CSS Level 3

Laura Farinetti

Dipartimento di Automatica e Informatica

Politecnico di Torino

laura.farinetti@polito.it

CSS level 3 (CSS3)

- Major change: introduction of modules
- Advantage of modules: they (supposedly) allow the specification to be completed and approved more quickly, because segments are completed and approved in chunks
    - This also allows browser and user-agent manufacturers to support sections of the specification but keep their code bloat to a minimum by only supporting those modules that make sense

CSS3: new features

- Selectors
- Color and opacity
- Multiple backgrounds
- Advanced layout: multi-column, grid
- Text: word wrap, shadow, @font-face
- Box and borders: radius, image, shadow
- Transformations
- Media queries
- Speech

http://www.css3.info/
http://www.w3.org/TR/#tr_CSS

Selectors

- Allow to target specific HTML elements without having to rely on unnecessary classes, IDs and JavaScripts
  - They can reduce the number of classes and IDs in the markup and make it easier for designers to maintain a style sheet

Selectors

- Attribute selectors: three new types
  - [att^="value"]
    Matches elements to an attribute that starts with the specified value
  - [att$="value"]
    Matches elements to an attribute that ends with the specified value
  - [att*="value"]
    Matches elements to an attribute that contains the specified value
- Example:

```
a[href$=".pdf"]
a[href^="http://"]
```

  - Selects all links to a PDF file

Selectors

- Combinators: one new type
  - General sibling selector: it targets all siblings of an element that have the same parent
- Example:

```
div~img
{ border: 1px solid #ccc; }
```

  - add a gray border to all images that are a sibling of a particular div (both the div and the images should have the same parent)

Selectors

- Pseudo-Classes
  - :nth-child(N)
    matches elements on the basis of their positions within a parent element's list of child elements
  - :nth-last-child(N)
    matches elements on the basis of their positions within a parent element's list of child elements
  - :nth-of-type(N)
    matches elements on the basis of their positions within a parent element's list of child elements of the same type
  - :nth-last-of-type(N)
    matches elements on the basis of their positions within a parent element's list of child elements of the same type
  - :last-child
    matches an element that's the last child element of its parent element
  - :first-of-type
    matches the first child element of the specified element type
  - :last-of-type
    matches the last child element of the specified element type

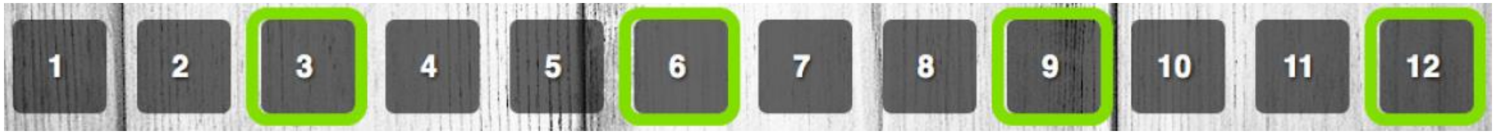
Selectors

- Pseudo-Classes
  - :only-child
    matches an element if it’s the only child element of its parent
  - :only-of-type
    matches an element that’s the only child element of its type
  - :root
    matches the element that’s the root element of the document
  - :empty
    matches elements that have no children
  - :target
    matches an element that’s the target of a fragment identifier in the document’s URI
  - :enabled
    matches user interface elements that are enabled
  - :disabled
    matches user interface elements that are disabled
  - :checked
    matches elements like checkboxes or radio buttons that are checked
  - :not(S)
    matches elements that aren’t matched by the specified selector

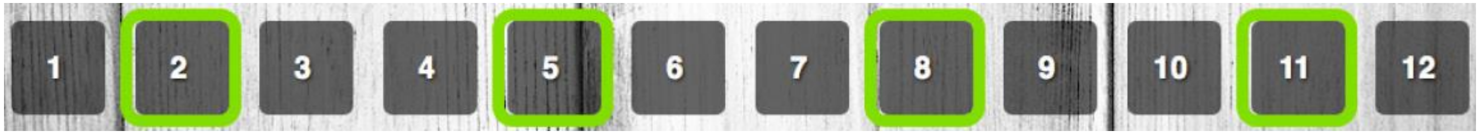
Selectors

- Examples of new pseudo-classes

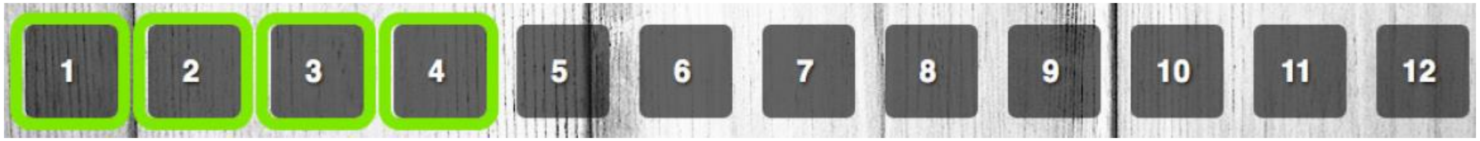
```
:nth-child(3n)
```



```
:nth-child(3n+2)
```

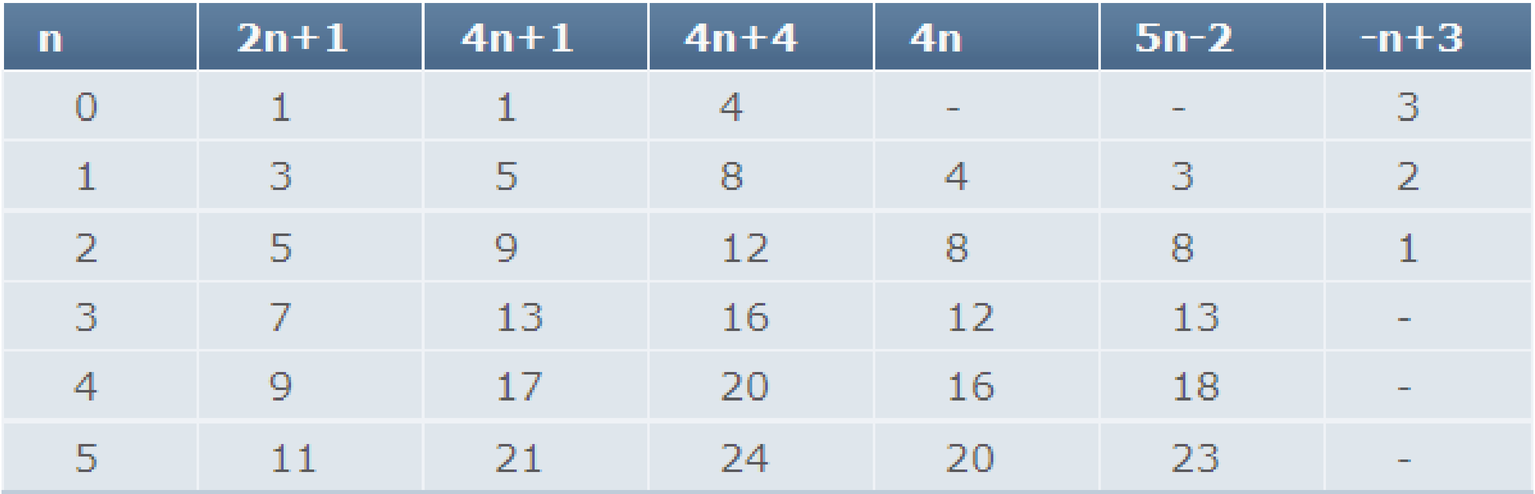


```
:nth-child(-n+4)
```



Selectors

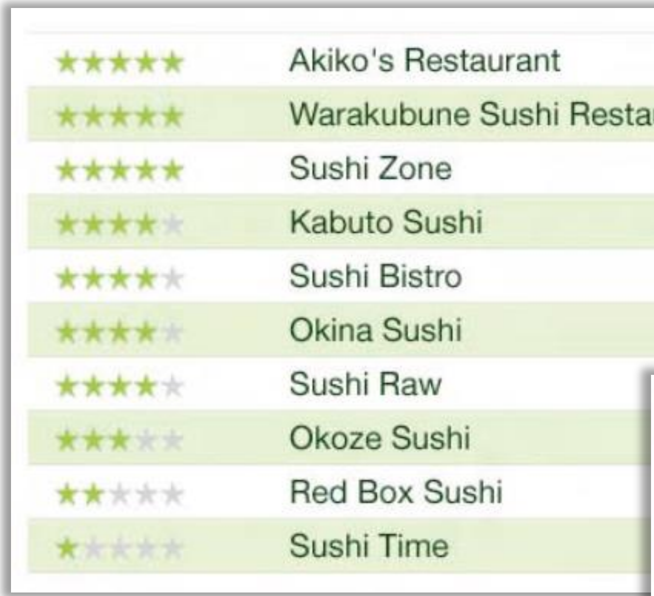
- N = pseudoclass expressions



n	2n+1	4n+1	4n+4	4n	5n-2	-n+3
0	1	1	4	-	-	3
1	3	5	8	4	3	2
2	5	9	12	8	8	1
3	7	13	16	12	13	-
4	9	17	20	16	18	-
5	11	21	24	20	23	-

Selectors

- Examples of new pseudo-classes



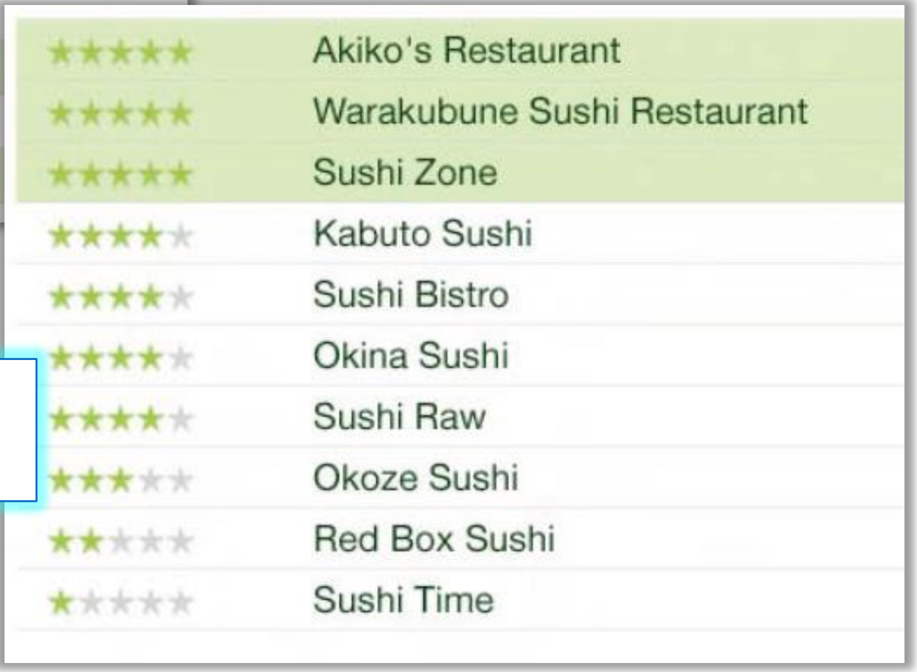
★★★★★	Akiko's Restaurant
★★★★★	Warakubune Sushi Restaurant
★★★★★	Sushi Zone
★★★★☆	Kabuto Sushi
★★★★☆	Sushi Bistro
★★★★☆	Okina Sushi
★★★★☆	Sushi Raw
★★★☆☆	Okoze Sushi
★★☆☆☆	Red Box Sushi
★☆☆☆☆	Sushi Time

```
tr:nth-child(odd) td
{ background: #ecffd9; }
```

odd = 2n+1

even = 2n

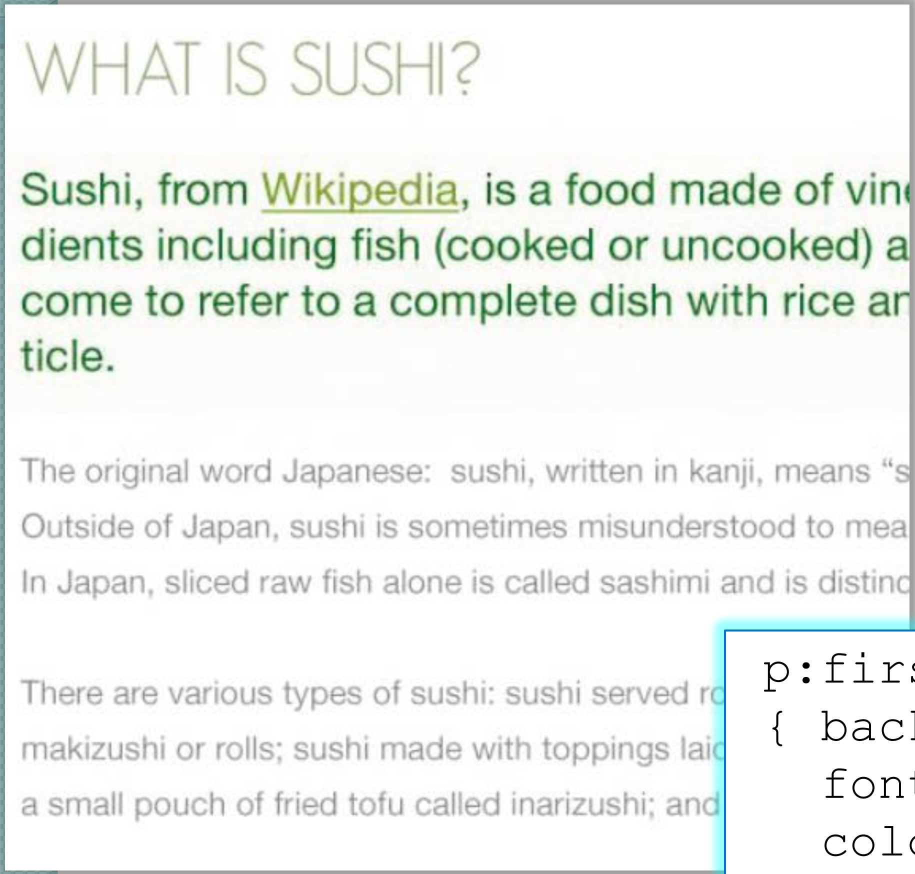
```
tr:nth-child(-n+3) td
{ background: #ecffd9; }
```



★★★★★	Akiko's Restaurant
★★★★★	Warakubune Sushi Restaurant
★★★★★	Sushi Zone
★★★★☆	Kabuto Sushi
★★★★☆	Sushi Bistro
★★★★☆	Okina Sushi
★★★★☆	Sushi Raw
★★★☆☆	Okoze Sushi
★★☆☆☆	Red Box Sushi
★☆☆☆☆	Sushi Time

Selectors

- Examples of new pseudo-classes



WHAT IS SUSHI?

Sushi, from Wikipedia, is a food made of vin
dients including fish (cooked or uncooked) a
come to refer to a complete dish with rice ar
ticle.

The original word Japanese: sushi, written in kanji, means "s
Outside of Japan, sushi is sometimes misunderstood to mea
In Japan, sliced raw fish alone is called sashimi and is distinc

There are various types of sushi: sushi served ro
makizushi or rolls; sushi made with toppings laid
a small pouch of fried tofu called inarizushi; and

```
:not (:first-child)
{ font-size: 75%; }
```

```
p:first-of-type
{ background: #fafcf5;
  font-size: 1.3em;
  color: #030;
}
```

Selectors

- Pseudo-elements
  - ::selection
    Targets elements that have been highlighted by the user
- Example

```
textarea::selection
```

match any user-selected text
within a textarea element

Color: RGBA color and opacity

- RGBA color
  - Like RGB color definitions, but allows a fourth field, defining the alpha value of the color being applied
  - Like opacity, the alpha value is between 0.0 (fully transparent) and 1.0 (fully opaque)

```
div { color: rgb(0,255,0); }
```

```
div { color: rgba(0,255,0,0.5); }
```

Color: HSLA color and opacity

- HSL color
  - Accepts three arguments: hue is a degree on a color wheel (0-360), saturation is a percentage, and lightness is a percentage
- HSLA color
  - Like HSL color, but allows a fourth field, defining the alpha value of the color being applied

```
div { color: hsl(240,50%,50%); }
```

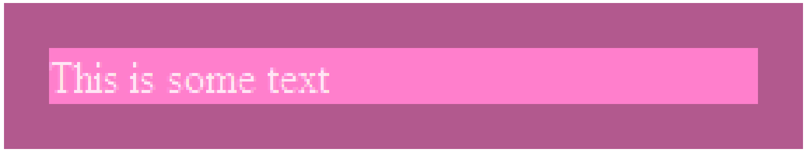
```
div { color: hsla(240,50%,50%,0.5); }
```

Color and opacity

- The difference between RGBA or HSLA and opacity is that the former applies transparency only to a particular element, whereas the latter affects the element we target and all of its children

```
div
{ color: #f00;
  opacity: 0.5; }
```

- Example:



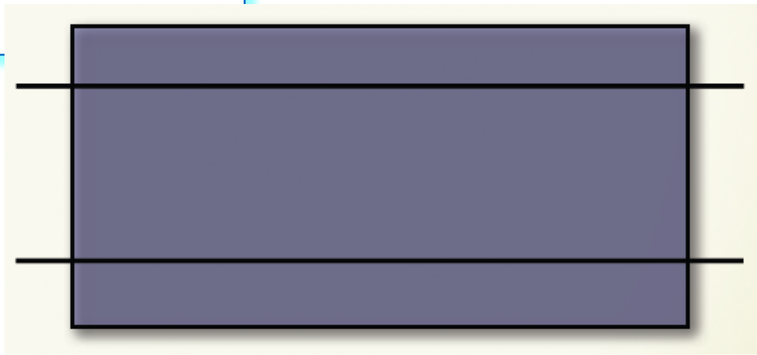
This is some text

```
#alpha-example
{ background: hsla(324, 100%, 50%, .5);
  border: 1em solid rgba(0, 0, 0, .3);
  color: rgba(255, 255, 255, .8);}
```

Multiple backgrounds

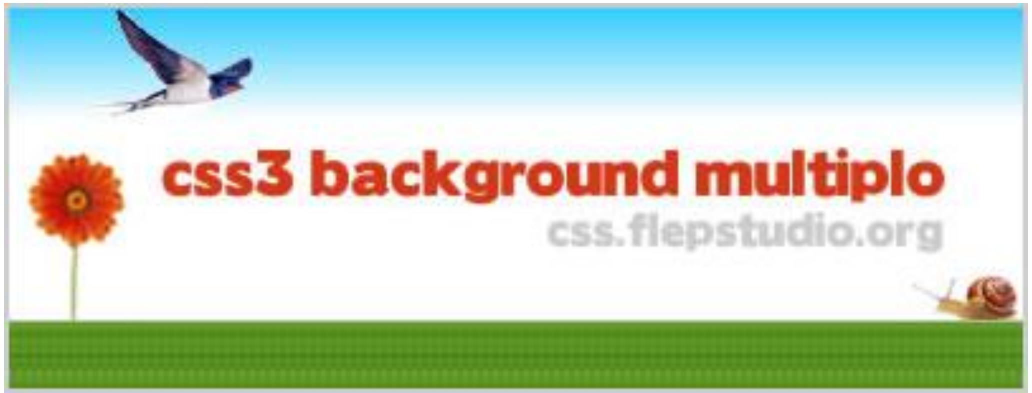
- It is possible to apply multiple layered backgrounds to an element using multiple properties such as background-image, background-repeat, background-size, background-position, background-origin and background-clip
  - The easiest way is to use the shorthand code

```
background:
  url(body-top.png) top left no-repeat,
  url(body-bottom.png) bottom left no-repeat,
  url(body-middle.png) left repeat-y;
```



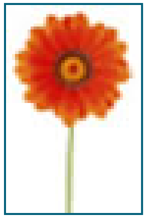
Multiple backgrounds

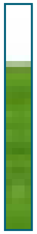
- Example



```
#multiple_background
{ width:400px;
  height:150px;
  border:2px solid #CCC;
  background:
    url(uccello.jpg) no-repeat 30px top,
    url(lumaca.jpg) no-repeat right 105px,
    url(logo.jpg) no-repeat 60px 55px,
    url(fiore.jpg) no-repeat 5px 55px,
    url(erba.jpg) repeat-x bottom,
    url(cielo.jpg) repeat-x top;
}
```

css3 background multiplo
css.flepstudio.org



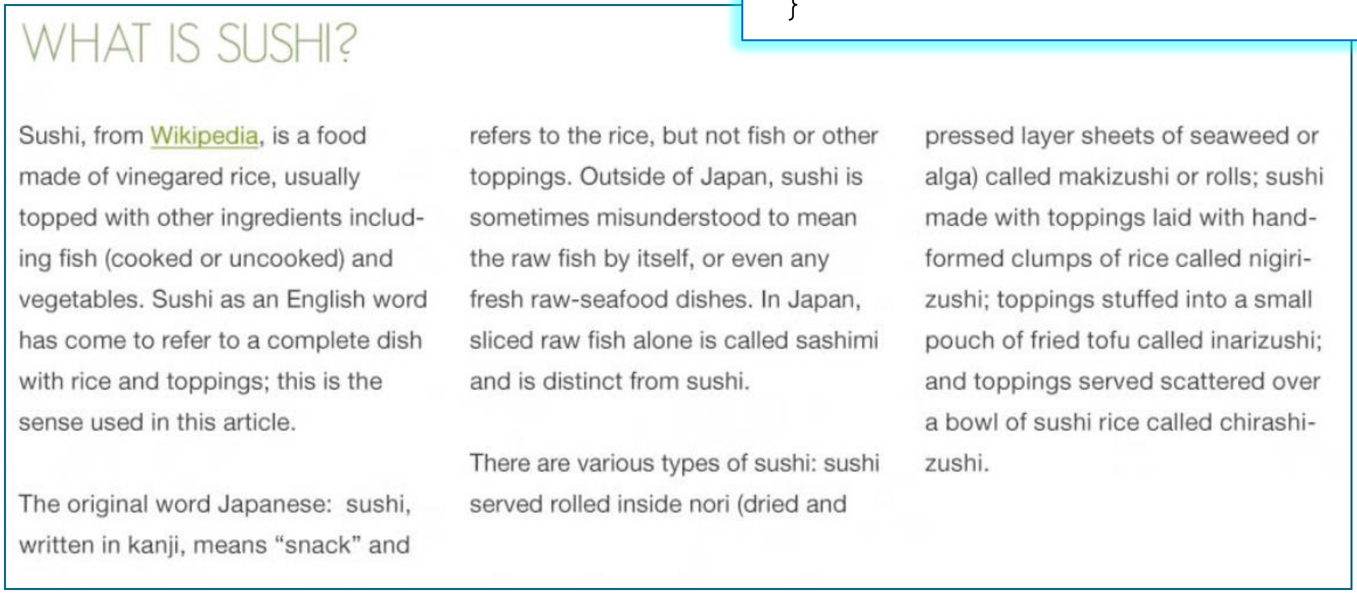


Multi-column layout

- Allows to get multi-column layouts without having to use multiple divs

```
.entry-content
{ column-count: 2;
  column-gap: 30px;
}
```

```
.entry-content
{ column-width: 270px;
  column-gap: 30px;
}
```



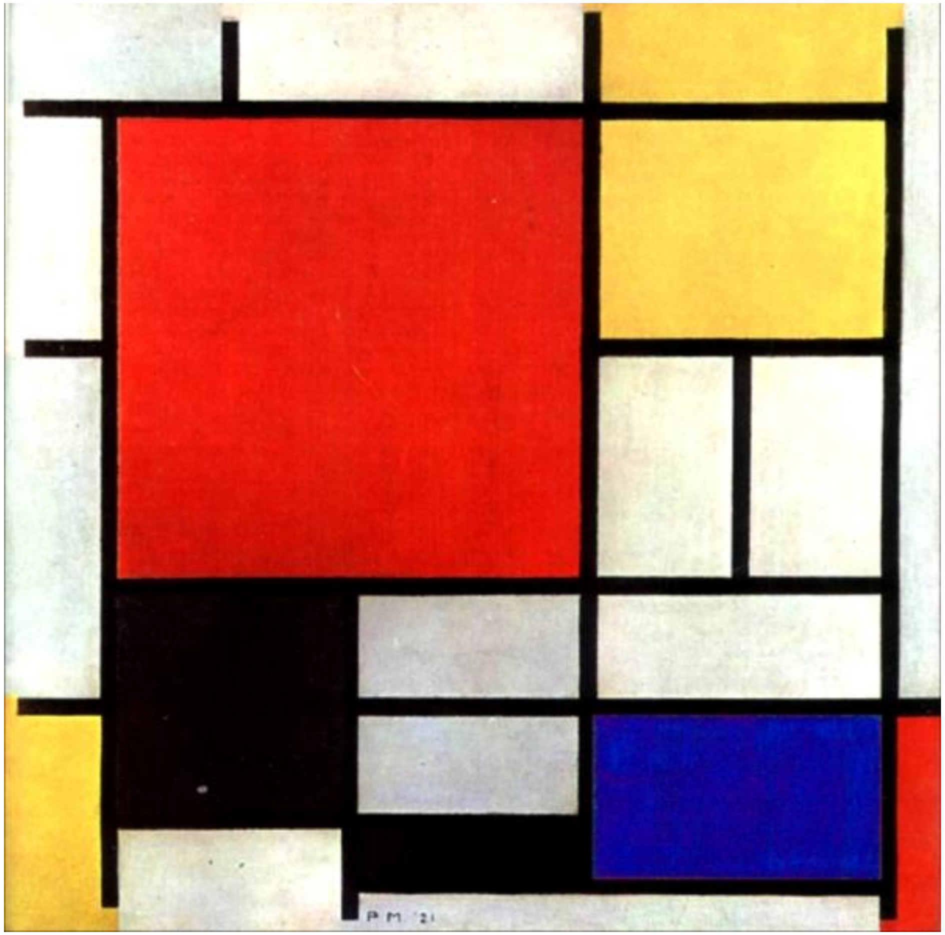
WHAT IS SUSHI?

Sushi, from Wikipedia, is a food made of vinegared rice, usually topped with other ingredients including fish (cooked or uncooked) and vegetables. Sushi as an English word has come to refer to a complete dish with rice and toppings; this is the sense used in this article.

The original word Japanese: sushi, written in kanji, means “snack” and refers to the rice, but not fish or other toppings. Outside of Japan, sushi is sometimes misunderstood to mean the raw fish by itself, or even any fresh raw-seafood dishes. In Japan, sliced raw fish alone is called sashimi and is distinct from sushi.

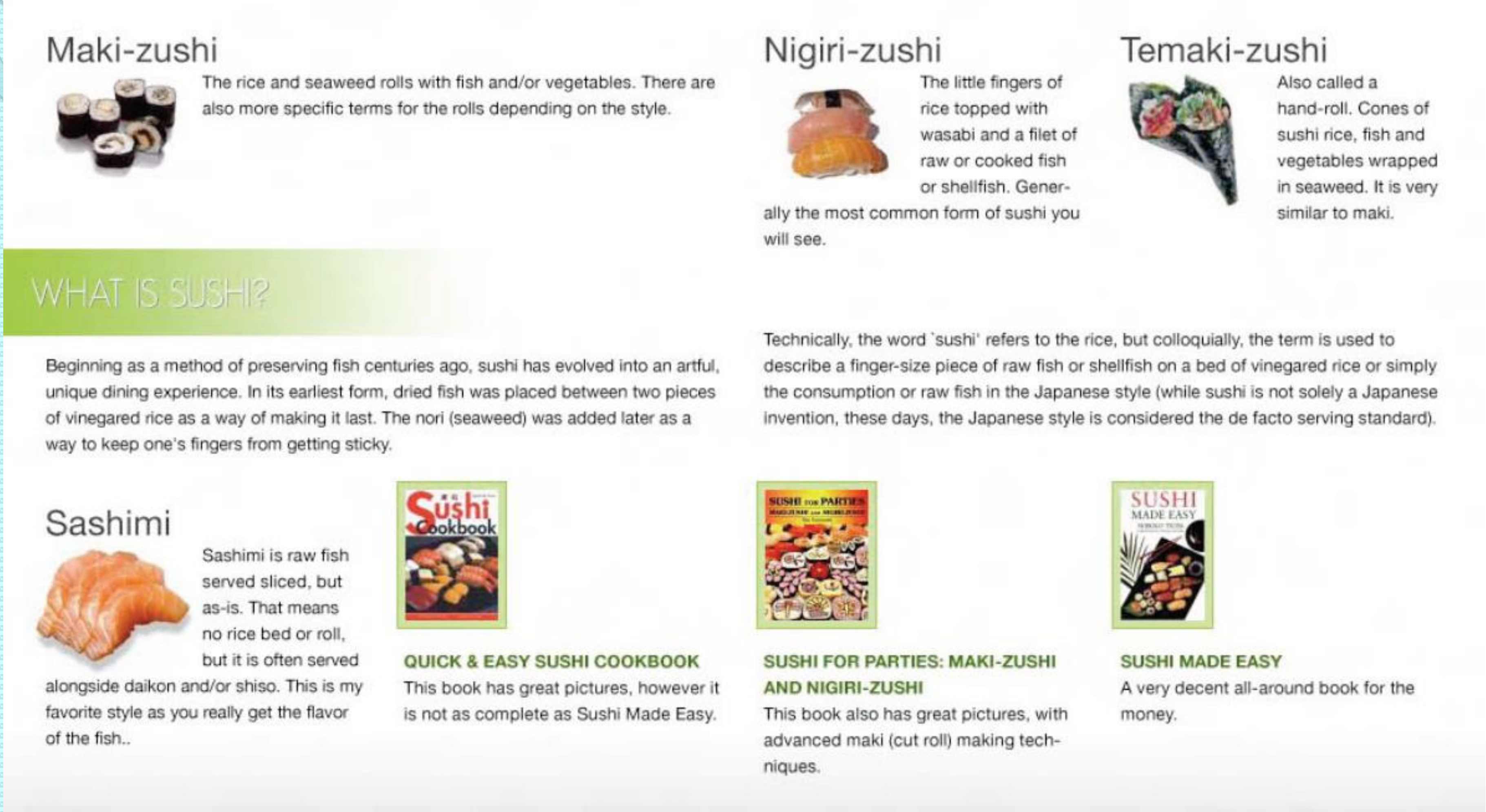
There are various types of sushi: sushi served rolled inside nori (dried and pressed layer sheets of seaweed or alga) called makizushi or rolls; sushi made with toppings laid with hand-formed clumps of rice called nigiri-zushi; toppings stuffed into a small pouch of fried tofu called inarizushi; and toppings served scattered over a bowl of sushi rice called chirashi-zushi.

Advanced layout: grid



Piet Mondrian, Composizione con grande piano rosso (1921)

Advanced layout: grid



Maki-zushi

The rice and seaweed rolls with fish and/or vegetables. There are also more specific terms for the rolls depending on the style.

Nigiri-zushi

The little fingers of rice topped with wasabi and a filet of raw or cooked fish or shellfish. Generally the most common form of sushi you will see.

Temaki-zushi

Also called a hand-roll. Cones of sushi rice, fish and vegetables wrapped in seaweed. It is very similar to maki.

WHAT IS SUSHI?

Beginning as a method of preserving fish centuries ago, sushi has evolved into an artful, unique dining experience. In its earliest form, dried fish was placed between two pieces of vinegared rice as a way of making it last. The nori (seaweed) was added later as a way to keep one's fingers from getting sticky.

Technically, the word `sushi' refers to the rice, but colloquially, the term is used to describe a finger-size piece of raw fish or shellfish on a bed of vinegared rice or simply the consumption or raw fish in the Japanese style (while sushi is not solely a Japanese invention, these days, the Japanese style is considered the de facto serving standard).

Sashimi

Sashimi is raw fish served sliced, but as-is. That means no rice bed or roll, but it is often served alongside daikon and/or shiso. This is my favorite style as you really get the flavor of the fish..

QUICK & EASY SUSHI COOKBOOK

This book has great pictures, however it is not as complete as Sushi Made Easy.

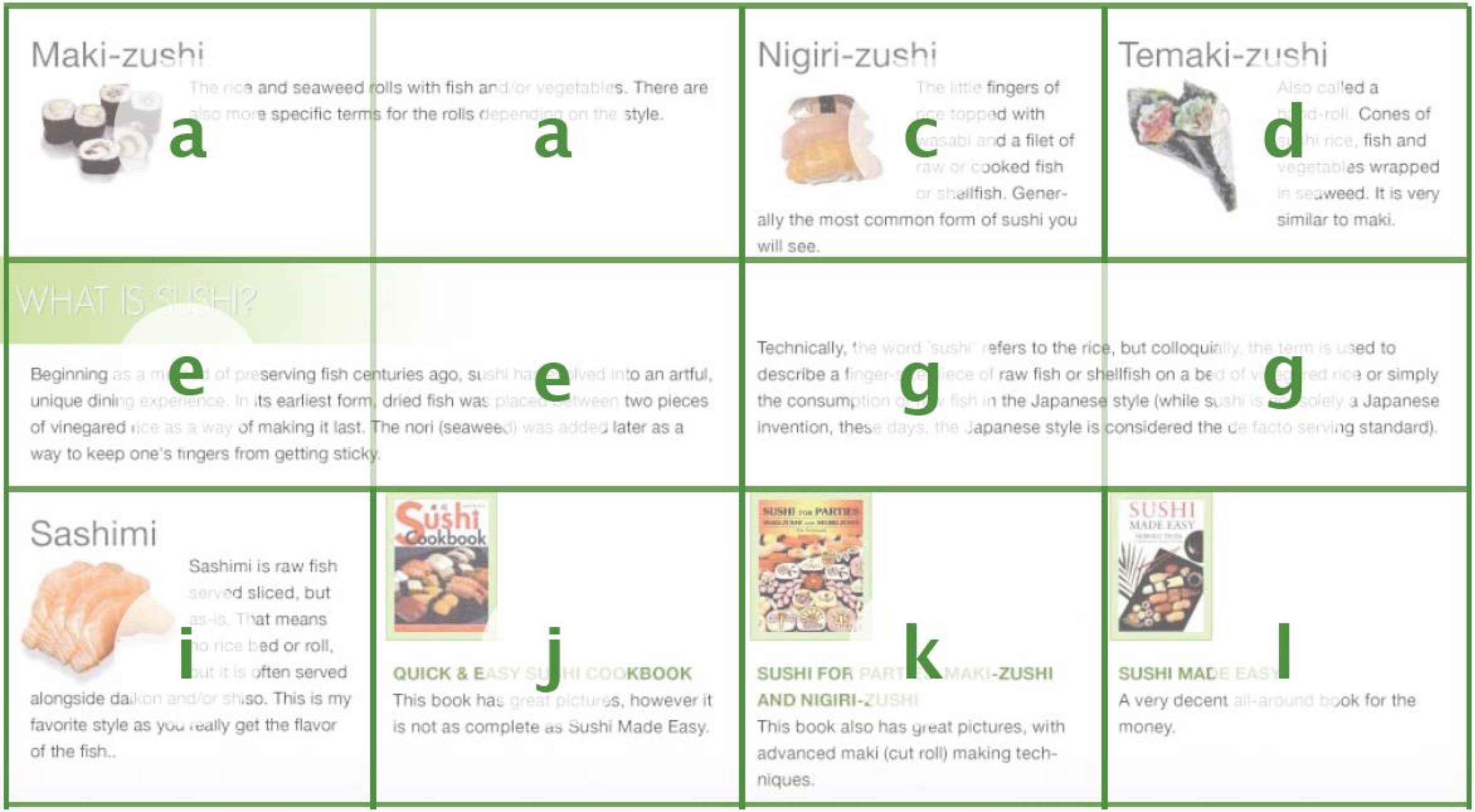
SUSHI FOR PARTIES: MAKI-ZUSHI AND NIGIRI-ZUSHI

This book also has great pictures, with advanced maki (cut roll) making techniques.

SUSHI MADE EASY

A very decent all-around book for the money.

Advanced layout: grid



Maki-zushi

The rice and seaweed rolls with fish and/or vegetables. There are also more specific terms for the rolls depending on the style.

Nigiri-zushi

The little fingers of rice topped with wasabi and a filet of raw or cooked fish or shellfish. Generally the most common form of sushi you will see.

Temaki-zushi

Also called a hand-roll. Cones of sushi rice, fish and vegetables wrapped in seaweed. It is very similar to maki.

WHAT IS SUSHI?

Beginning as a method of preserving fish centuries ago, sushi has evolved into an artful, unique dining experience. In its earliest form, dried fish was placed between two pieces of vinegared rice as a way of making it last. The nori (seaweed) was added later as a way to keep one's fingers from getting sticky.

Technically, the word 'sushi' refers to the rice, but colloquially, the term is used to describe a finger-size piece of raw fish or shellfish on a bed of vinegared rice or simply the consumption of raw fish in the Japanese style (while sushi is not solely a Japanese invention, these days, the Japanese style is considered the de facto serving standard).

Sashimi

Sashimi is raw fish served sliced, but as-is. That means no rice bed or roll, but it is often served alongside daikon and/or shiso. This is my favorite style as you really get the flavor of the fish..

QUICK & EASY SUSHI COOKBOOK

This book has great pictures, however it is not as complete as Sushi Made Easy.

SUSHI FOR PARTIES: MAKI-ZUSHI AND NIGIRI-ZUSHI

This book also has great pictures, with advanced maki (cut roll) making techniques.

SUSHI MADE EASY

A very decent all-around book for the money.

Advanced layout: grid

- It is possible to define a grid in which content can flow or be placed, or that remain empty
- There are three ways to define a grid
  - Explicit grid: defined with ‘grid-columns’ and ‘grid-rows’ properties
  - Natural grid: automatically created by elements with a natural grid structure (multi-column elements and tables)
  - Default grid: all other block elements define a single-cell grid

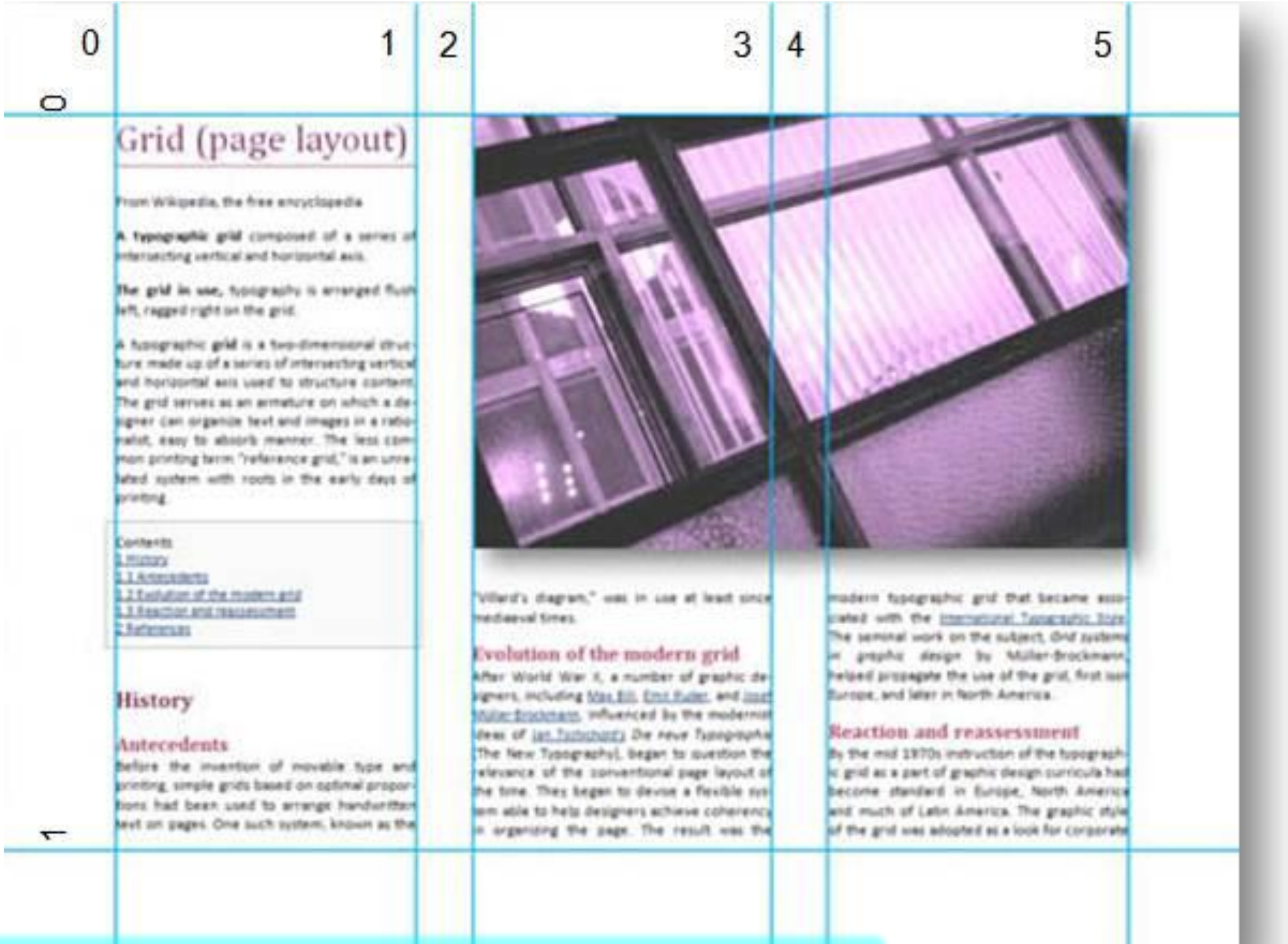
```
div
{ grid-columns:50% * * 200px; }
```

Add one grid line in the middle of the div element, another one 200 pixels from the right, and another one in the middle of remaining space

```
div
{ grid-rows: 100px (30px 60px); }
```

Define a header row of 100 pixels, and add as many additional rows as necessary, alternating heights of 30 and 60 pixels

Advanced layout: grid



Grid (page layout)

From Wikipedia, the free encyclopedia

A typographic grid composed of a series of intersecting vertical and horizontal axis.

The grid in use, typography is arranged flush left, ragged right on the grid.

A typographic **grid** is a two-dimensional structure made up of a series of intersecting vertical and horizontal axis used to structure content. The grid serves as an armature on which a designer can organize text and images in a rational, easy to absorb manner. The less common printing term "reference grid," is an unrelated system with roots in the early days of printing.

Contents

History

Antecedents

Before the invention of movable type and printing, simple grids based on optimal proportions had been used to arrange handwritten text on pages. One such system, known as the "Villard's diagram," was in use at least since mediaeval times.

Evolution of the modern grid

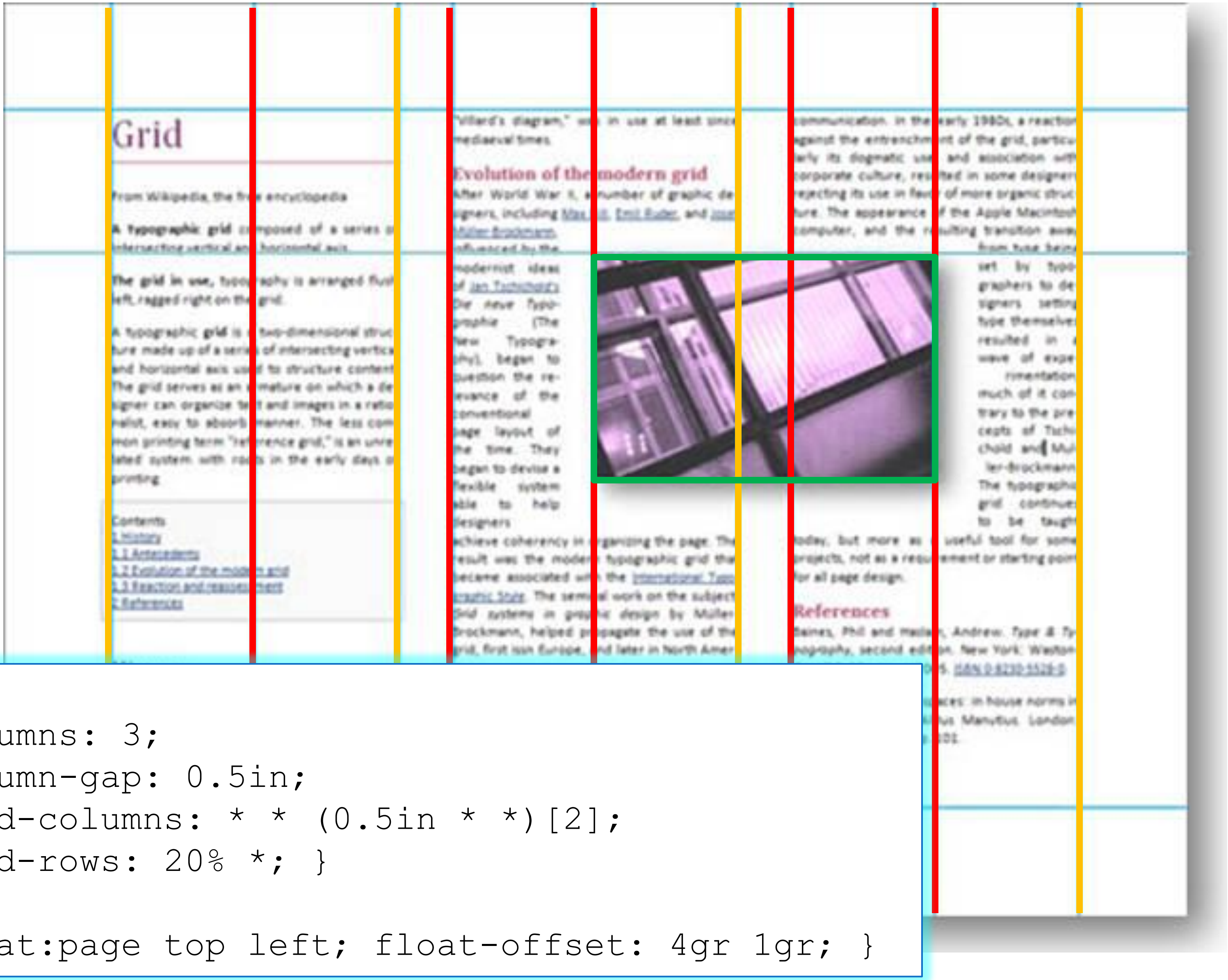
After World War II, a number of graphic designers, including Max Bill, Emil Ruder, and Josef Müller-Brockmann, influenced by the modernist ideas of Jan Tschichold's *Die neue Typographie* (The New Typography), began to question the relevance of the conventional page layout of the time. They began to devise a flexible system able to help designers achieve coherency in organizing the page. The result was the modern typographic grid that became associated with the International Typographic Style. The seminal work on the subject, *Grid systems in graphic design* by Müller-Brockmann, helped propagate the use of the grid, first in Europe, and later in North America.

Reaction and reassessment

By the mid 1970s instruction of the typographic grid as a part of graphic design curricula had become standard in Europe, North America and much of Latin America. The graphic style of the grid was adopted as a look for corporate

```
body { columns:3; column-gap:0.5in; }
img { float:page top right; width:3gr; }
```

Advanced layout: grid



Grid

From Wikipedia, the free encyclopedia

A typographic grid composed of a series of intersecting vertical and horizontal axis.

The grid in use, typography is arranged flush left, ragged right on the grid.

A typographic **grid** is a two-dimensional structure made up of a series of intersecting vertical and horizontal axis used to structure content. The grid serves as an armature on which a designer can organize text and images in a rationalist, easy to absorb manner. The less common printing term "reference grid," is an unrelated system with roots in the early days of printing.

"Villard's diagram," was in use at least since mediaeval times.

Evolution of the modern grid

After World War II, a number of graphic designers, including Max Bill, Emil Ruder, and Josef Müller-Brockmann, influenced by the modernist ideas of Jan Tschichold's Die neue Typographie (The New Typography), began to question the relevance of the conventional page layout of the time. They began to devise a flexible system able to help designers achieve coherency in organizing the page. The result was the modern typographic grid that became associated with the International Typographic Style. The seminal work on the subject Grid systems in graphic design by Müller-Brockmann, helped propagate the use of the grid, first in Europe, and later in North Amer

communication. In the early 1980s, a reaction against the entrenchment of the grid, particularly its dogmatic use and association with corporate culture, resulted in some designers rejecting its use in favor of more organic structure. The appearance of the Apple Macintosh computer, and the resulting transition away from type being set by typographers to designers setting type themselves resulted in a wave of experimentation much of it contrary to the precepts of Tschichold and Müller-Brockmann. The typographic grid continues to be taught today, but more as a useful tool for some projects, not as a requirement or starting point for all page design.

References

Baines, Phil and Haslam, Andrew. *Type & Typography*, second edition. New York: Watson
05. ISBN 0-8230-5528-5

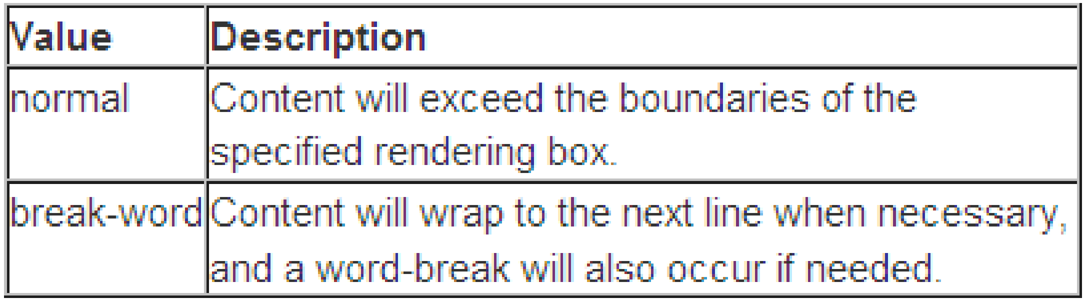
aces: in house norms in
dus Manutius. London
101

```
body
{ columns: 3;
  column-gap: 0.5in;
  grid-columns: * * (0.5in * *)[2];
  grid-rows: 20% *; }
img
{ float:page top left; float-offset: 4gr 1gr; }
```

Word wrap

- Specifies whether the current rendered line should break if the content exceeds the boundary of the specified rendering box for an element
- Example

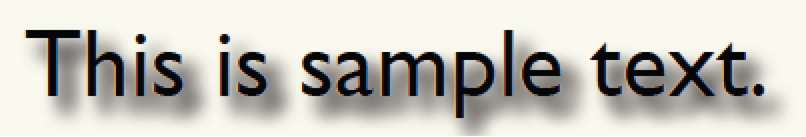
```
div
{ word-wrap: break-word }
```



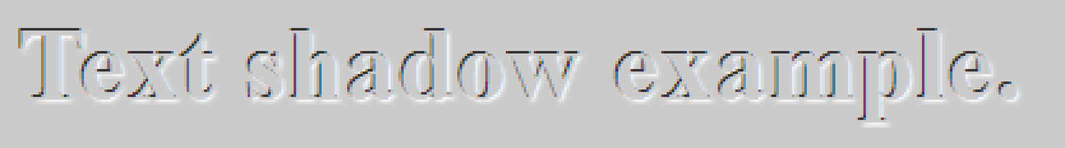
Value	Description
normal	Content will exceed the boundaries of the specified rendering box.
break-word	Content will wrap to the next line when necessary, and a word-break will also occur if needed.

Text shadow

- Arguments: horizontal offset, vertical offset, blur radius and color to be used as the shadow
- Multiple shadow definitions may be separated using commas
- Examples


This is sample text.

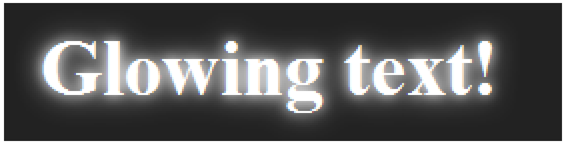
```
text-shadow: 10px 10px 10px #333;
```


Text shadow example.

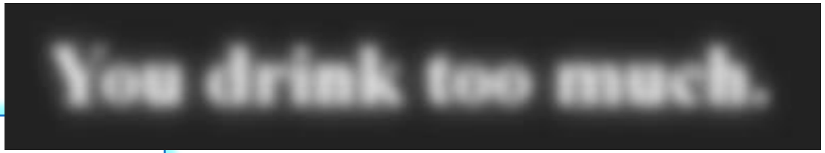
```
body
{ background: #ccc; }
p
{ margin: 0 0 1em;
  font-size: 60px;
  font-weight: bold;
  color: #ccc;letter-spacing: 1px;
  text-shadow: -1px -1px 0px #333, 1px 1px 1px #fff; }
```

Text shadow

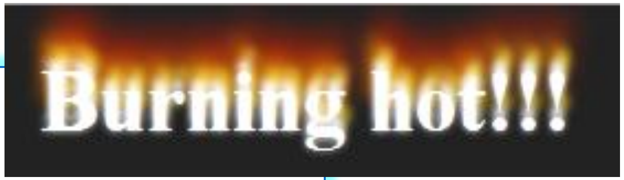
- Examples


Glowing text!

```
text-shadow: 0 0 .2em white, 0 0 .5em white;
```


You drink too much.

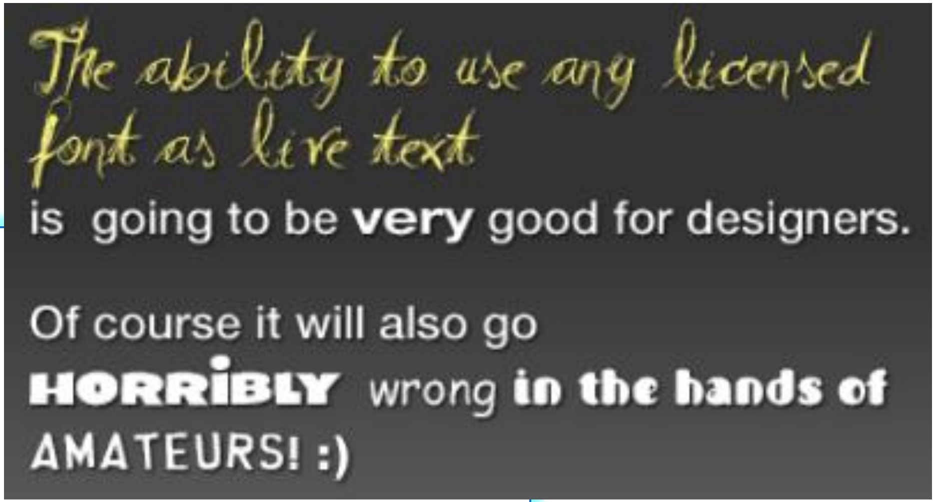
```
color: transparent:
text-shadow: 0 0 .2em white;
```


Burning hot!!!

```
text-shadow:
       0 0 4px white,
       0 -5px 4px #fff,
       2px -10px 6px #fd3,
       -2px -15px 11px #f80,
       2px -25px 18px #f20;
```

@font-face

- Simple technique for allowing designers to use their own fonts for display on the web, eliminating the constrictions that currently exist



The ability to use any licensed font as live text

is going to be **very** good for designers.

Of course it will also go

HORRIBLY wrong **in the hands of** AMATEURS! :)

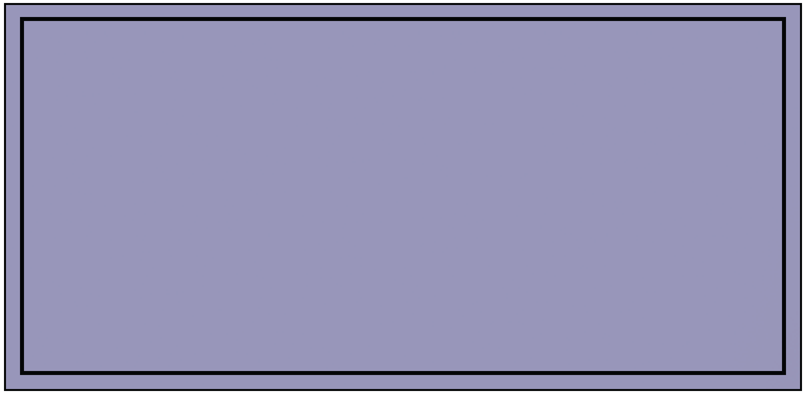
```
@font-face
{ font-family: 'DroidSans';
  src: url('droidsans.ttf')
       format('truetype');
}

h2
{ font-family: 'DroidSans', Impact, sans-serif;
}
```

Box and borders

- Border-color
  - Allows for multiple border colors to be specified, one pixel at a time

```
border: 5px solid #000;
border-color: #000 transparent transparent #000;
```



Border radius

- Border-radius
  - Curves the corners of the border using the radius given, usually in pixels
  - Can be given to all corners, or only to individual corners as specified

```
border-radius: 25px;
```

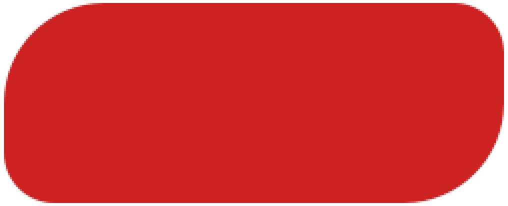
```
border-top-right-radius: 25px;
```

Border radius

top-right & bottom-left

```
border-radius: 40px 25px;
```

top-left & bottom-right

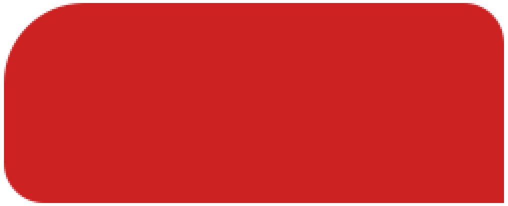


top-right & bottom-left

```
border-radius: 40px 20px 0;
```

top-left

bottom-right



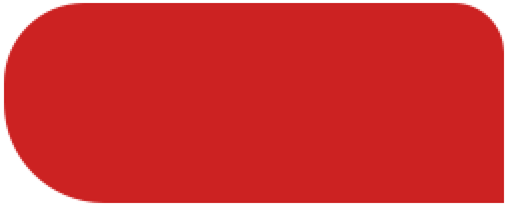
top-right

bottom-right

```
border-radius: 40px 25px 0 50px;
```

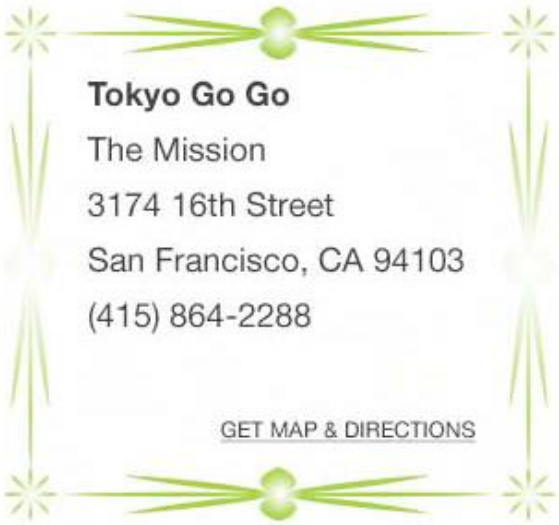
top-left

bottom-left



Border image

- Border-image
  - Authors can specify an image to be used in place of the border styles
  - In this case, the border's design is taken from the sides and corners of a specified image, whose pieces may be sliced, scaled and stretched in various ways to fit the size of the border image area
  - The border-image properties do not affect layout: layout of the box, its content, and surrounding content is based on the 'border-width' and 'border-style' properties only



Tokyo Go Go

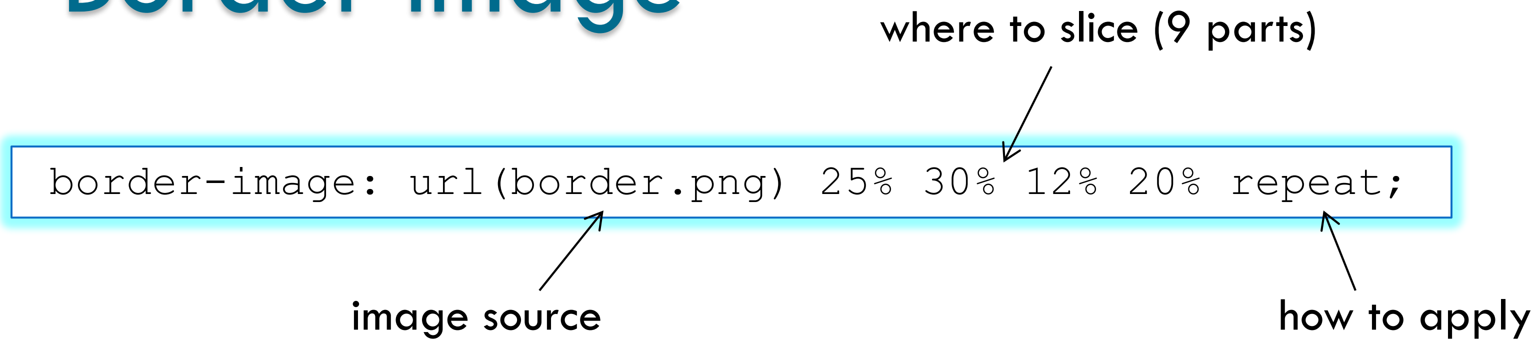
The Mission

3174 16th Street

San Francisco, CA 94103

(415) 864-2288

Border image

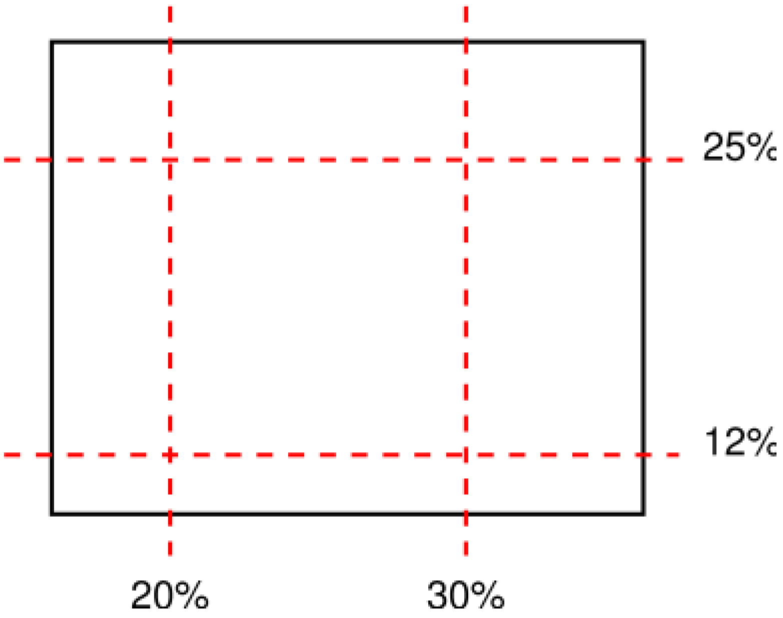


where to slice (9 parts)

```
border-image: url(border.png) 25% 30% 12% 20% repeat;
```

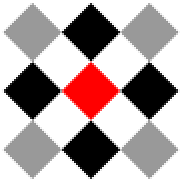
image source

how to apply



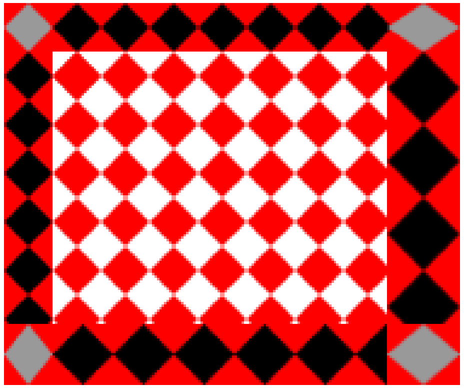
- *'stretch'*
  - The image is stretched to fill the area
- *'repeat'*
  - The image is tiled (repeated) to fill the area
- *'round'*
  - The image is tiled (repeated) to fill the area
  - If it does not fill the area with a whole number of tiles, the image is rescaled so that it does

Border image

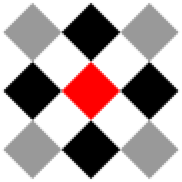


diamond.png

```
div#demo
{ border: solid transparent;
  border-width: 20px 30px 25px 20px;
  border-image: url("diamonds.png") 33% repeat }
```

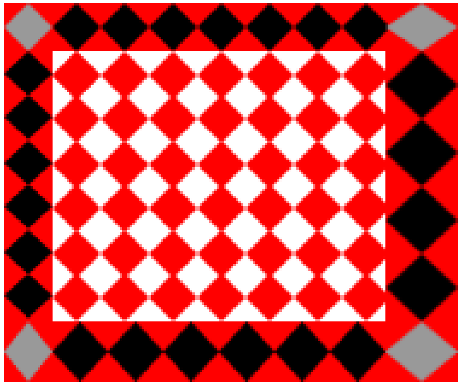


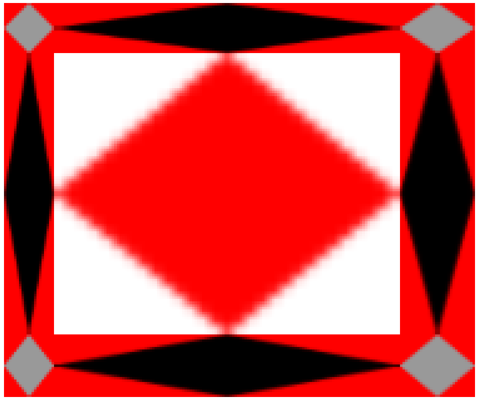
Border image



diamond.png

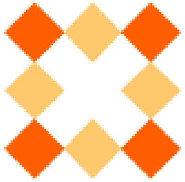
```
div#demo
{ border: solid transparent;
  border-width: 20px 30px 25px 20px;
  border-image: url("diamonds.png") 33% round }
```



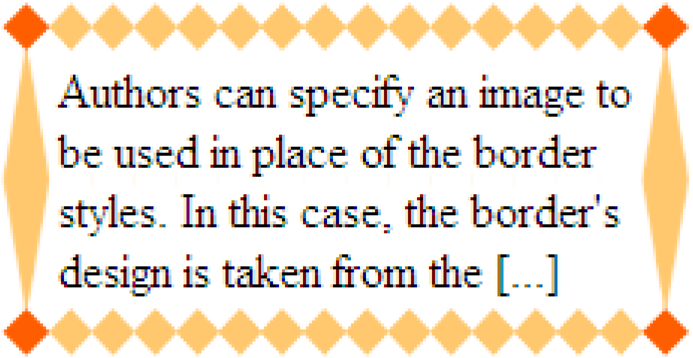


```
div#demo
{ border: solid transparent;
  border-width: 20px 30px 25px 20px;
  border-image: url("diamonds.png") 33% stretch }
```

Border image



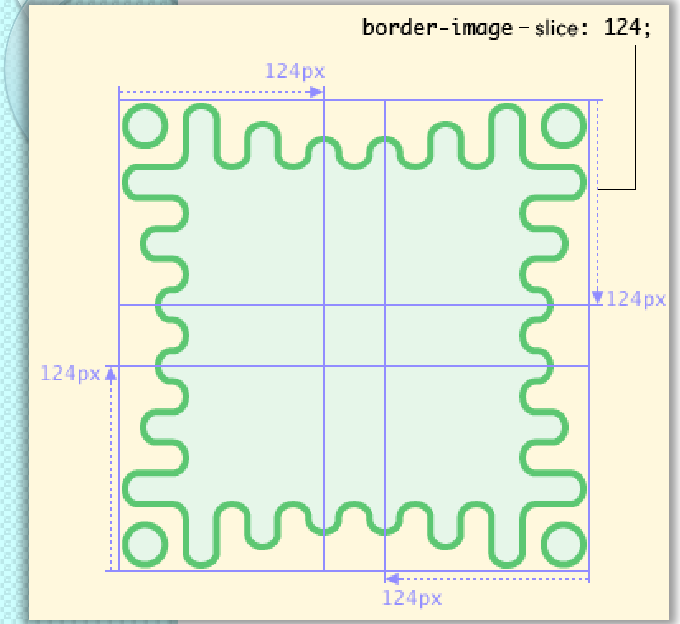
border.png
(81x81 px)

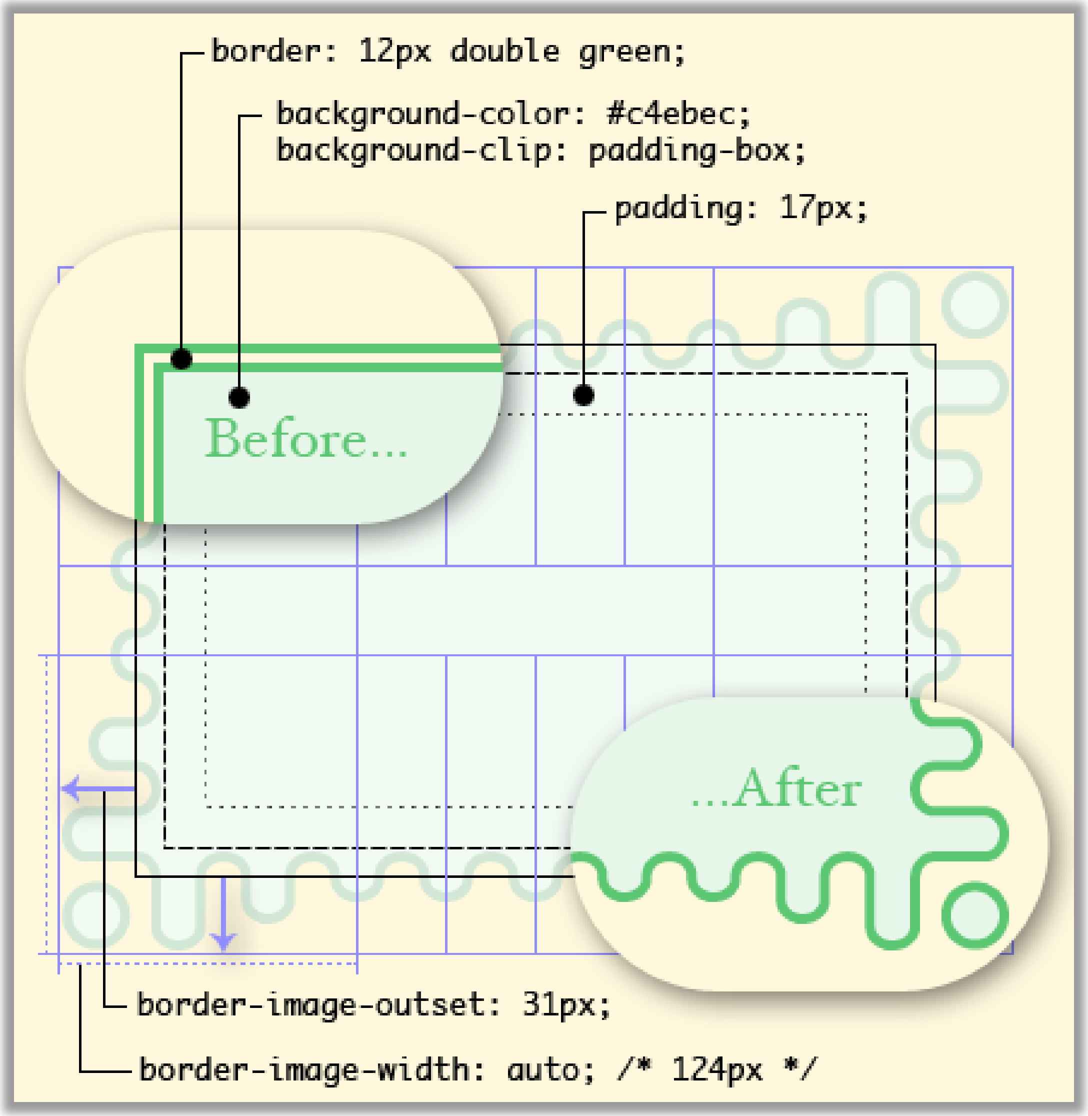


Authors can specify an image to be used in place of the border styles. In this case, the border's design is taken from the [...]

```
p
{ width: 12em;
  height: 5em;
  margin: 5em;
  padding:3px;
  border: double orange 1em;
  border-image: url("border.png") 27 27 27 27 round stretch; }
```

Border image

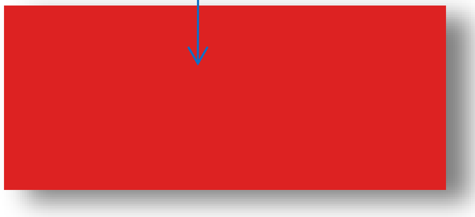


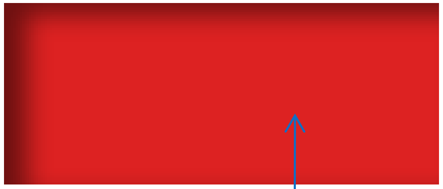

border: 12px double green;
background-color: #c4ebec;
background-clip: padding-box;
padding: 17px;
Before...
...After
border-image-outset: 31px;
border-image-width: auto; /* 124px */

Box shadow

- Box-shadow
  - Creates a drop shadow beneath the selected element
  - The first argument is the horizontal offset, the second is the vertical offset, the third is the blur radius, and the final argument is the color to be used as the shadow

```
box-shadow: 10px 5px 15px rgba(0,0,0,.5);
```

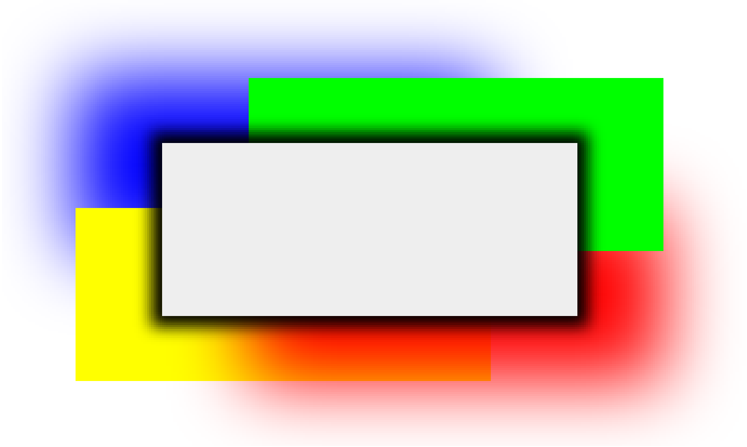


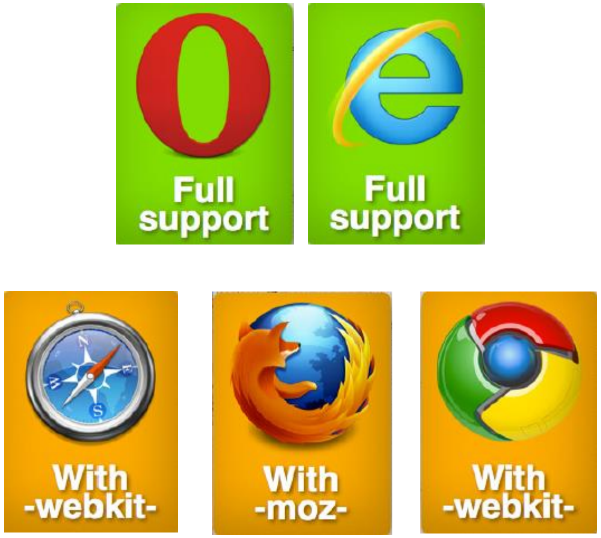


```
box-shadow: 10px 5px 15px rgba(0,0,0,.5) inset;
```

Box shadow

- Multiple shadows



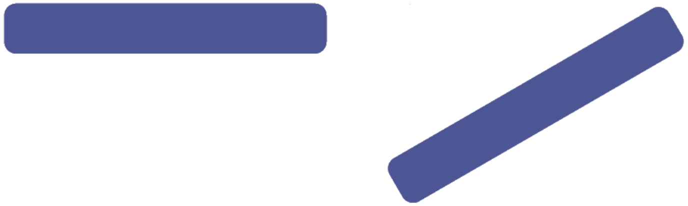


```
#Example_M {
  -moz-box-shadow: 0 0 5px black, 40px -30px lime,40px 30px
    50px red, -40px 30px yellow, -40px -30px 50px blue;
  -webkit-box-shadow: 0 0 5px black, 40px -30px lime, 40px
    30px 50px red, -40px 30px yellow, -40px -30px 50px blue;
  box-shadow: 0 0 5px black, 40px -30px lime, 40px 30px
    50px red, -40px 30px yellow, -40px -30px 50px blue;
}
```

Transformations

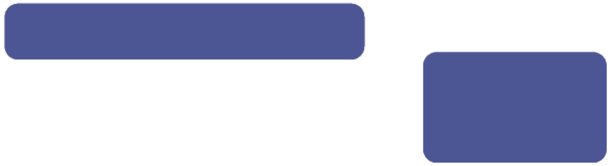
- Rotate
  - Rotates the selected element at the defined angle, defined in degrees
  - The rotation doesn't affect layout, and elements that are transformed are treated similarly to position:relative

```
transform: rotate(30deg);
```



- Scale
  - Scales the element in question based on the specified unit-less numbers given for the X and Y axes
  - If only one number is given, it is applied to both axes

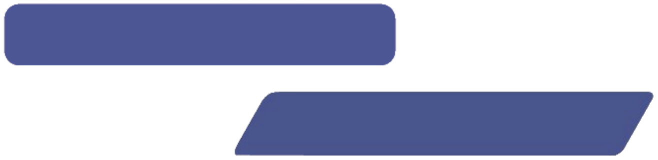
```
transform: scale(0.5,2.0);
```



Transformations

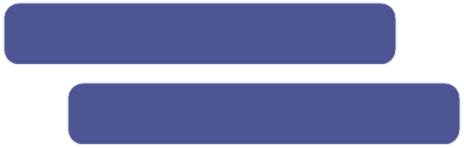
- Skew
  - Skews the element around the X and Y axes by the specified angles, in degrees
  - If it's only one number, the Y axis is assumed to be zero

```
transform: skew(-30deg);
```



- Translate
  - Moves the object along each axis by the length specified
  - The unit can be anything accepted as a length in CSS, such as px, em, percentages, …

```
transform: translate(30px, 0);
```



Transitions

- Create an effect when changing from one style to another

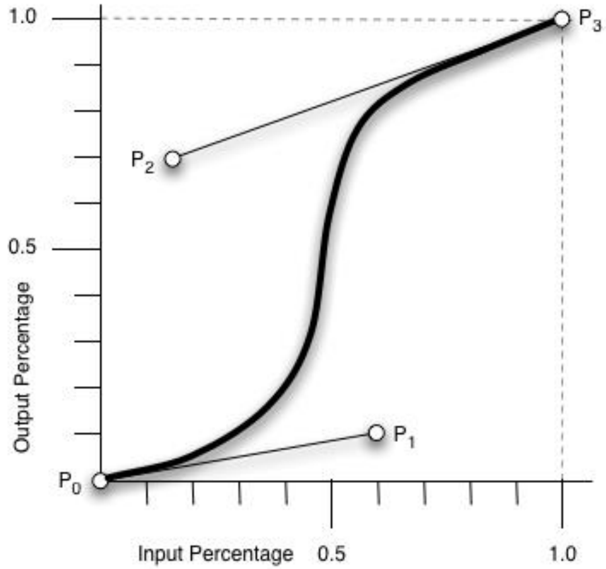
```
div
{
  width:100px;
  height:100px;
  background:red;
  transition:width 2s;
  -moz-transition:width 2s; /* Firefox 4 */
  -webkit-transition:width 2s; /* Safari and Chrome */
-  o-transition:width 2s; /* Opera */
}

div:hover
{
width:300px;
}
```

http://www.w3schools.com/css/css3_transitions.asp

Transitions

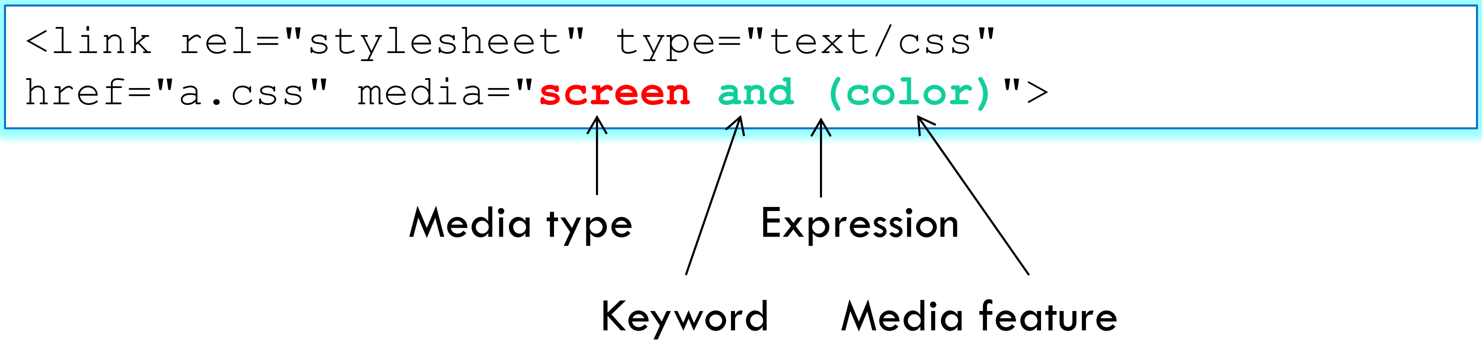
- Funzione di timing
  - Il modo di calcolare i valori intermedi durante la transizione: permette di cambiare la velocità della transizione durante la sua durata
  - Valori: ease, ease-in, ease-out, ease-in-out, linear (equivalenti a specifiche curve di Bezier), cubic-bezier (P_0, P_1, P_2, P_3)



- http://www.html5today.it/tutorial/css3-transitions-tutorial-completo-transizioni-css3
- http://www.w3.org/TR/css3-transitions/#transition-timing-function_tag

Media queries

- Allow to change layout to suit the exact need of different devices without changing the content
- A media query generally consists of a media type and zero or more expressions
- An expression consists of zero or more keywords and a media feature



```
<link rel="stylesheet" type="text/css"
href="a.css" media="screen and (color)">
```

- A media feature can be used without a media type or keyword: in this case the media type is assumed to be “all”

```
media="(color)"
```

Media types

- CSS can be used to specify how a document is presented in different media
- CSS 2.1defines ten media types

all	suitable for all devices
aural	for speech synthesizers
braille	for Braille tactile feedback devices
embossed	for paged Braille printers
handheld	for handheld devices
print	for print material
projection	for projected presentations
screen	for color computer screens
tty	for teletypes and terminals
tv	for television type devices

Media queries

- Most media features accept “min-” or “max-” prefixes

```
media="screen and (min-height: 20em)"
```

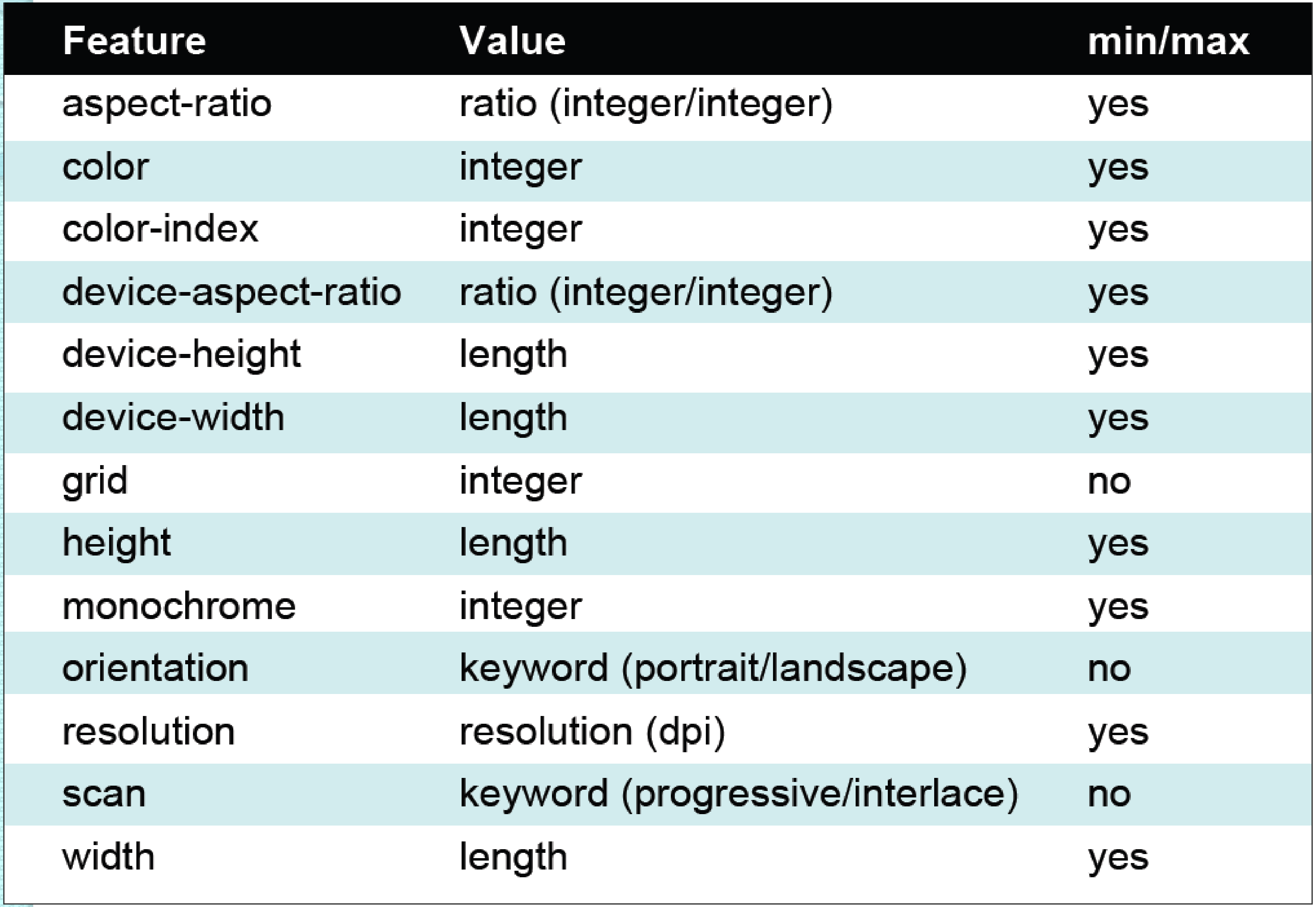
- Media features can often be used without a value

```
media="screen and (color)"
```

- Media features only accept single values: one keyword, one number, or a number with a unit identifier

```
(orientation: portrait)
(min-width: 20em)
(min-color: 2)
(device-aspect-ratio: 16/9)
```

Media features



Feature	Value	min/max
aspect-ratio	ratio (integer/integer)	yes
color	integer	yes
color-index	integer	yes
device-aspect-ratio	ratio (integer/integer)	yes
device-height	length	yes
device-width	length	yes
grid	integer	no
height	length	yes
monochrome	integer	yes
orientation	keyword (portrait/landscape)	no
resolution	resolution (dpi)	yes
scan	keyword (progressive/interlace)	no
width	length	yes

Media queries

- Examples

```
media="handheld and (min-width:20em)
       and (max-width:40em)"
```

applied by hand-held devices, but only if the viewport width is between 20em and 40em

```
media="screen and (color),
handheld and (color)">
```

applied to screen with color or handheld devices with color

Media queries

- Examples

```
media="not screen and (color)">
```

applied to all devices except those with color screens

```
media="only screen and (color)">
```

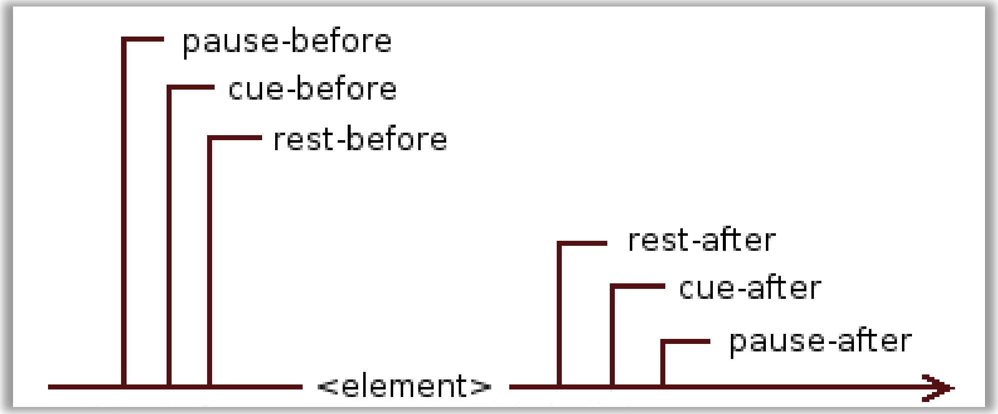
applied only to all devices with color screens

http://www.webdesignerwall.com/demo/media-queries/

Speech

- Allows to specify the speech style of screen readers by controlling various aspects of the speech, such as:
  - voice-volume
    Set a volume using a number from 0 to 100 (0 being silence), percentages or a keyword (silent, x-soft, soft, medium, loud and x-loud)
  - voice-family
    Set specific types of voices and voice combinations (as you do with fonts)
  - voice-balance
    Control which channel sound comes from (if the user's sound system supports stereo)
  - speak
    Instruct the screen reader to spell out particular words, digits or punctuation. Available keywords are none, normal, spell-out, digits, literal-punctuation, no-punctuation and inherit

Speech



- pauses and rests
  Set a pause or rest before or after an element’s content is spoken. You can use either time units (for example, “2s” for 2 seconds) or keywords (none, x-weak, weak, medium, strong and x-strong)
- cues
  Use sounds to delimit particular elements and control their volume
- voice-rate
  Control the speed at which elements are spoken. This can be defined as a percentage or keyword: x-slow, slow, medium, fast and x-fast
- voice-stress
  Indicate any emphasis that should be applied, using different keywords: none, moderate, strong and reduced

Speech

- Examples

```
h2
{ voice-family: female;
  voice-balance: left;
  voice-volume: soft;
  cue-after: url(sound.au);
}
```

Tells a screen reader to read all h2 tags in a female voice, from the left speaker, in a soft tone and followed by a particular sound

```
h1 { voice-family: announcer, old male }
p.part.romeo { voice-family: romeo, young male }
p.part.juliet { voice-family: juliet, female }
p.part.mercutio { voice-family: male 2 }
p.part.tybalt { voice-family: male 3 }
p.part.nurse { voice-family: child female }
```

Current status

- Info about development status
  - http://www.w3.org/Style/CSS/current-work
  - http://www.css3.info/modules/
- CSS Properties Index
  - http://meiert.com/en/indices/css-properties/
- Browser support for CSS3
  - http://www.normansblog.de/demos/browser-support-checklist-css3/

Licenza d'uso

- Queste diapositive sono distribuite con licenza Creative Commons "Attribuzione - Non commerciale - Condividi allo stesso modo 2.5 Italia (CC BY-NC-SA 2.5)"
- Sei libero:
  - di riprodurre, distribuire, comunicare al pubblico, esporre in pubblico, rappresentare, eseguire e recitare quest'opera
  - di modificare quest'opera
- Alle seguenti condizioni:
  - **Attribuzione** — Devi attribuire la paternità dell'opera agli autori originali e in modo tale da non suggerire che essi avallino te o il modo in cui tu usi l'opera.
  - **Non commerciale** — Non puoi usare quest'opera per fini commerciali.
  - **Condividi allo stesso modo** — Se alteri o trasformi quest'opera, o se la usi per crearne un'altra, puoi distribuire l'opera risultante solo con una licenza identica o equivalente a questa.
- http://creativecommons.org/licenses/by-nc-sa/2.5/it/