



JONAS.IO
SCHMEDTMANN

ADVANCED CSS AND SASS

TAKE YOUR CSS TO THE NEXT LEVEL!



@JONASSCHMEDTMAN

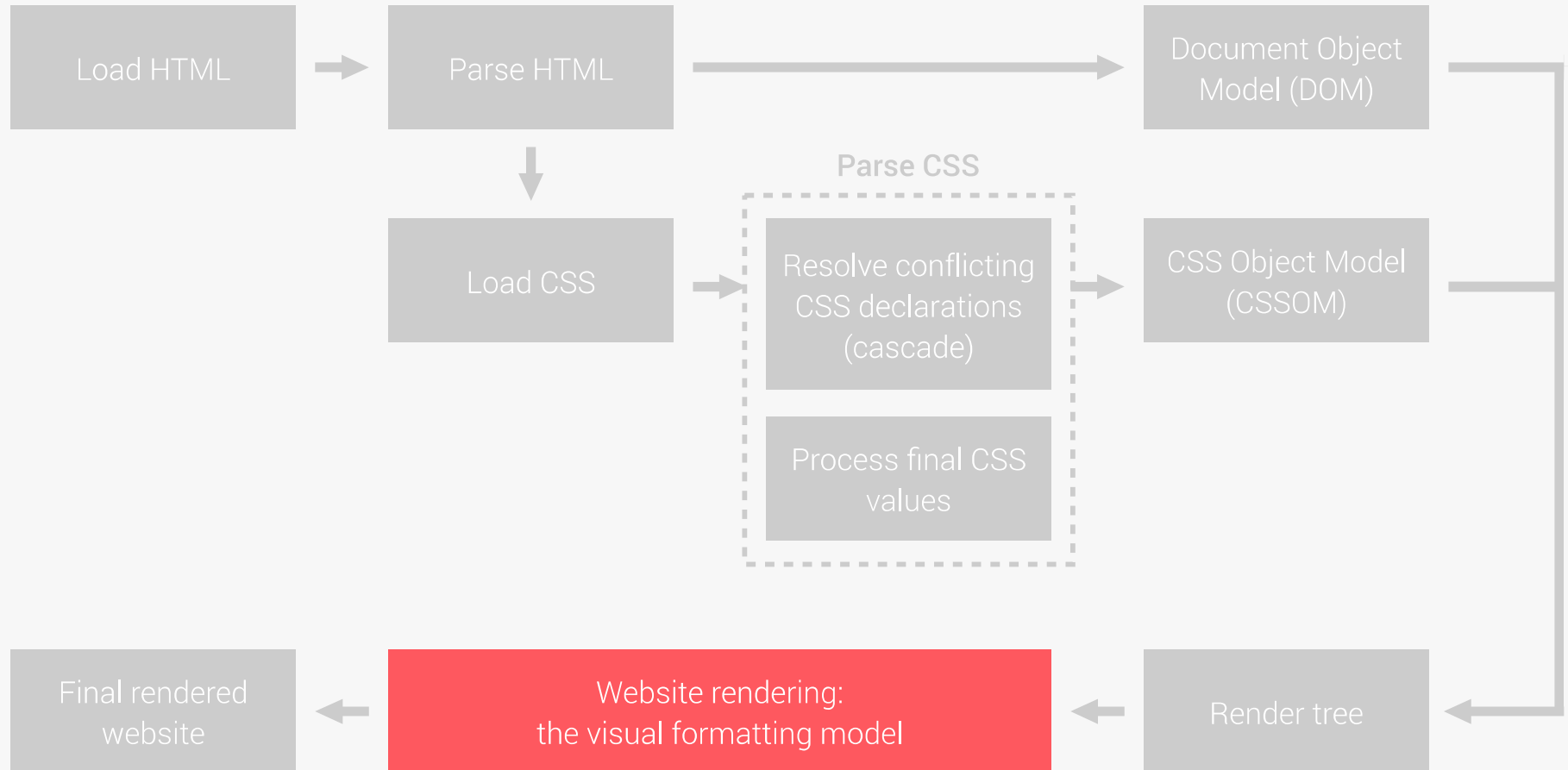
SECTION

HOW CSS WORKS: A LOOK BEHIND THE SCENES

LECTURE

HOW CSS RENDERS A WEBSITE: THE VISUAL FORMATTING MODEL

REMEMBER...?

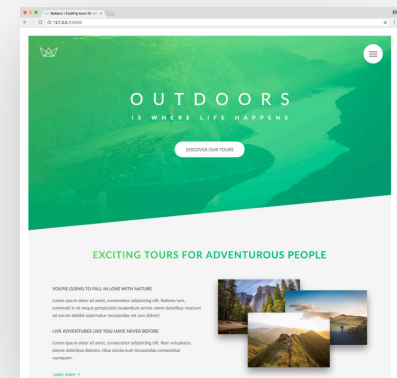
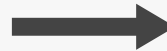


THE VISUAL FORMATTING MODEL

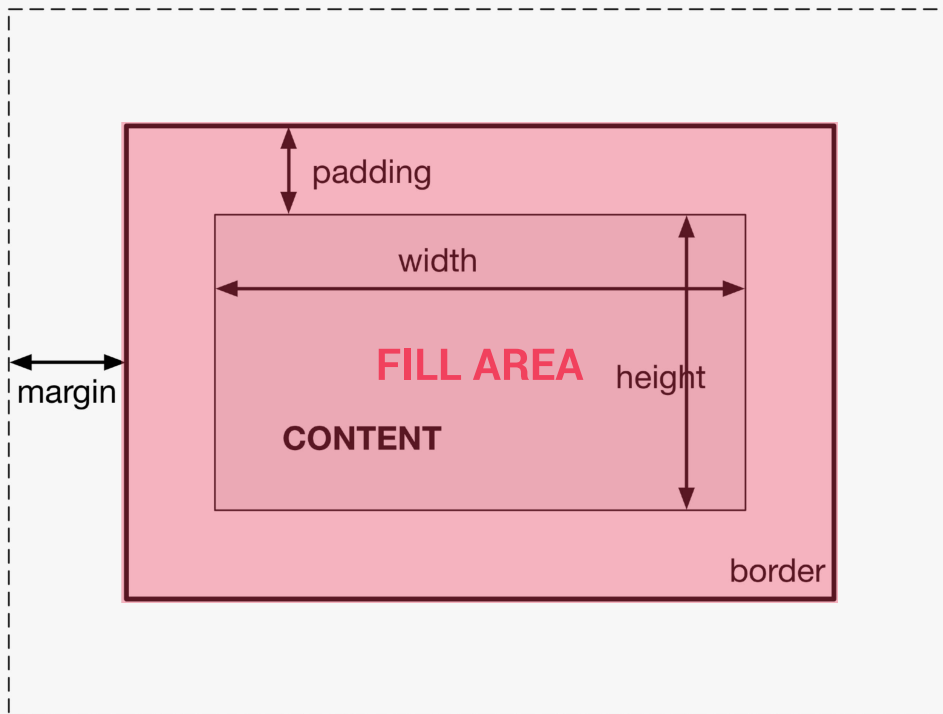
DEFINITION

Algorithm that calculates boxes and determines the layout of these boxes, for each element in the render tree, in order to determine the final layout of the page.

- **Dimensions of boxes:** the box model;
- **Box type:** inline, block and inline-block;
- **Positioning scheme:** floats and positioning;
- **Stacking contexts;**
- Other elements in the render tree;
- Viewport size, dimensions of images, etc.



1. THE BOX MODEL



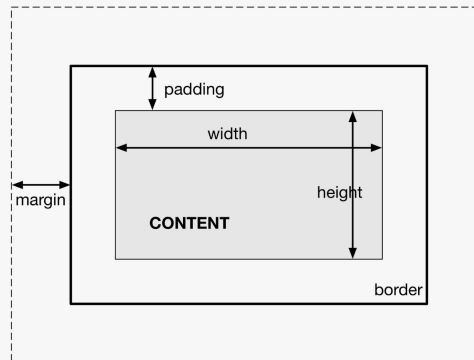
- **Content:** text, images, etc;
- **Padding:** transparent area around the content, inside of the box;
- **Border:** goes around the padding and the content;
- **Margin:** space between boxes;
- **Fill area:** area that gets filled with background color or background image.

1. THE BOX MODEL: HEIGHTS AND WIDTHS

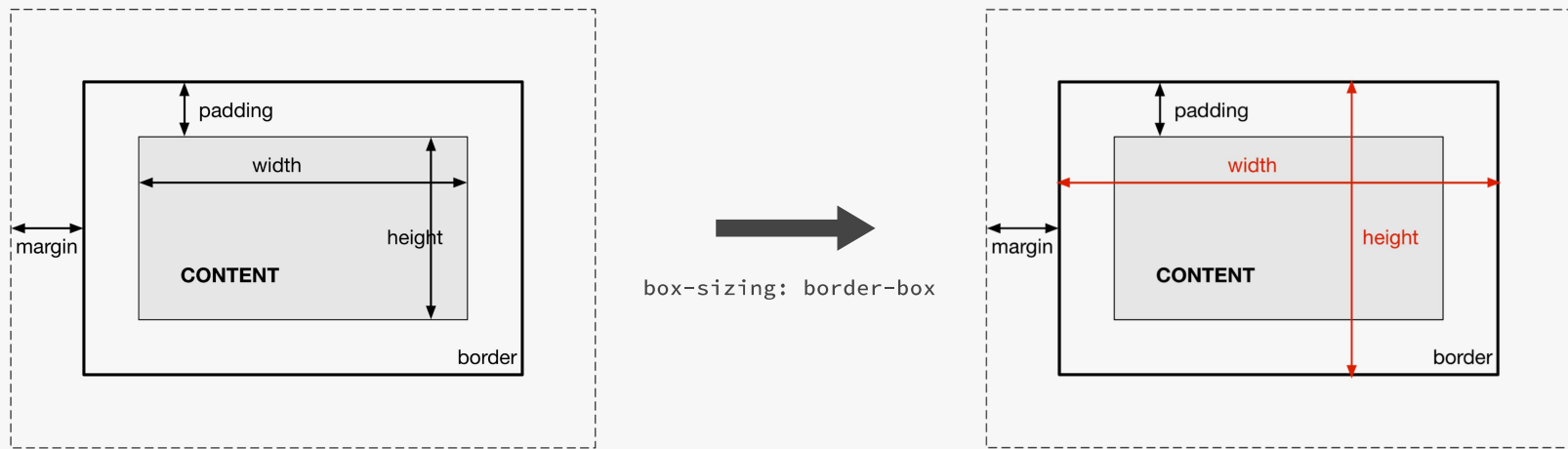
total width = right border + right padding + specified width + left padding + left border

total height = top border + top padding + specified height + bottom padding + bottom border

Example: $\text{height} = 0 + 20\text{px} + 100\text{px} + 20\text{px} + 0 = 140\text{px}$



1. THE BOX MODEL WITH BOX-SIZING: BORDER-BOX



total width = ~~right border~~ + ~~right padding~~ + specified width + ~~left padding~~ + ~~left border~~

total height = ~~top border~~ + ~~top padding~~ + specified height + ~~bottom padding~~ + ~~bottom border~~

Example: height = 0 + ~~20px~~ + 100px + ~~20px~~ + 0 = 100px

2. BOX TYPES: INLINE, BLOCK-LEVEL AND INLINE-BLOCK

Block-level boxes

- Elements formatted visually as blocks
- 100% of parent's width
- Vertically, one after another
- Box-model applies as showed

display: block

(display: flex)
(display: list-item)
(display: table)

Inline-block boxes

- A mix of block and inline
- Occupies only content's space
- No line-breaks
- Box-model applies as showed

display: inline-block

Inline boxes

- Content is distributed in lines
- Occupies only content's space
- No line-breaks
- No heights and widths
- Paddings and margins only horizontal (left and right)

display: inline

3. POSITIONING SCHEMES: NORMAL FLOW, ABSOLUTE POSITIONING AND FLOATS

Normal flow

- Default positioning scheme;
- **NOT** floated;
- **NOT** absolutely positioned;
- Elements laid out according to their source order.

Default
`position: relative`

Floats

- **Element is removed from the normal flow;**
- Text and inline elements will wrap around the floated element;
- The container will not adjust its height to the element.

`float: left`
`float: right`

=

≠

Absolute positioning

- **Element is removed from the normal flow**
- No impact on surrounding content or elements;
- We use top, bottom, left and right to offset the element from its relatively positioned container.

`position: absolute`
`position: fixed`

4. STACKING CONTEXTS

