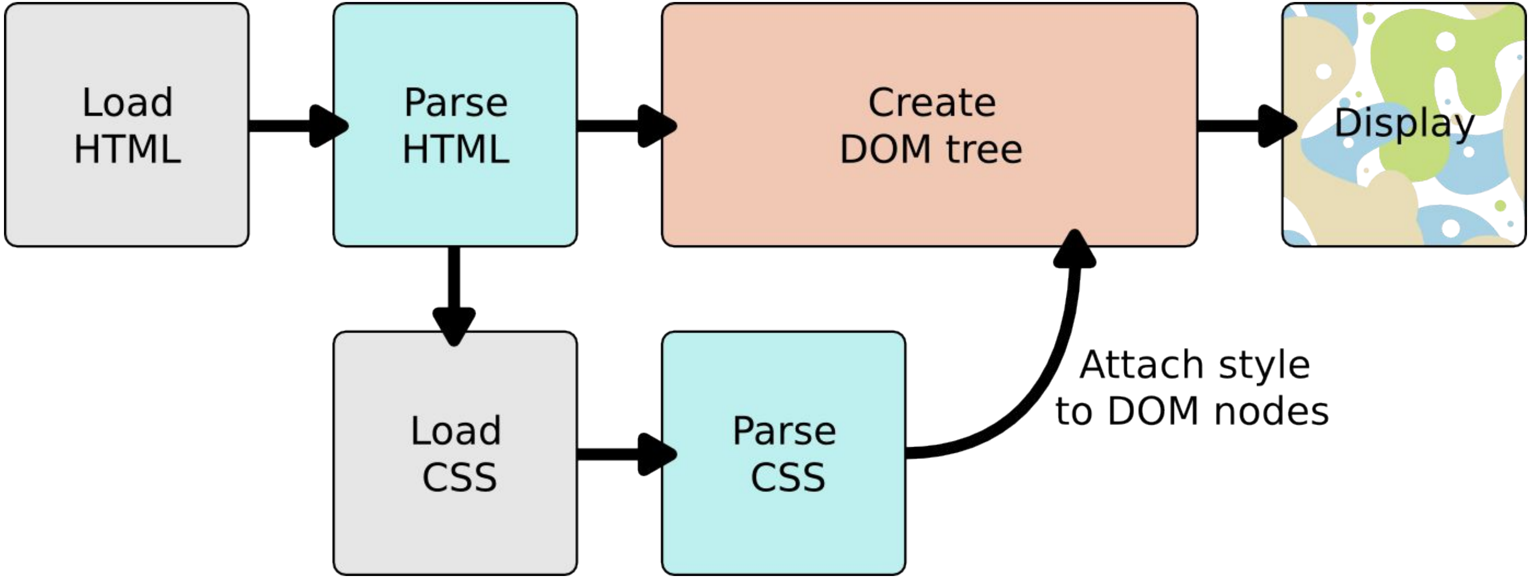
# Web Technologies & Image Processing

Jean-Christophe Taveau

2016-2017

# CSS: Introduction



# CSS: Introduction

## **CSS**: **C**ascading **S**tyle **S**heets

→ Used to set the visual style of web pages & user interfaces written in HTML

→ How to set style to HTML elements?

  → *via* an attribute of a HTML element.

  → *via* the <style> element in <head>

  → *via* an external file with <link> element in <head>

```
<p style=”color: red”>My paragraph</p>
```

```
<style>
p {
    color : red;
}
</style>
```

```
<link href="path/to/file.css" rel="stylesheet">
```

# CSS: Introduction

**CSS**: **C**ascading **S**tyle **S**heets

→ CSS declaration

→ CSS declaration block

→ CSS rule

→ Selectors

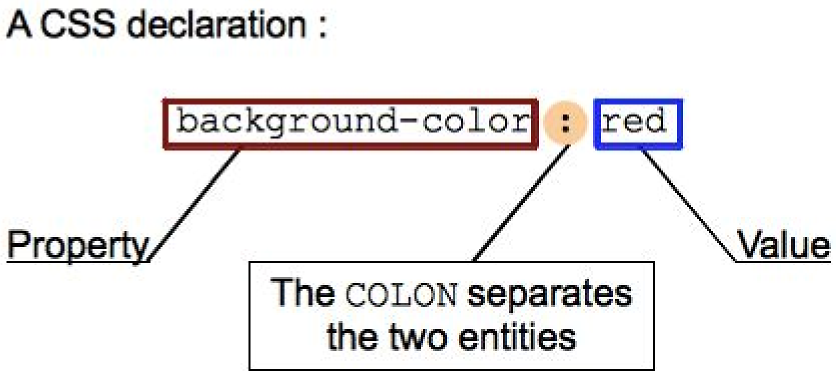
    → Simple

    → Class

    → ID

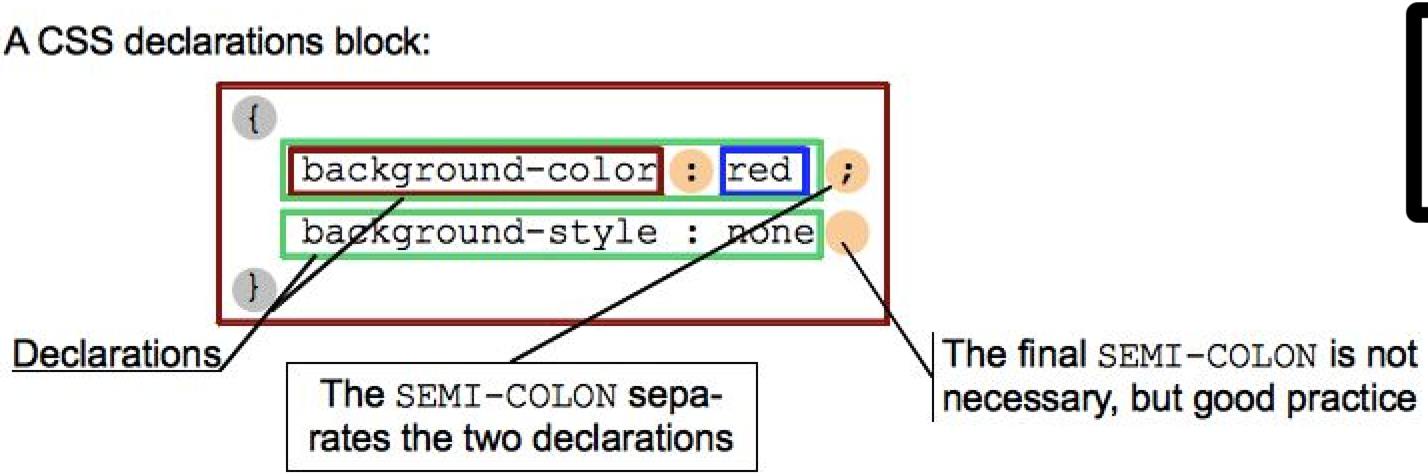
# CSS: vocabulary

CSS declaration: pair **property/value**



*Source: https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS*

# CSS: Vocabulary


A CSS declarations block:

```
{
  background-color : red ;
  background-style : none
}
```

Declarations

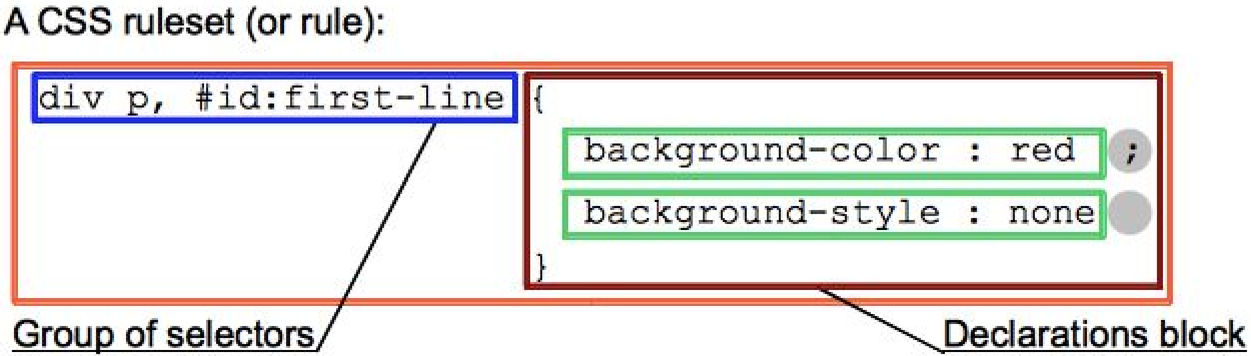
The SEMI-COLON separates the two declarations

The final SEMI-COLON is not necessary, but good practice

**block:** Declarations wrapped by an opening curly brace, ({) and a closing one (}).

*Source: https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS*

# CSS: Vocabulary



A CSS ruleset (or rule):

```
div p, #id:first-line {
    background-color : red ;
    background-style : none
}
```

**Ruleset (rule):** Each declaration block with a **selector.**
**Selector**: pattern that matches some elements on the page.

*Source: https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS*

# CSS: Selectors

## **CSS**: **C**ascading **S**tyle **S**heets

→ Selectors

→ Simple

→ Class: `.classname`

→ ID : `#idname`

```
<p>My text</p>
<p class=”one”>This text</p>
<p id=”two”>Lorem ipsum</p>
```

```
p {
    color : red;
}

.one {
    color: green;
}

#two {
    color: blue;
}
```

# CSS: Selectors

## **CSS**: **C**ascading **S**tyle **S**heets

→ Selectors

→ Pseudo-classes - Syntax :keyword

→ Style the selected elements *only* when they are in certain state

```
<p>My text</p>
```

```
p {
    color : red;
}

p:hover {
    background-color: green;
}
```

# CSS: Selectors

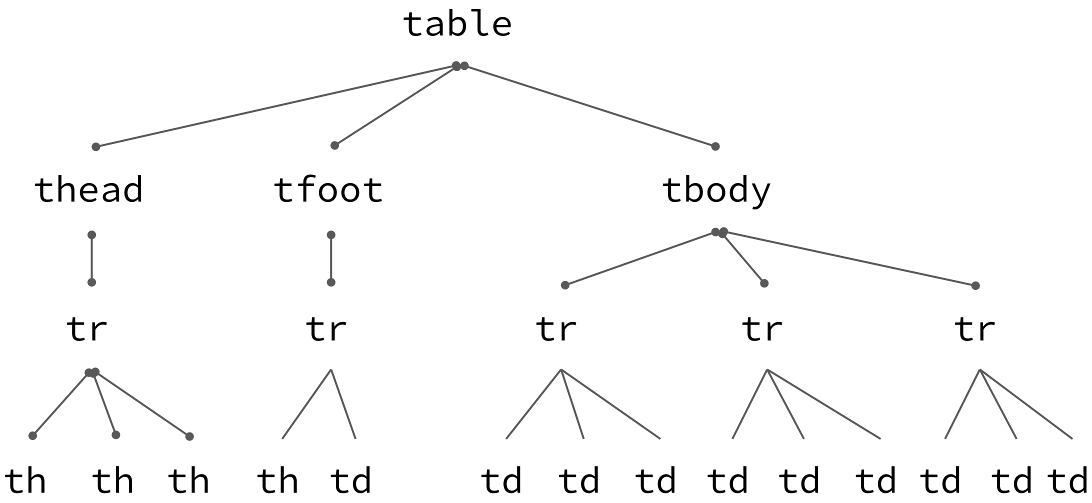
## Combinators

CSS selectors become even more useful when you start combining them to perform fine-grained selections. CSS has several ways to select elements based on how they are related to one another.

| Combinators | Select |
|---|---|
| AB | Any element matching both A and B at the same time. |
| A B | Any element matching B that is a *descendant* of an element matching A<br>⟶ that is: a child, or a child of a child, *etc*. |
| A > B | Any element matching B that is a *direct child* of an element matching A. |
| A + B | Any element matching B that is the next *sibling* of an element matching A<br>⟶ that is: the next child of the same parent. |
| A ~ B | Any element matching B that is among the next *sibling* of an element matching A<br>⟶ that is: one of the next children of the same parent. |

*Source: https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/Combinators_and_multiple_selectors*

# CSS: Selectors - Combinators

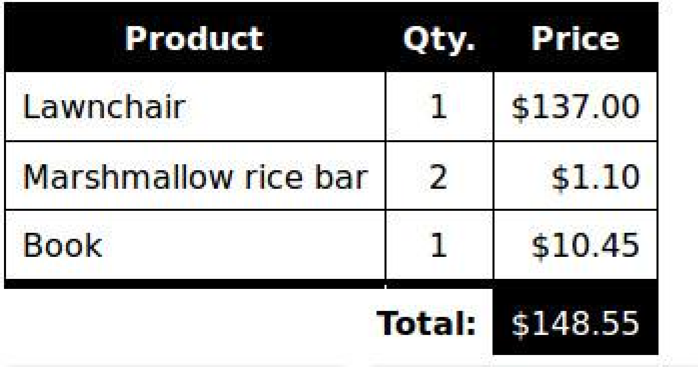


table

thead tfoot tbody

tr tr tr tr tr

th th th th td td td td td td td td td td



| Product | Qty. | Price |
|---|---|---|
| Lawnchair | 1 | $137.00 |
| Marshmallow rice bar | 2 | $1.10 |
| Book | 1 | $10.45 |
|  | Total: | $148.55 |

*Source: https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/Combinators_and_multiple_selectors*

# CSS: Selectors - Combinators

```
/* All <td>s within a <table and all <th>s within a <table> */
table td, table th {}

/* All <th>s within <thead>s that are within <table>s */
table thead th {}

/* All <td>s preceded by another <td>,within a <tbody>, within a <table> */
table tbody td + td {}

/* All <td>s that are a last child, within a <tbody>, within a <table> */
table tbody td:last-child {}

/* All <th>s, within a <tfoot>s, within a <table> */
table tfoot th {}

/* All <td>s preceded by a <th>, within a <table> */
table th + td {}

/* All pseudo-elements "before" <td>s that are a last child,appearing within elements with a class of "with-currency" that also have an
attribute "lang" with the value "en-US" */
.with-currency[lang="en-US"] td:last-child::before {
  content: '$';
}

/* All pseudo-elements "after" <td>s that are a last child, appearing within elements with the class "with-currency" that also have an attribute
"lang" with the value "fr" */
.with-currency[lang="fr"] td:last-child::after {
  content: ' €';
}
```

# CSS: Cascade - Inheritance

**CSS**: **C**ascading **S**tyle **S**heets

→ Cascade (or inheritance) depends of:

→ Importance (don’t use it - difficult to debug)

→ Specificity

→ Source order

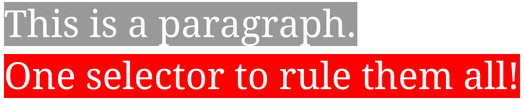
# CSS: Cascade - Inheritance

## **CSS**: **C**ascading **S**tyle **S**heets

→ Cascade (or inheritance) depends of:

→ Importance (don’t use it - difficult to debug)

```
<p class="better">This is a paragraph.</p>
<p class="better" id="winning">One selector to rule them
all!</p>
```



This is a paragraph.

One selector to rule them all!

```
#winning {
  background-color: red;
  border: 1px solid black;
}

.better {
  background-color: gray;
  border: none !important;
}

p {
  background-color: blue;
  color: white;
  padding: 5px;
}
```

# CSS: Cascade - Inheritance

**CSS**: **C**ascading **S**tyle **S**heets

→ Cascade (or inheritance) depends of:

→ Specificity

simple < class < ID

# CSS: Cascade - Inheritance

## **CSS**: **C**ascading **S**tyle **S**heets

→ Cascade (or inheritance) depends of:

→ Source order

```
p {
  color: blue;
}

/* This rule will win over the first one */
p {
  color: red;
}
```

# CSS: Cascade - Inheritance

## **CSS**: **C**ascading **S**tyle **S**heets

→ Rule mixing

  → Source order

```
<p>I'm <strong>important</strong></p>
```

```
/* weight: 0002 */
p strong {
  background-color: khaki;
  color: green;
}

/* weight: 0001 */
strong {
  text-decoration: underline;
  color: red;
}
```

# CSS: Cascade - Inheritance

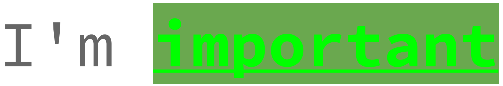
**CSS**: **C**ascading **S**tyle **S**heets

→ Rule mixing

- → Importance
- → Specificity
- → Source order

```
/* weight: 0002 */
p strong {
  background-color: khaki;
  color: green;
}

/* weight: 0001 */
strong {
  text-decoration: underline;
  color: red;
}
```

```
<p>I'm <strong>important</strong></p>
```



I'm **important**

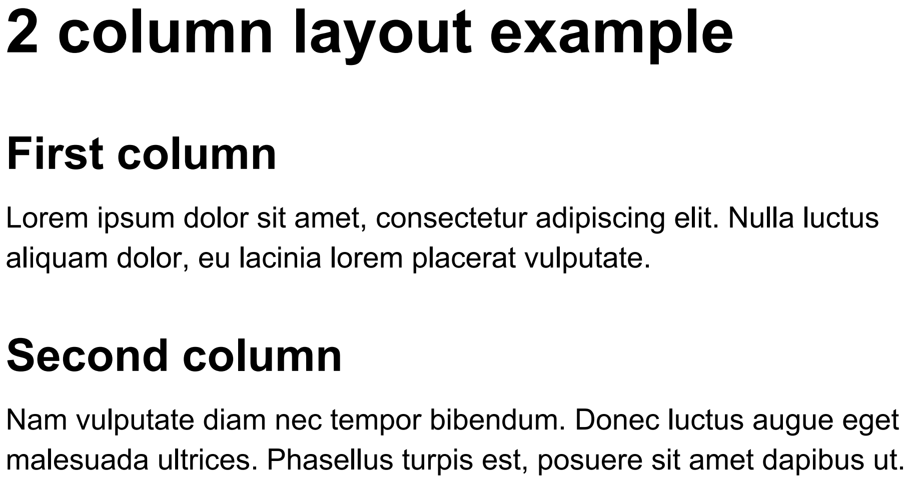
‘important’ is bolded because it is the default style of *strong* in web browser.

# CSS: STYLING

- → Text styling
    - → Fonts: family, size, weight
    - → Text: shadows, alignment, height, letter and word spacing
    - → Lists
    - → Links
- → Boxes
    - → Background
    - → Border: type, color, thickness, corner
    - → Padding, margins

# CSS: STYLING

- → CSS layout
  - → Positioning: absolute, relative
  - → Float: left, right, none



# 2 column layout example

## First column

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, eu lacinia lorem placerat vulputate.

## Second column

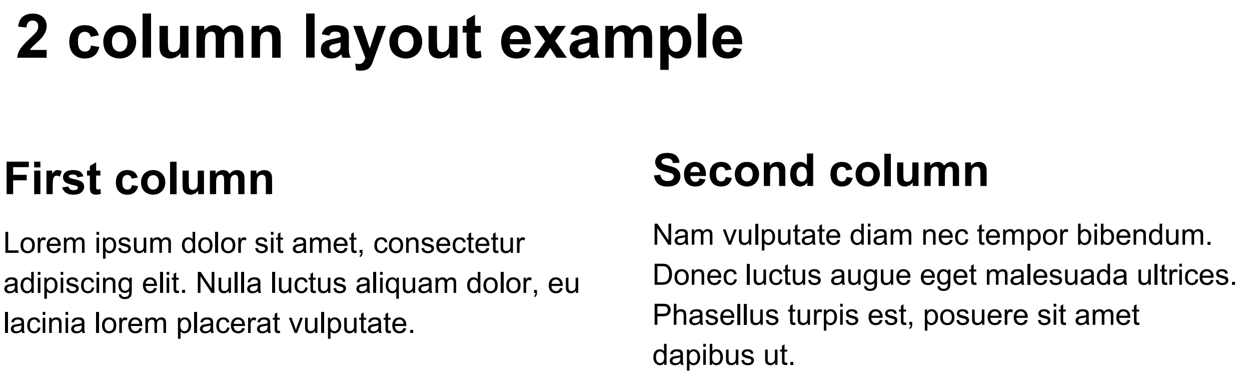
Nam vulputate diam nec tempor bibendum. Donec luctus augue eget malesuada ultrices. Phasellus turpis est, posuere sit amet dapibus ut.

```
<h1>2 column layout example</h1>
<div>
  <h2>First column</h2>
  <p> Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Nulla luctus aliquam dolor, eu
lacinia lorem placerat vulputate. </p>
</div>

<div>
  <h2>Second column</h2>
  <p>Nam vulputate diam nec tempor bibendum. Donec
luctus augue eget malesuada ultrices. Phasellus
turpis est, posuere sit amet dapibus ut.</p>
</div>
```

# CSS: Styling

- → CSS layout
  - → Positioning: absolute, relative
  - → Float: left, right, none

```
div:nth-of-type(1) {
  width: 48%;
  float: left;
}

div:nth-of-type(2) {
  width: 48%;
  float: right;
}
```



# 2 column layout example

## First column

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, eu lacinia lorem placerat vulputate.

## Second column

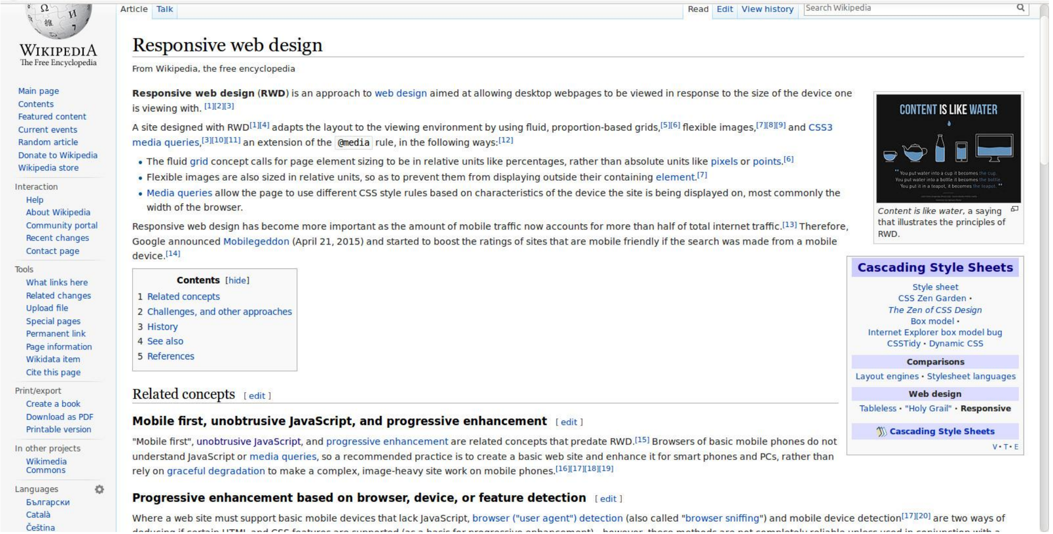
Nam vulputate diam nec tempor bibendum. Donec luctus augue eget malesuada ultrices. Phasellus turpis est, posuere sit amet dapibus ut.

# CSS: responsive Design

Responsive design

→ CSS ruleset for given screen resolutions

```
@media (max-width: 600px) {
  .facet_sidebar {
    display: none;
  }
}
```



WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

In other projects
Wikimedia Commons

Languages
Български
Català
Čeština

Article Talk Read Edit View history Search Wikipedia

## Responsive web design

From Wikipedia, the free encyclopedia

**Responsive web design** (**RWD**) is an approach to web design aimed at allowing desktop webpages to be viewed in response to the size of the device one is viewing with. [1][2][3]

A site designed with RWD[1][4] adapts the layout to the viewing environment by using fluid, proportion-based grids,[5][6] flexible images,[7][8][9] and CSS3 media queries,[3][10][11] an extension of the `@media` rule, in the following ways:[12]

- The fluid grid concept calls for page element sizing to be in relative units like percentages, rather than absolute units like pixels or points.[6]
- Flexible images are also sized in relative units, so as to prevent them from displaying outside their containing element.[7]
- Media queries allow the page to use different CSS style rules based on characteristics of the device the site is being displayed on, most commonly the width of the browser.

Responsive web design has become more important as the amount of mobile traffic now accounts for more than half of total internet traffic.[13] Therefore, Google announced Mobilegeddon (April 21, 2015) and started to boost the ratings of sites that are mobile friendly if the search was made from a mobile device.[14]

CONTENT IS LIKE WATER
"You put water into a cup it becomes the cup.
You put water into a bottle it becomes the bottle.
You put it in a teapot, it becomes the teapot."

*Content is like water*, a saying that illustrates the principles of RWD.

**Cascading Style Sheets**
Style sheet
CSS Zen Garden •
*The Zen of CSS Design*
Box model •
Internet Explorer box model bug
CSSTidy • Dynamic CSS
**Comparisons**
Layout engines • Stylesheet languages
**Web design**
Tableless • "Holy Grail" • **Responsive**
**Cascading Style Sheets**
V • T • E

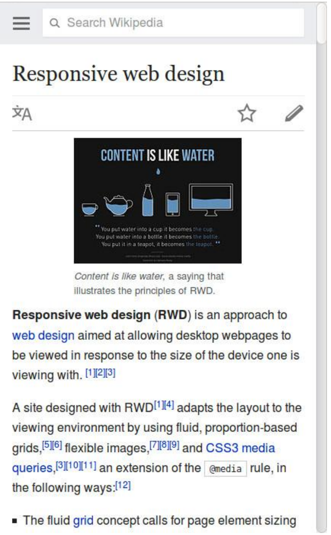
### Related concepts [ edit ]

#### Mobile first, unobtrusive JavaScript, and progressive enhancement [ edit ]

"Mobile first", unobtrusive JavaScript, and progressive enhancement are related concepts that predate RWD.[15] Browsers of basic mobile phones do not understand JavaScript or media queries, so a recommended practice is to create a basic web site and enhance it for smart phones and PCs, rather than rely on graceful degradation to make a complex, image-heavy site work on mobile phones.[16][17][18][19]

#### Progressive enhancement based on browser, device, or feature detection [ edit ]

Where a web site must support basic mobile devices that lack JavaScript, browser ("user agent") detection (also called "browser sniffing") and mobile device detection[17][20] are two ways of



Search Wikipedia

## Responsive web design

CONTENT IS LIKE WATER
"You put water into a cup it becomes the cup.
You put water into a bottle it becomes the bottle.
You put it in a teapot, it becomes the teapot."

*Content is like water*, a saying that illustrates the principles of RWD.

**Responsive web design** (**RWD**) is an approach to web design aimed at allowing desktop webpages to be viewed in response to the size of the device one is viewing with. [1][2][3]

A site designed with RWD[1][4] adapts the layout to the viewing environment by using fluid, proportion-based grids,[5][6] flexible images,[7][8][9] and CSS3 media queries,[3][10][11] an extension of the `@media` rule, in the following ways:[12]

- The fluid grid concept calls for page element sizing