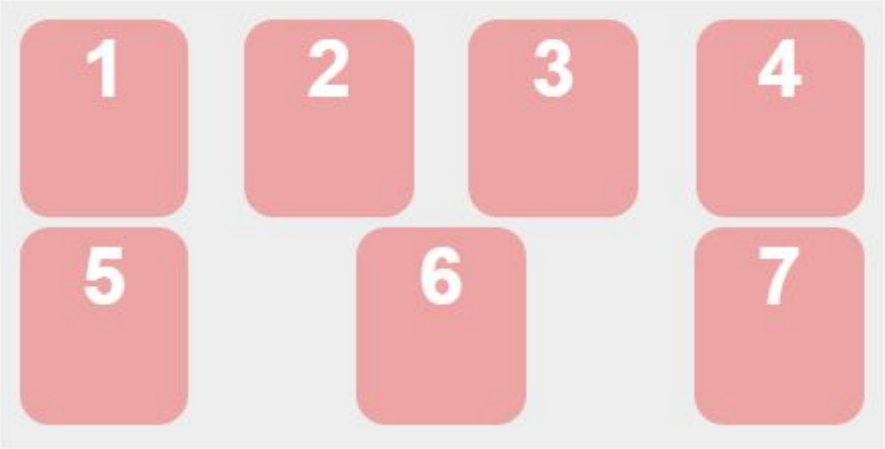






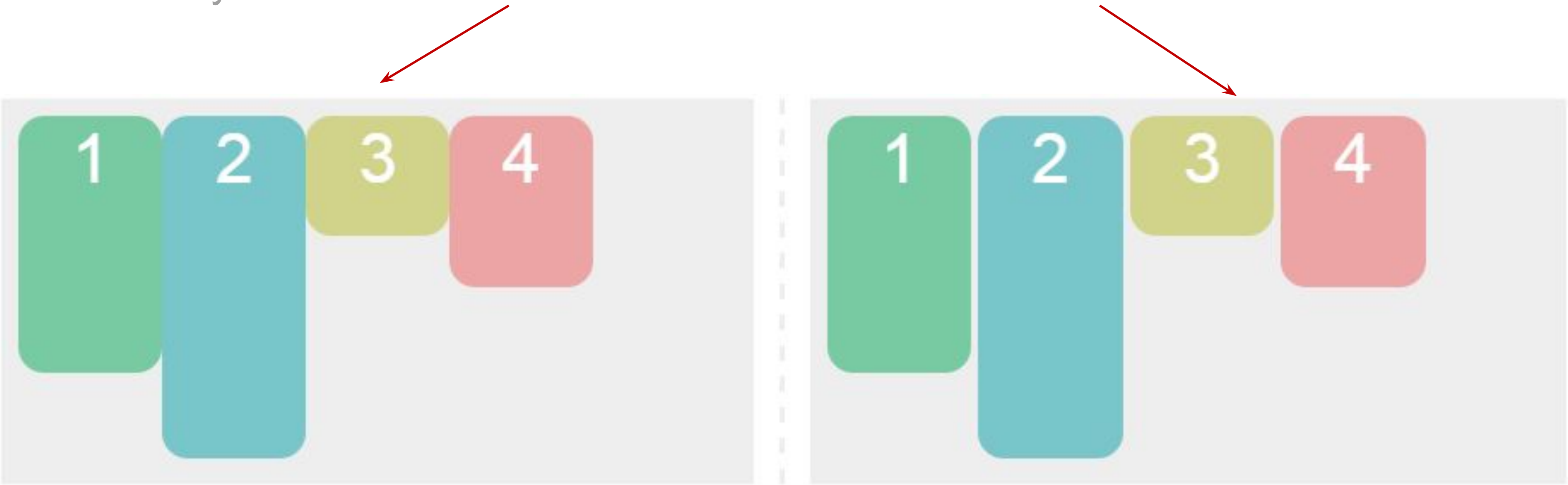
Text-align: **center** **justify**





FLOAT VS INLINE-BLOCK

for side-by-side elements



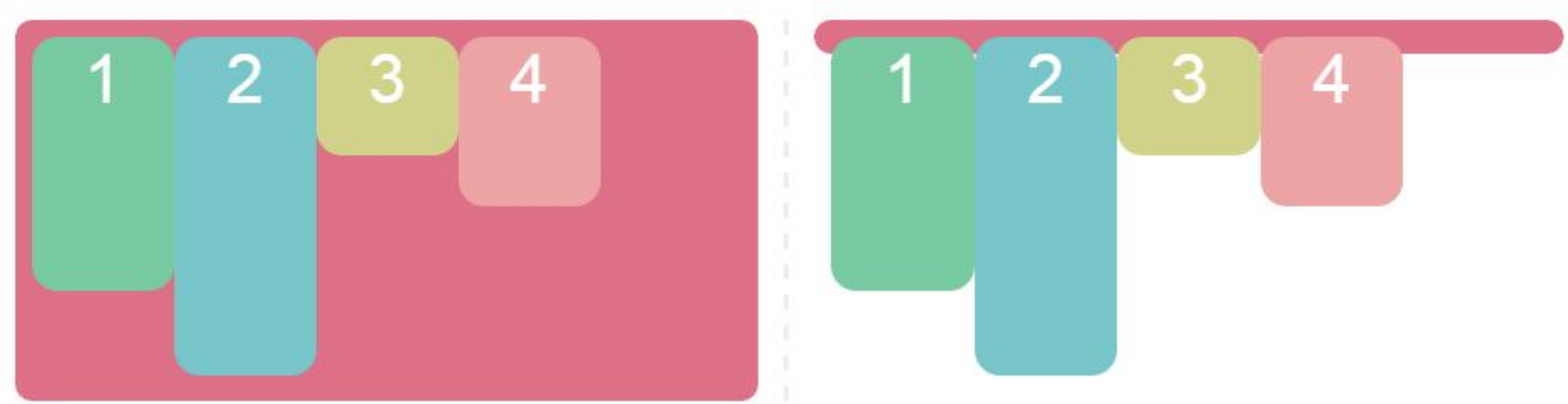
- ‘Clearfix’ needed for parent element
- Elements’ order depends on left/right value
- Top aligned by default

- Space between elements due to ‘inline’ display
- Baseline aligned by default

FLOAT VS INLINE-BLOCK

Floats must have a cleat fix!

Parent with a ‘clear fix’… **…and without it**

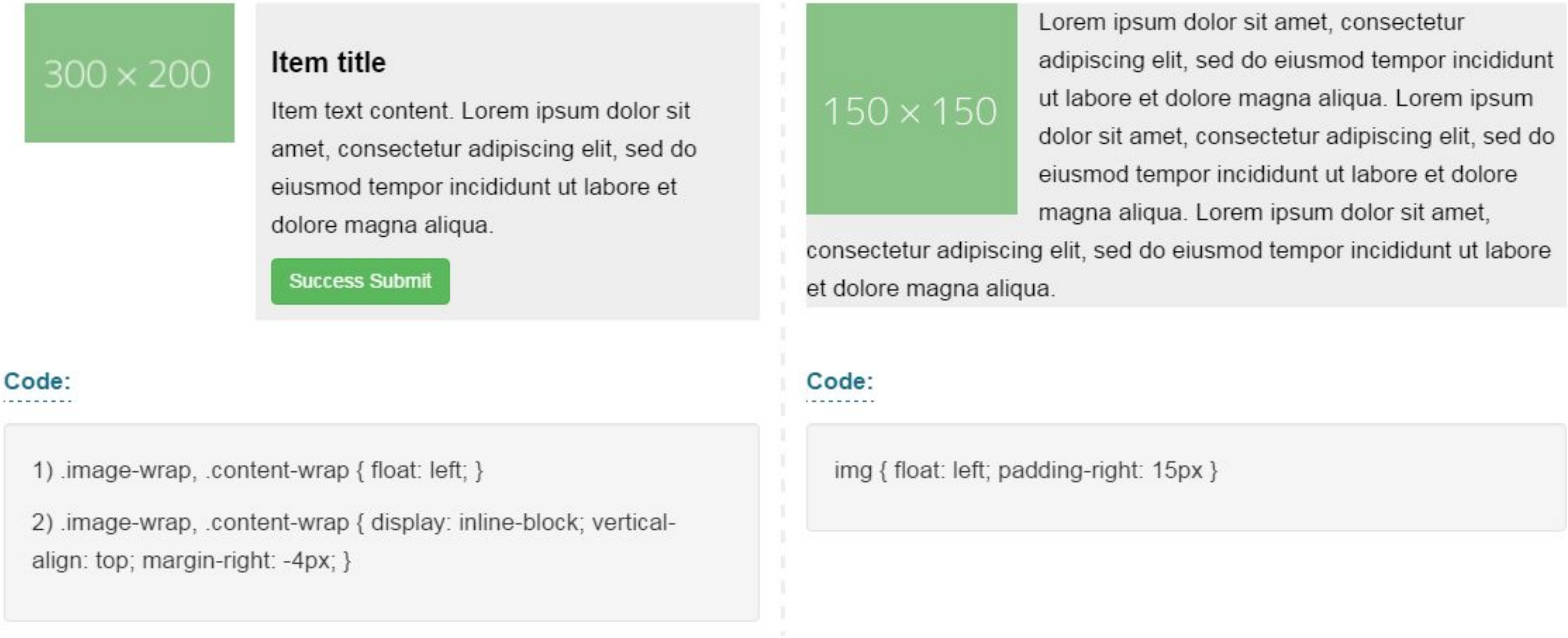


```
.parent: before,  .parent: after {
     content: ‘ ’;
     display: table;
}
.parent: after { clear: both; }
```

```
.parent { display: flow-root; }
```

FLOAT VS INLINE-BLOCK

for media alongside some content



Item title

Item text content. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Code:

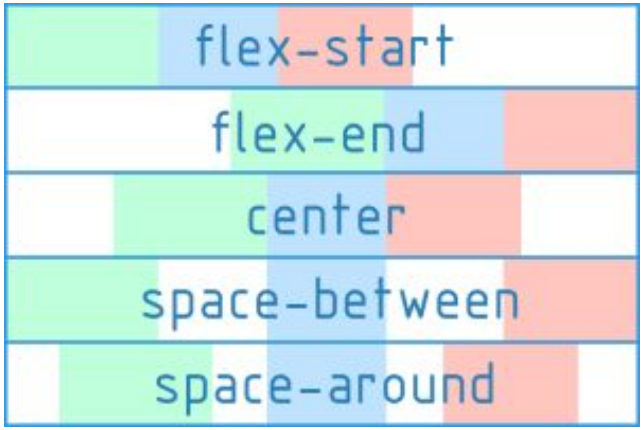
```
1) .image-wrap, .content-wrap { float: left; }

2) .image-wrap, .content-wrap { display: inline-block; vertical-align: top; margin-right: -4px; }
```

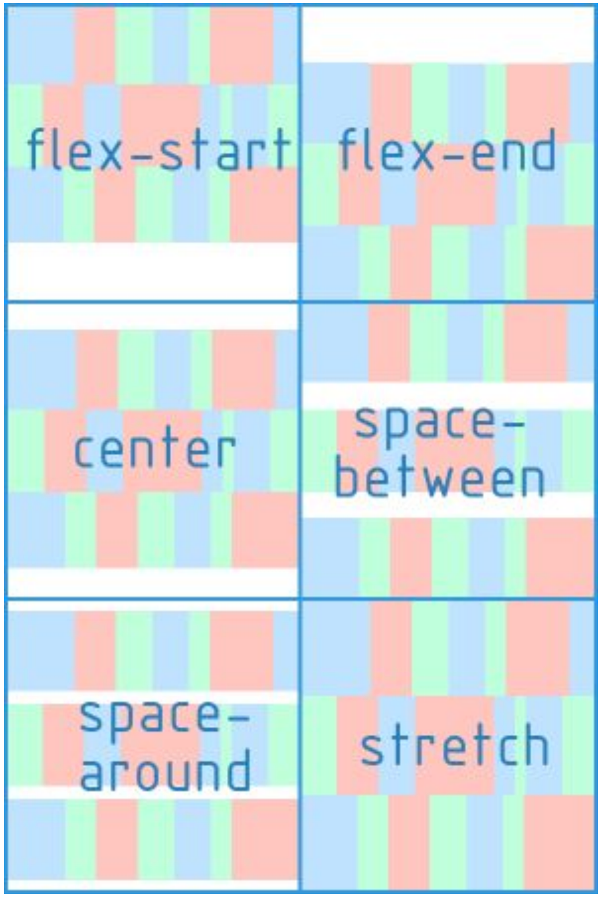
Code:

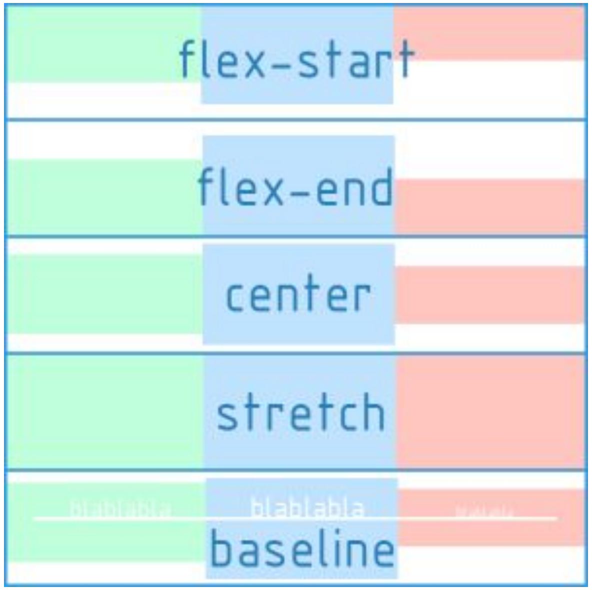
```
img { float: left; padding-right: 15px }
```

DISPLAY: FLEX

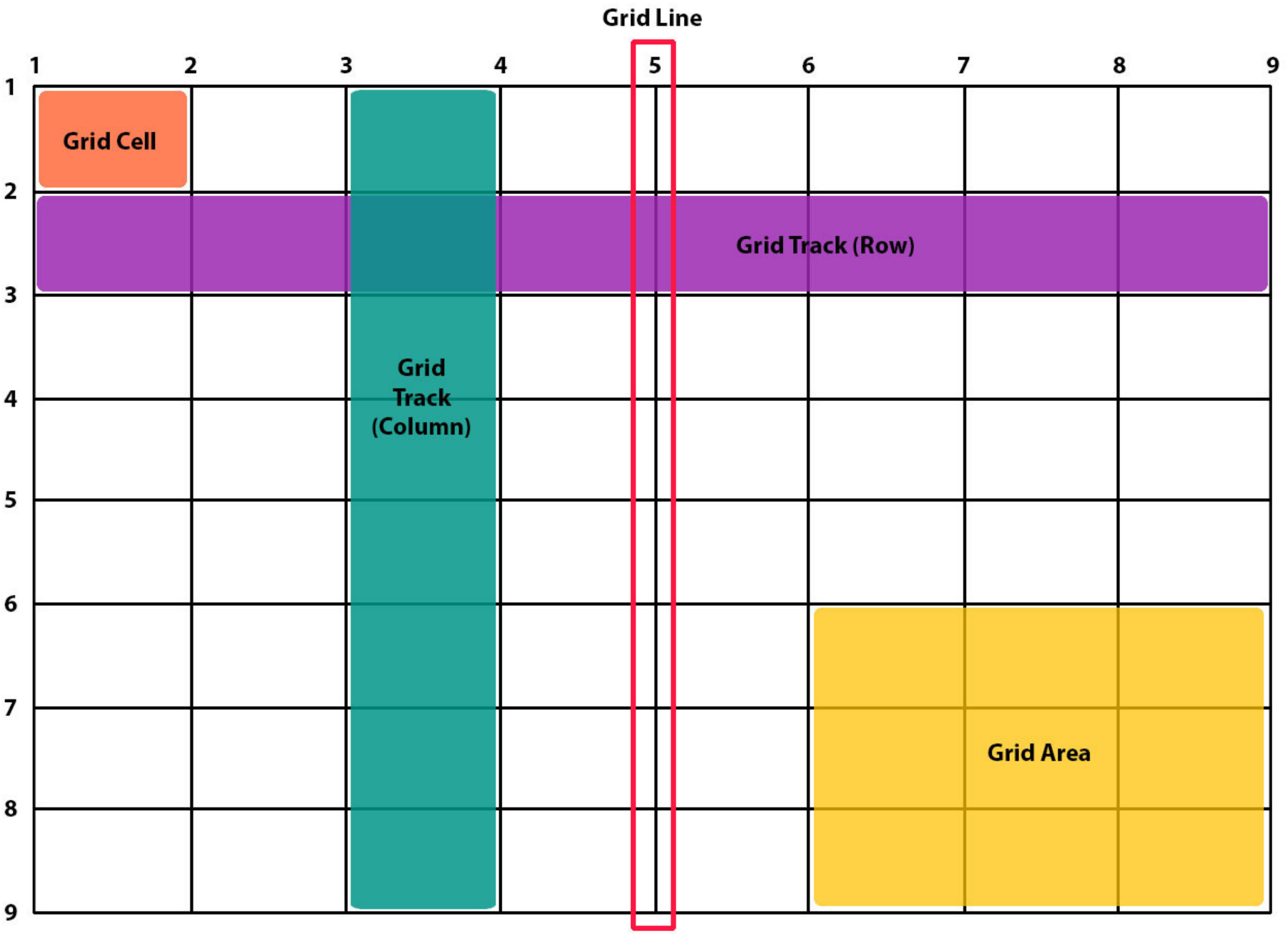
https://css-tricks.com/snippets/css/a-guide-to-flexbox/







DISPLAY: GRID | INLINE-GRID



https://css-tricks.com/snippets/css/complete-guide-grid/

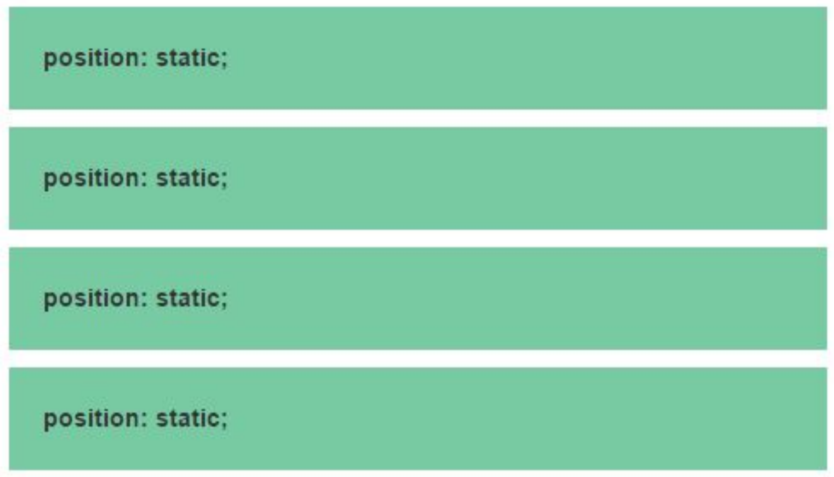
POSITION

→ Static, relative, absolute, fixed, sticky

POSITION: STATIC, RELATIVE

Left / Right / Top / Bottom / Z-index

✗



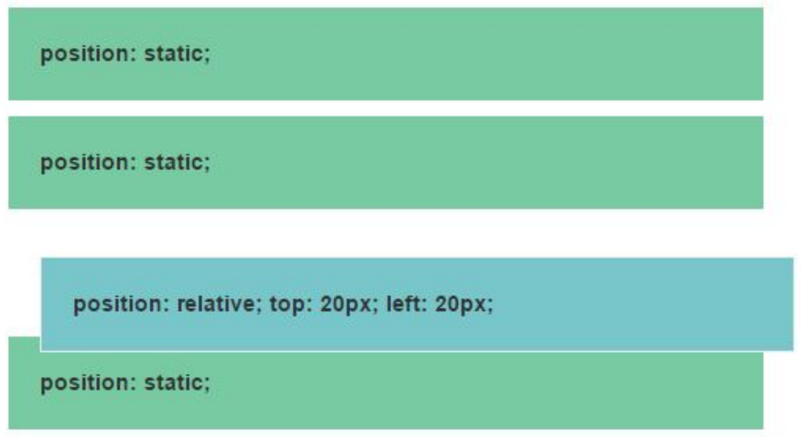
position: static;

position: static;

position: static;

position: static;

✓



position: static;

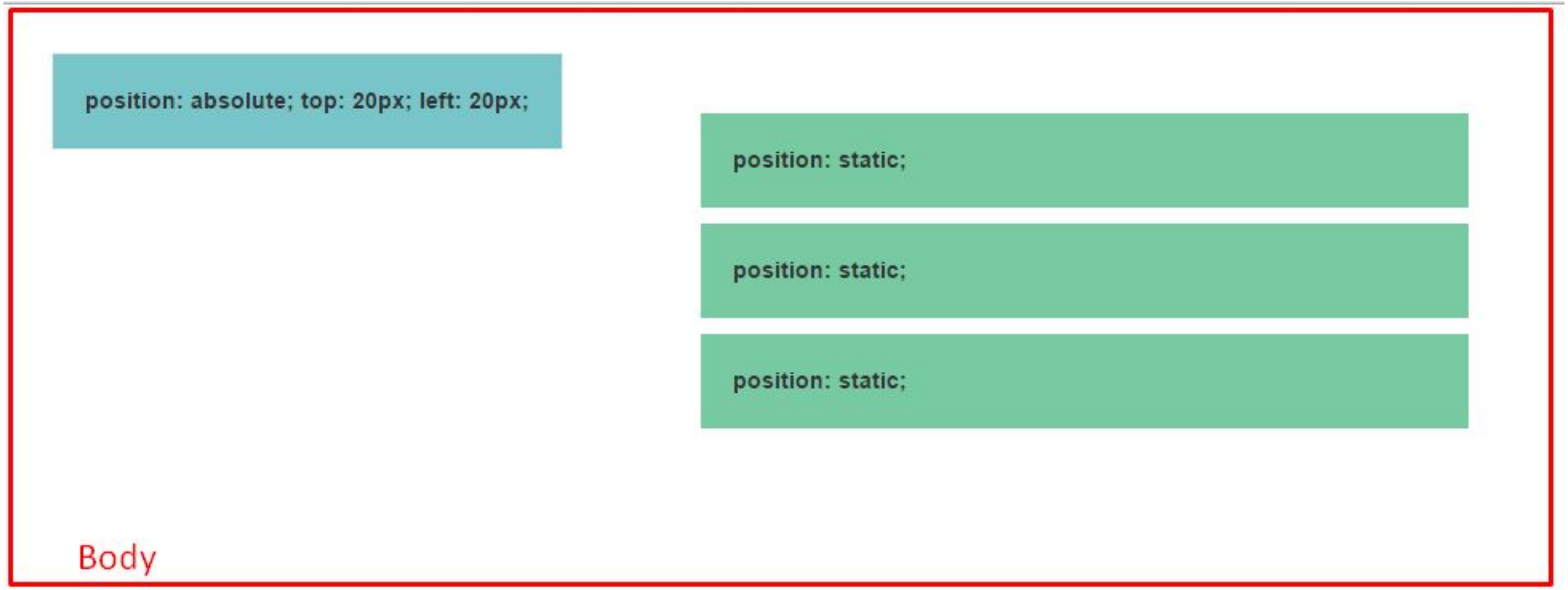
position: static;

position: relative; top: 20px; left: 20px;

position: static;

POSITION: ABSOLUTE

Positioned relative to body



position: absolute; top: 20px; left: 20px;

position: static;

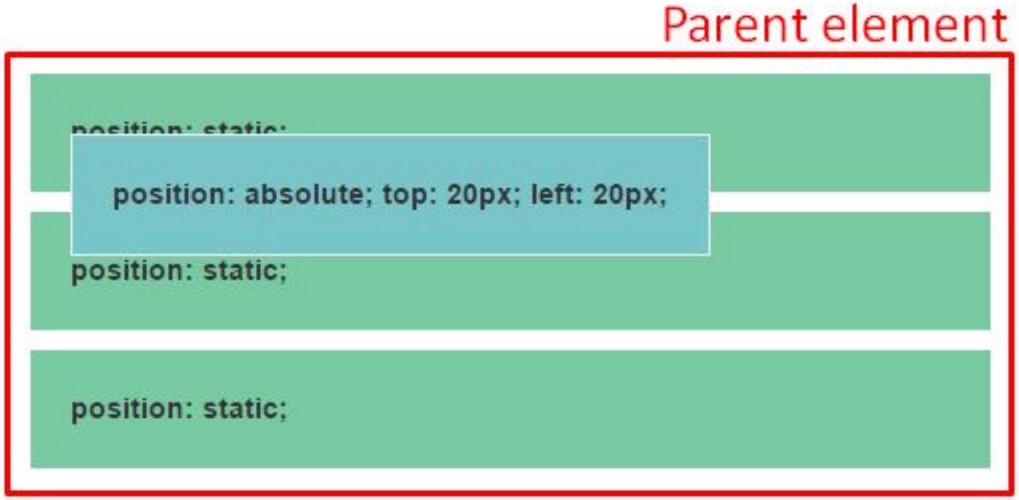
position: static;

position: static;

POSITION: RELATIVE, ABSOLUTE

Absolute elements are positioned

relative to the first positioned parent



Parent element

position: static;

position: absolute; top: 20px; left: 20px;

position: static;

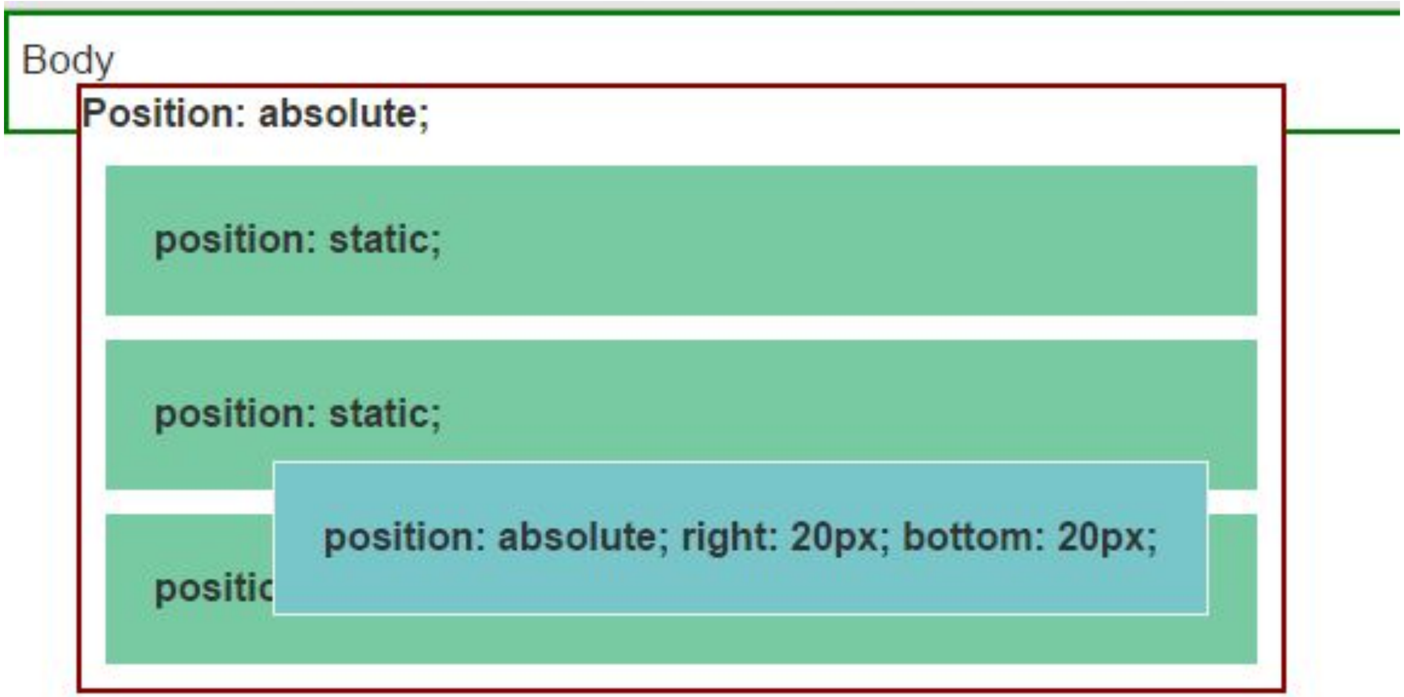
position: static;

POSITION: ABSOLUTE

‘Parent’ is positioned relative to ‘body’;

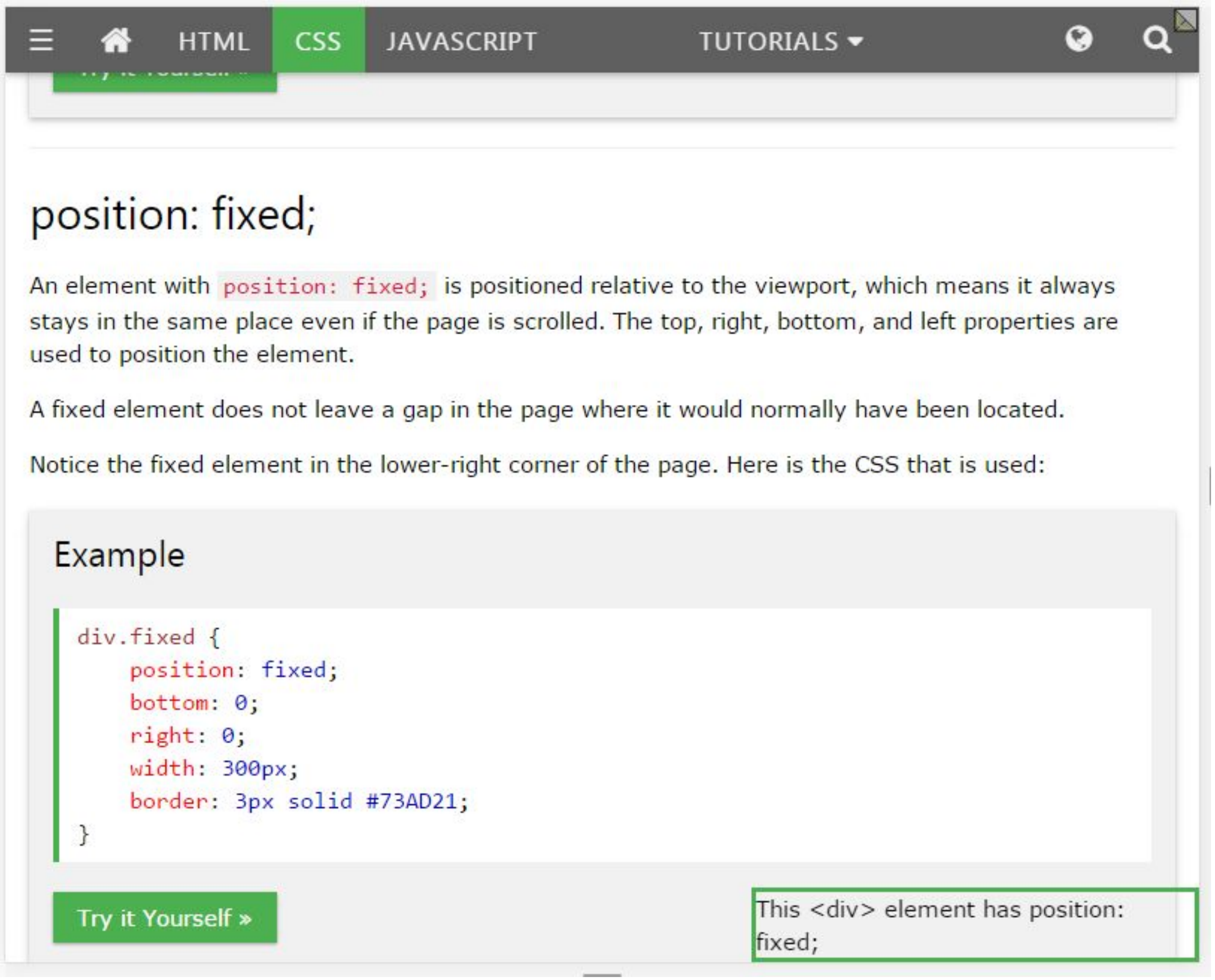
‘children’ are positioned relative to ‘parent’;

Absolute positioning is not included in width/height;



POSITION: FIXED, STICKY

Positioned relative to the viewport



position: fixed;

An element with `position: fixed;` is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element.

A fixed element does not leave a gap in the page where it would normally have been located.

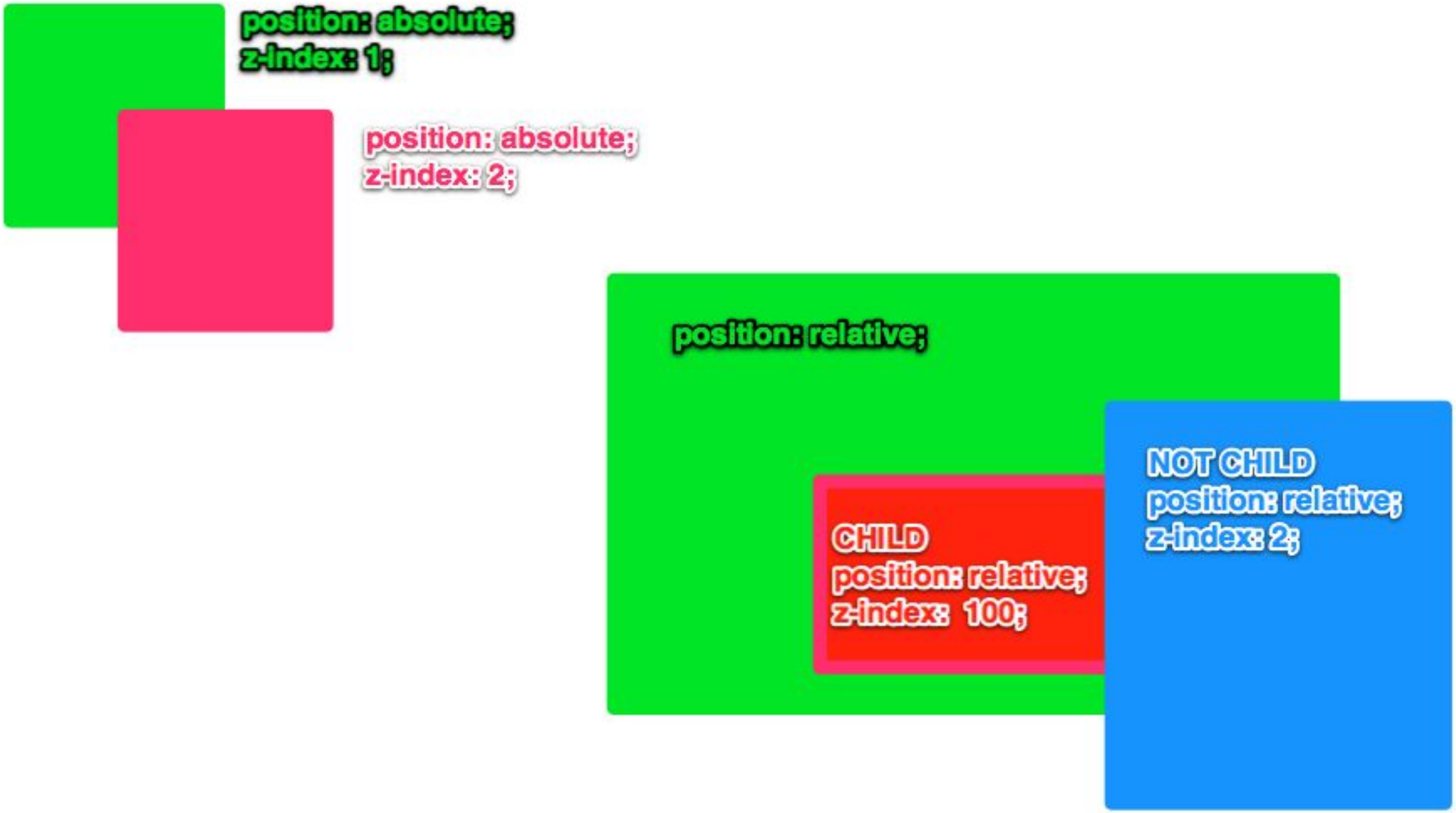
Notice the fixed element in the lower-right corner of the page. Here is the CSS that is used:

Example

```
div.fixed {
    position: fixed;
    bottom: 0;
    right: 0;
    width: 300px;
    border: 3px solid #73AD21;
}
```

This <div> element has position: fixed;

Z-INDEX: WTF



position: absolute;
z-index: 1;

position: absolute;
z-index: 2;

position: relative;

CHILD
position: relative;
z-index: 100;

NOT CHILD
position: relative;
z-index: 2;

PSEUDO CLASSES AND ELEMENTS

➔ Or how to make your content dynamic

LINKS AND USER ACTIONS

```
a:link { color: green }
a:visited { color: purple }

a:hover { color: yellow }
a:active { color: red }
a:focus { outline: none }
```

Example

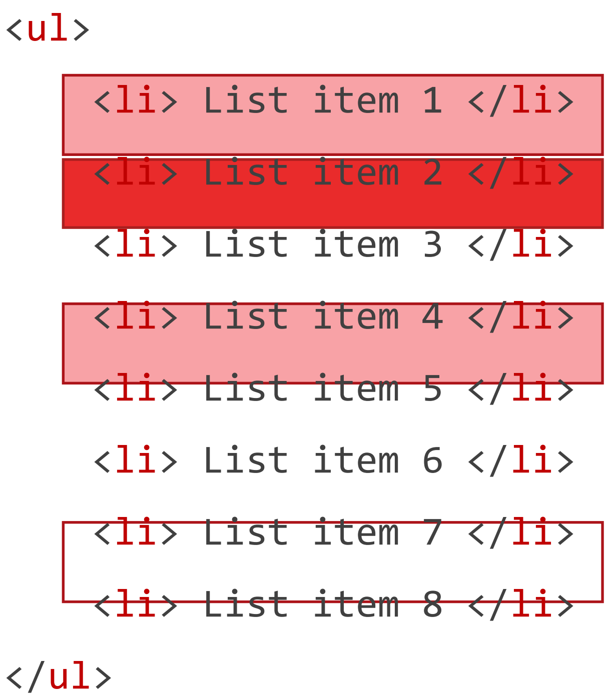
STRUCTURE PSEUDO CLASSES

```
:first-child {

        color: red;

}

:last-child {

        color: grey;

}
```

```
<ul>

   <li> List item 1 </li>

   <li> List item 2 </li>

   <li> List item 3 </li>

   <li> List item 4 </li>

   <li> List item 5 </li>

   <li> List item 6 </li>

   <li> List item 7 </li>

   <li> List item 8 </li>

</ul>
```

STRUCTURE PSEUDO CLASSES

```
:nth-child(2),
:nth-of-type(2) {
      color: red;
}
:nth-child(3n+1) {
      color: pink;
}
:nth-child(odd/even){…}
```



```
<ul>
   <li> List item 1 </li>
   <li> List item 2 </li>
   <li> List item 3 </li>
   <li> List item 4 </li>
   <li> List item 5 </li>
   <li> List item 6 </li>
   <li> List item 7 </li>
   <li> List item 8 </li>
</ul>
```

Example

STRUCTURE PSEUDO CLASSES

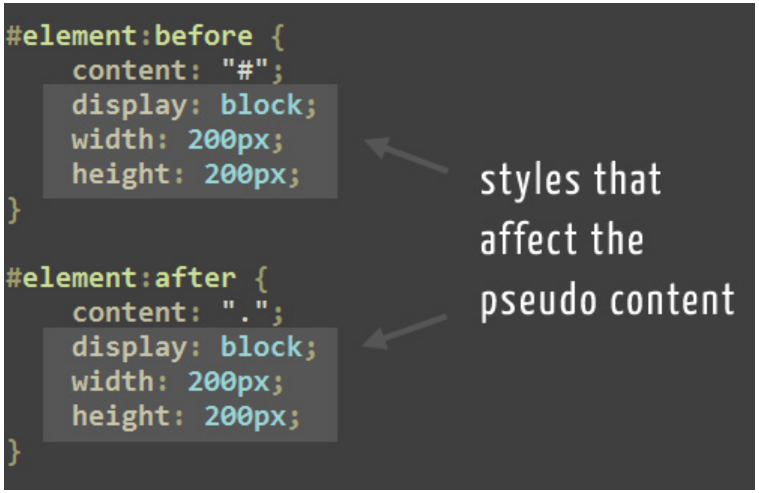
```
:empty { color: green }

:not(selector) { color: red }

:only-child { color: blue }
```

STATES PSEUDO CLASSES

```
input:checked { border-color: green }

input:enabled { background: green }

input:disabled { color: gray }
```

PSEUDO ELEMENTS

```
::after

::before

::first-letter

::first-line

::selection
```



```
#element:before {
    content: "#";
    display: block;
    width: 200px;
    height: 200px;
}

#element:after {
    content: ".";
    display: block;
    width: 200px;
    height: 200px;
}
```

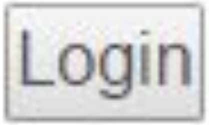
styles that affect the pseudo content

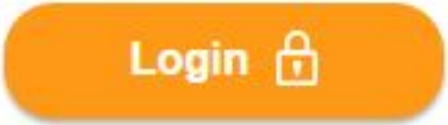
FORM ELEMENTS

➔ and how to style them

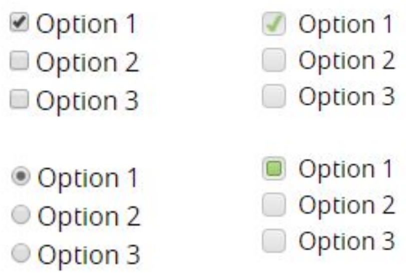
BUTTONS AND INPUTS

```
<input type="submit" value="Login">
<button type="submit">Login <img src=“lock.png”></button>

<input type="button" value="Login">
<button>Login <img src=“lock.png”></button>
```





CHECKBOXES AND RADIO BUTTONS



☑ Option 1
□ Option 2
□ Option 3

☑ Option 1
□ Option 2
□ Option 3

● Option 1
○ Option 2
○ Option 3

▣ Option 1
□ Option 2
□ Option 3

1) **<label> before <input>**
   input[type=radio] + label { … }
2) **hide input**
   input[type=radio] { opacity: 0; }
3) **::before**
   input[type=radio] + label::before { content: "\00a0"; }
4) **:checked**
   input[type=radio]:checked + label::before { … }

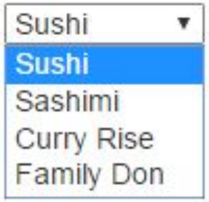
SELECT

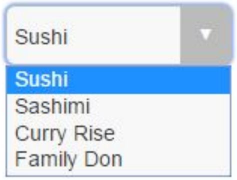
1) **<select>**

```
select { -webkit-appearance: none; }
```

2) **::after**

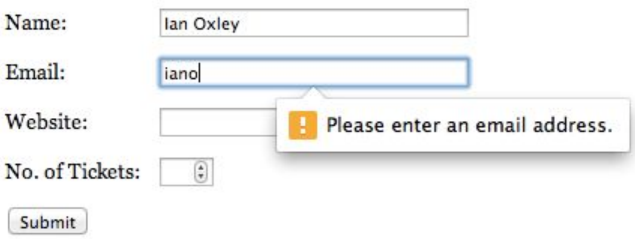
```
.select-wrap::after {
  content: "\25bc";
  position: absolute;
  pointer-events: none;
}
```





HTML5 VALIDATION

1. The '**required**' attribute (input)
2. Text INPUT types
3. The '**pattern**' attribute (input)
4. Pseudo classes:
    1. :required / :optional
    2. :valid / :invalid
    3. :in-range / :out-of-range
       (input[type=number])
    4. :read-only / :read-write
       (inputs with 'readonly' attribute)
5. The '**novalidate**' attribute (form)



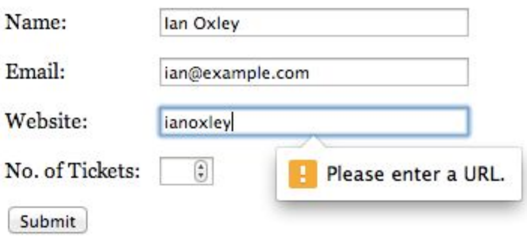
Name: Ian Oxley

Email: iano

Website:

No. of Tickets:

Please enter an email address.



Name: Ian Oxley

Email: ian@example.com

Website: ianoxley

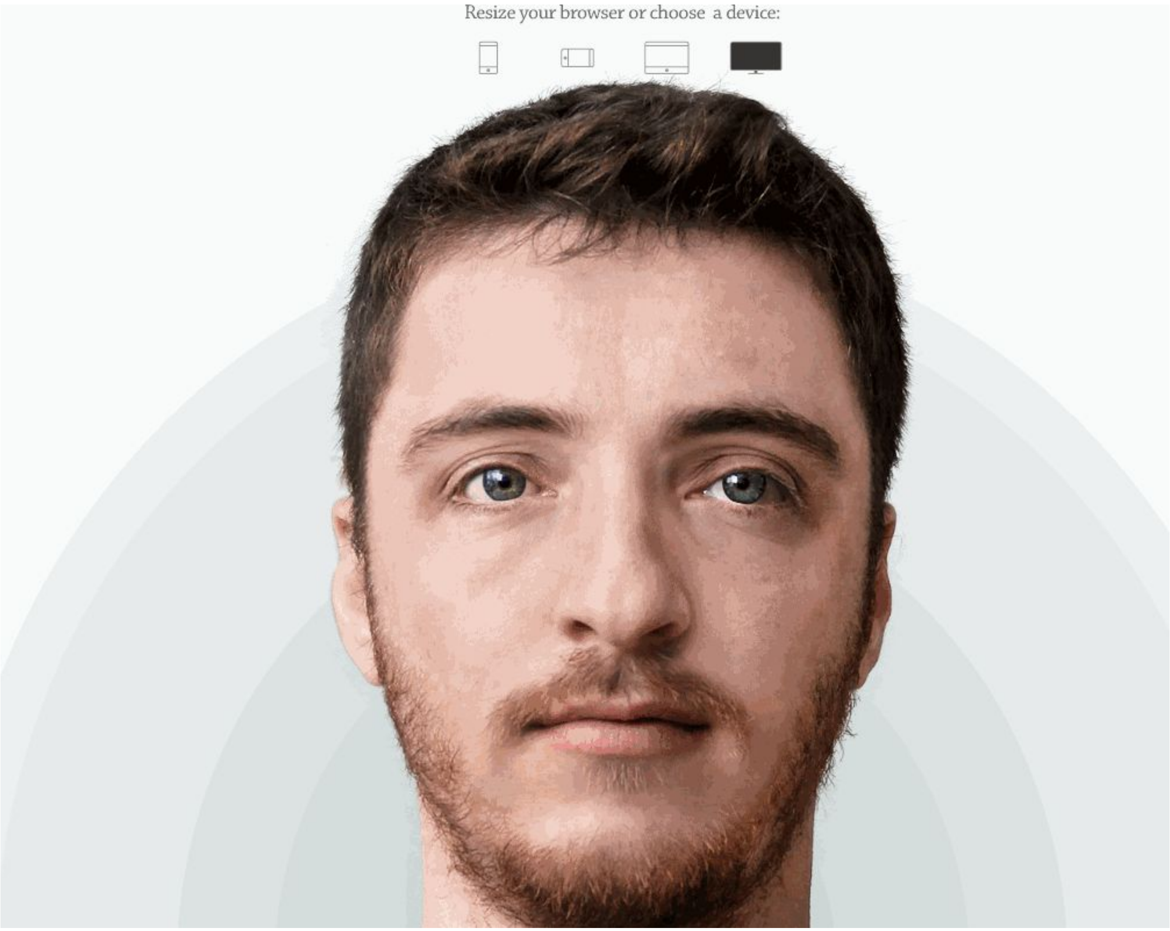
No. of Tickets:

Please enter a URL.

RESPONSIVE WEB DESIGN

➔ Bootstrap’s grid system

WHAT IS RESPONSIVE WEB DESIGN



Resize your browser or choose a device:

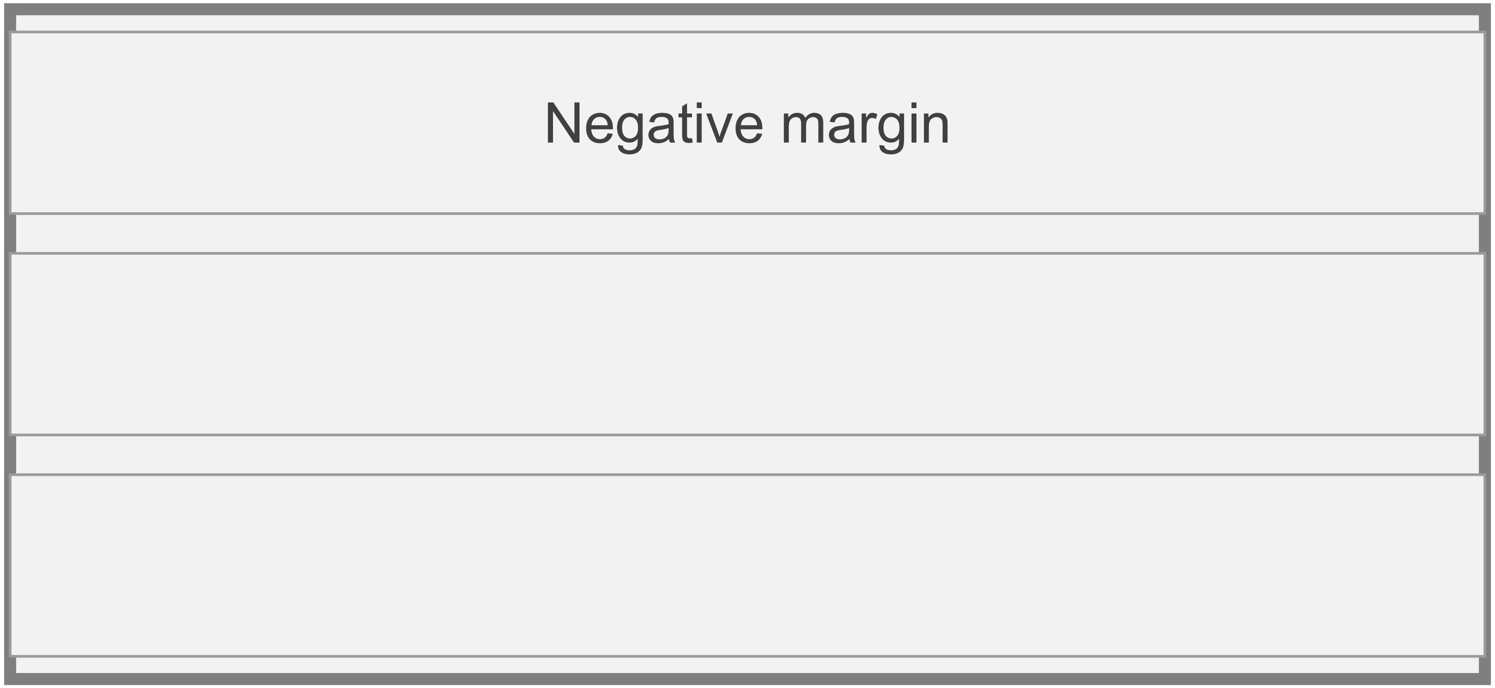
.CONTAINER / .CONTAINER-FLUID

Clearfix

Width

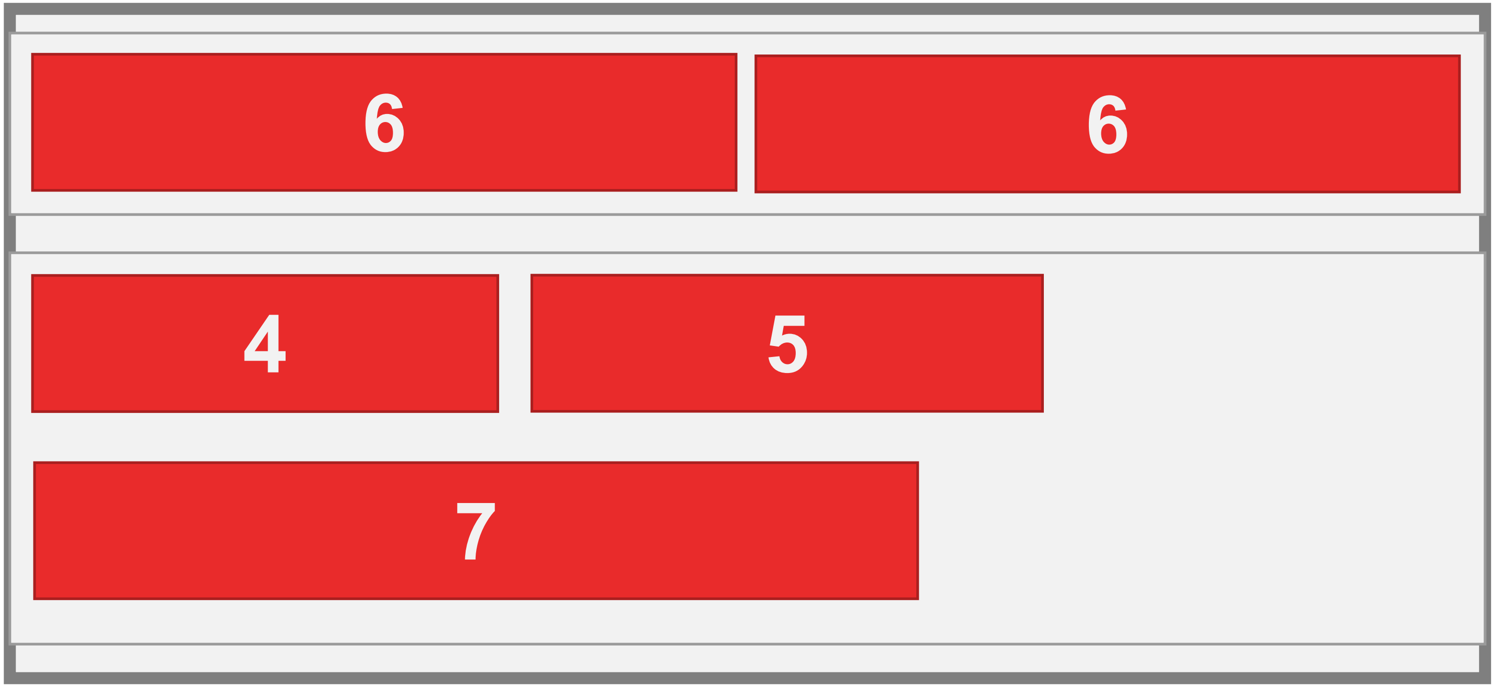
Paddings

.ROW



Negative margin

.COL-XS-* / .COL-SM-* / .COL-MD-* / .COL-LG-*

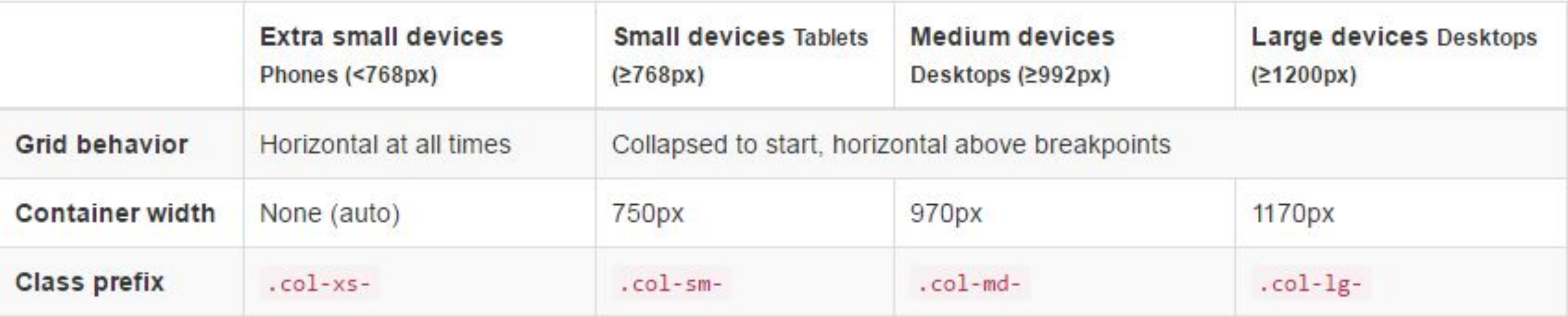


EXAMPLE

```
<div class="container">
  <div class="row">
    <div class="col-*-*"></div>
    <div class="col-*-*"></div>
  </div>
  <div class="row">
    <div class="col-*-*"></div>
    <div class="col-*-*"></div>
    <div class="col-*-*"></div>
  </div>
  <div class="row">
    ...
  </div>
</div>
```

@MEDIA

```
@media (min-width: 768px) {
    .item {
        float: none;
    }
}
```

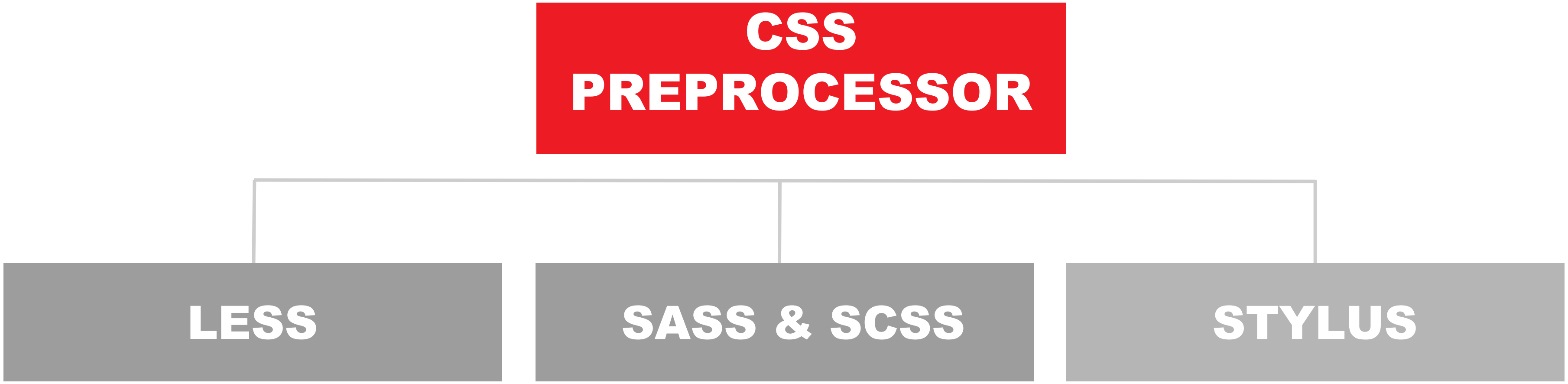


	Extra small devices Phones (<768px)	Small devices Tablets (≥768px)	Medium devices Desktops (≥992px)	Large devices Desktops (≥1200px)
Grid behavior	Horizontal at all times	Collapsed to start, horizontal above breakpoints		
Container width	None (auto)	750px	970px	1170px
Class prefix	`.col-xs-`	`.col-sm-`	`.col-md-`	`.col-lg-`

USEFUL TOOLS

➔ To speed up your development

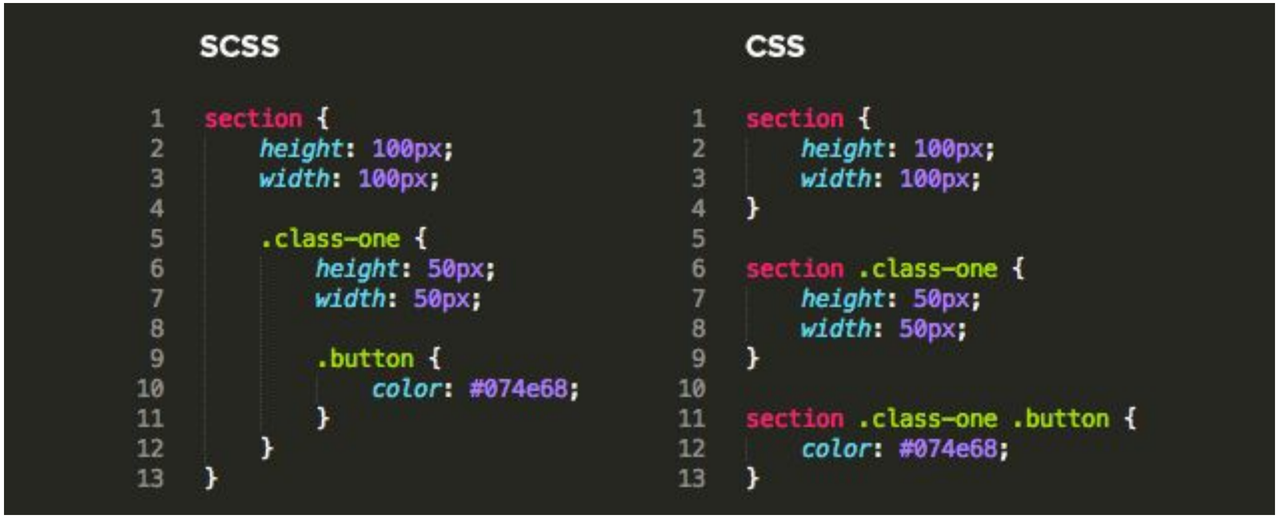
CSS PREPROCESSORS


CSS PREPROCESSOR

LESS

SASS & SCSS

STYLUS


SCSS

```
section {
    height: 100px;
    width: 100px;

    .class-one {
        height: 50px;
        width: 50px;

        .button {
            color: #074e68;
        }
    }
}
```

CSS

```
section {
    height: 100px;
    width: 100px;
}

section .class-one {
    height: 50px;
    width: 50px;
}

section .class-one .button {
    color: #074e68;
}
```

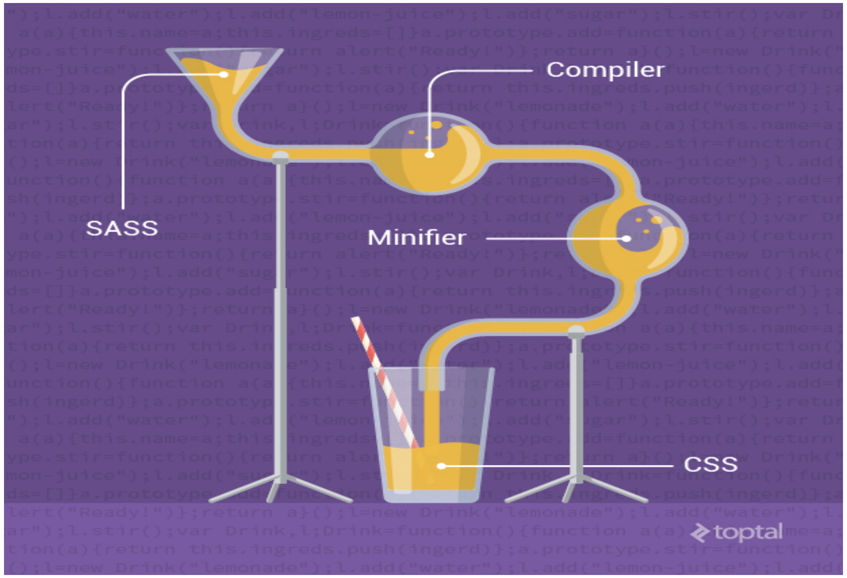
BUILD SYSTEM AUTOMATING TASKS

BUILD SYSTEMS

WEBPACK

GULP

GRUNT



CHROME EXTENSIONS

MB Ruler / Page Ruler

PerfectPixel

Yslow

LiveReload

PREPROCESSORS: VARIABLES

```
$font-stack: Helvetica;
$primary-color: #333;

body {
    font: 100% $font-stack;
    color: $primary-color;
}
```

→

```
body {
    font: 100% Helvetica;
    color: #333;
}
```

```
@width: 10px;
@height: @width + 10px;

#header {
  width: @width;
  height: @height;
}
```

→

```
#header {
  width:
10px;
  height:
20px;
}
```

PREPROCESSORS: NESTING

```
#header {
    color: black;

    .navigation { font-size: 12px; }
    .logo { width: 300px; }
}
```

→

```
#header { color: black; }
#header .navigation { font-size:
12px; }
#header .logo { width: 300px; }
```

```
#header {
    color: black;

    &:hover { color: red; }
}
```

→

```
#header { color: black; }
#header:hover { color:
red; }
```

PREPROCESSORS: NESTING @-RULES

```
.component {
    width: 300px;

    @media (min-width: 768px) {
        width: 600px;
    }

    @media (min-width: 1280px) {
        width: 800px;
    }
}
```

→

```
.component { width: 300px; }

@media (min-width: 768px) {
    .component {
        width: 600px;
    }
}

@media (min-width: 1280px) {
    .component {
        width: 800px;
    }
}
```

PREPROCESSORS: OTHER FEATURES

- Functions
- Operations
- Importing (@import)
- Mixins (@mixin)
- Extend(% and @extend)

POSTCSS

A tool for transforming CSS
with JavaScript

CSS METHODOLOGIES

➔ BEM

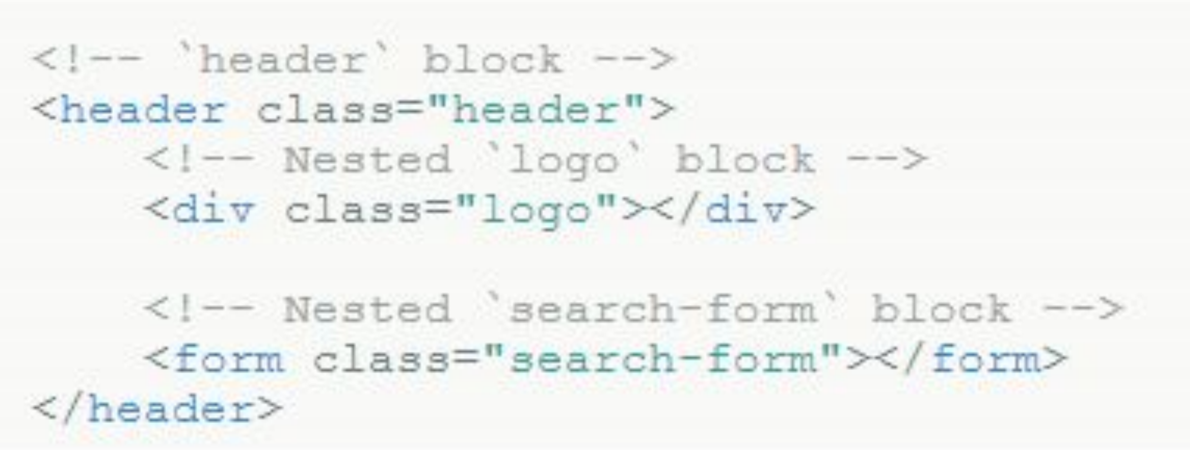
BEM (BLOCK, ELEMENT, MODIFIER)

- **Block - a functionally independent page component that can be reused.**

- **The block name describes its purpose.**

- **The block shouldn't influence its environment.**

- **You also shouldn't use CSS tag or ID selectors when using BEM.**



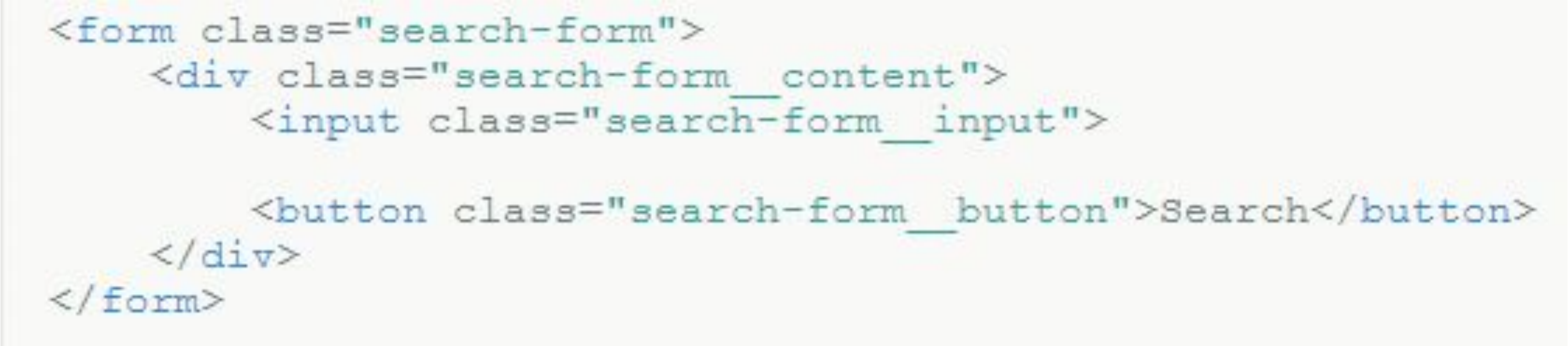
```
<!-- `header` block -->
<header class="header">
    <!-- Nested `logo` block -->
    <div class="logo"></div>

    <!-- Nested `search-form` block -->
    <form class="search-form"></form>
</header>
```

BEM (BLOCK, ELEMENT, MODIFIER)

- Element - a composite part of a block that can't be used separately from it.

- The element name describes its purpose.

- The structure of an element's full name is block-name__element-name.



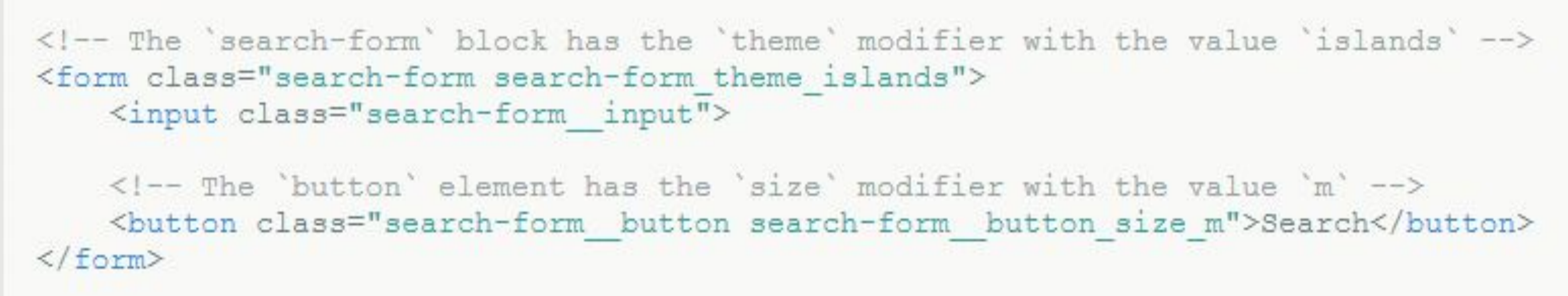
```
<form class="search-form">
    <div class="search-form__content">
        <input class="search-form__input">

        <button class="search-form__button">Search</button>
    </div>
</form>
```

BEM (BLOCK, ELEMENT, MODIFIER)

- **Modifier - an entity that defines the appearance, state, or behavior of a block or element.**

- **The modifier name describes its appearance.**

- **The modifier name is separated from the block or element name by a single underscore (_).**



```
<!-- The `search-form` block has the `theme` modifier with the value `islands` -->
<form class="search-form search-form_theme_islands">
    <input class="search-form__input">

    <!-- The `button` element has the `size` modifier with the value `m` -->
    <button class="search-form__button search-form__button_size_m">Search</button>
</form>
```

BEM (BLOCK, ELEMENT, MODIFIER)

```
.articles {
  &__filter-container { }

   &__container { }

  &__loader-container {
    &_from-bottom {}

    &_from-top {}
  }

  &__footer {}
}
```

```
<section class="articles">
  <section class="articles__filter-container">
    <exp-filter class="articles__filter">
  </section>
  <section class="articles__container">
    <section
      class="articles__loader-container"
      [class.articles__loader-container_from-bottom]="fromButtom“
      [class.articles__loader-container_from-top]="fromTop“
    ></section>
  </section>
  <footer class="articles__footer"></footer>
</section>
```

USEFUL LINKS

http://caniuse.com/
http://css-tricks.com/
https://github.com/hail2u/html-best-practices
https://css-tricks.com/autoprefixer/
https://css-tricks.com/snippets/css/complete-guide-grid/

https://htmlacademy.ru/courses/44

https://slides.com/fasvald/html-css/#/
https://slides.com/fasvald/css-methodologies#/

THANK YOU FOR YOUR ATTENTION!

iTechArt