



CSS

Web Engineering

Dierk König



@ProfDKoenig

The Two Faces



CSS Rule

selector

value



h1 { color: blue; }



property

CSS Selectors

h1 <h1></h1> element


#top <.. id="top"> id

.cool <.. class="cool"> class

* all

*can be a list like
"cool drink"*

Special Selectors

[title]	<.. title="..">	attribute
[title="x"]	<.. title="x">	attribute/value ^= \$= *= ~=
p:hover	<p ... 	pseudo-class

Special Selectors

`h1+`

next sibling

`h1~`

all siblings

P.S. there is much more that we don't cover

:not, :nth-child, :nth-child-of-type pseudo-class-functions

::first-letter pseudo-elements

Selector Combinations

h1, h2

union

main p

descendant

ul > li

direct child

Selector Combinations

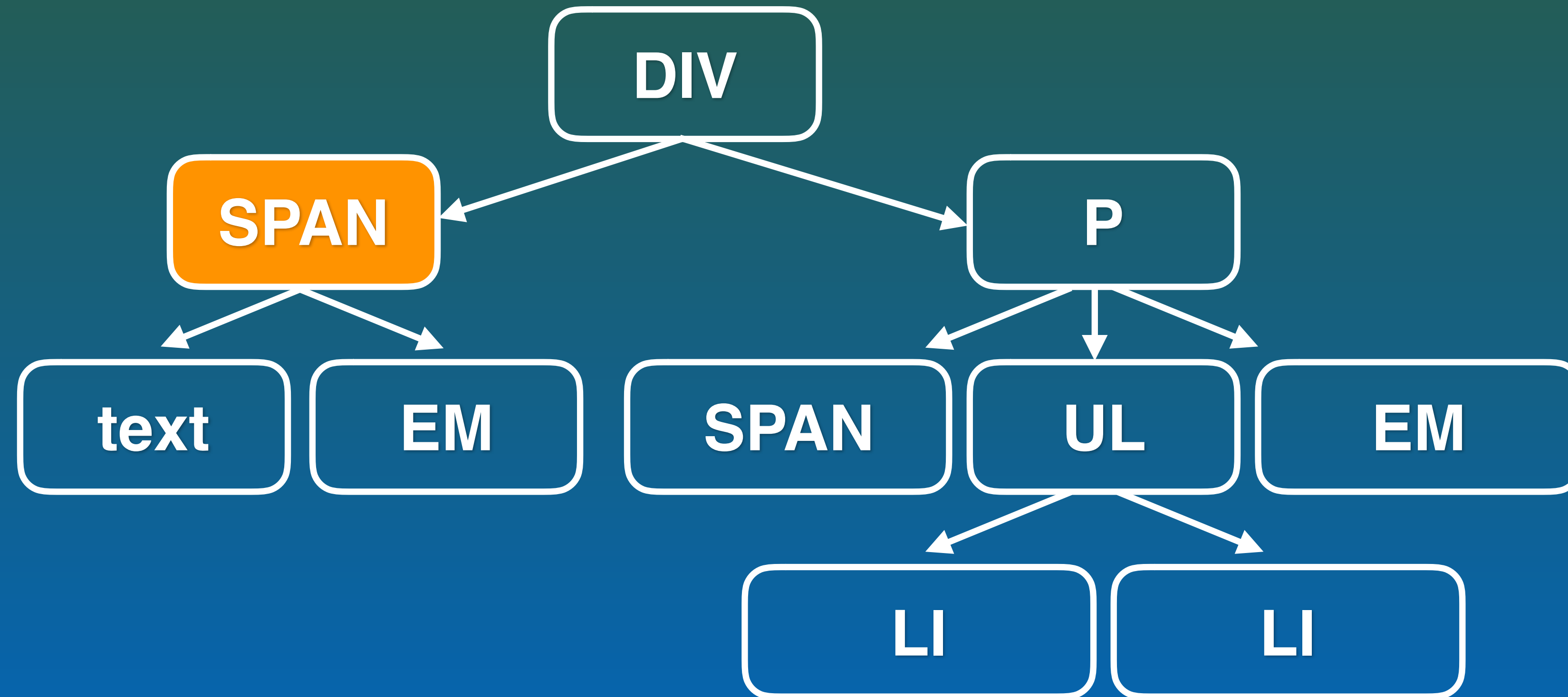
div.cool

element with class

div[cool]

element with
attribute

What selects: div > span



CSS Inline

HTML

..

```
<p style="color:blue;">
```

..

```
</p>
```

..

Style Element

```
HTML
<head>
  <style>
    p { color:blue; }
  </style>
</head>
```

Link Element

HTML

<head>

<link

rel = "stylesheet"

href = "style.css">

</head>

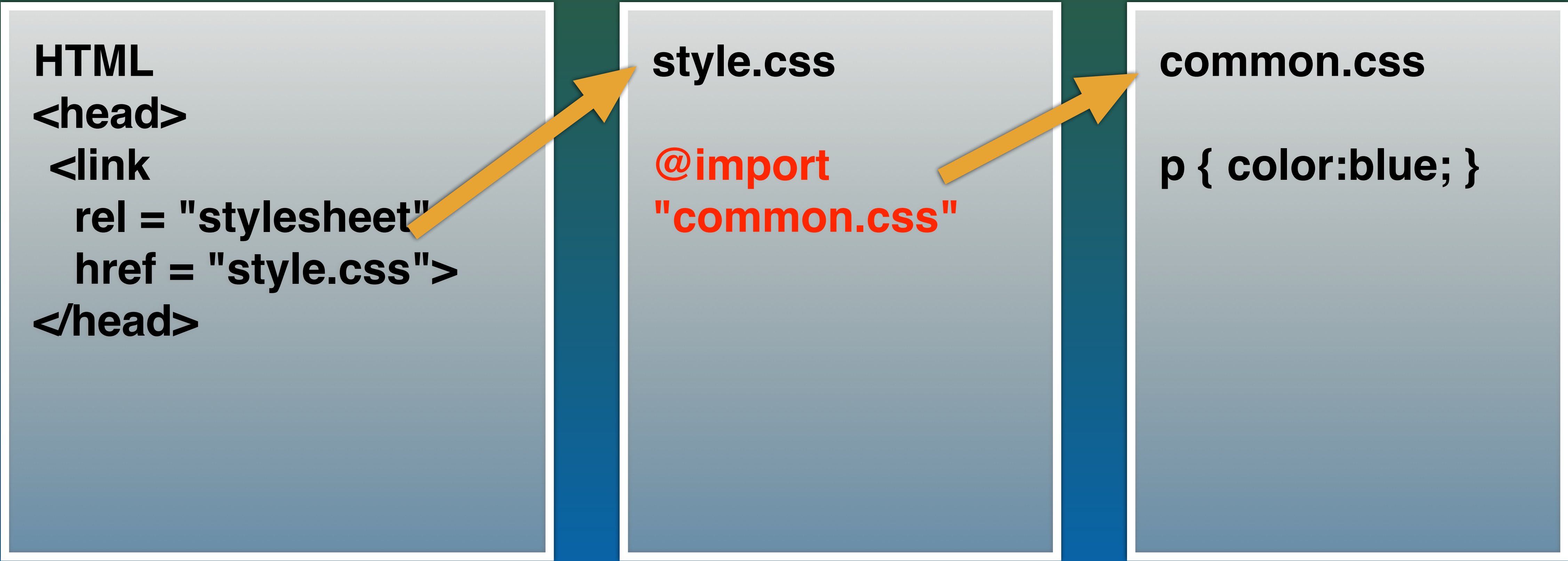
style.css

p { color:blue; }



CSS Import

HTML
<head>
<link
rel = "stylesheet"
href = "style.css">
</head>



```
graph LR; HTML[HTML] --> stylecss[style.css]; stylecss --> commoncss[common.css]
```

The diagram illustrates the process of CSS import. It consists of three light gray rectangular boxes with white borders, arranged horizontally. The first box on the left contains HTML code for linking a stylesheet. An orange arrow points from the 'style.css' href in the HTML code to the top of the second box. The second box is titled 'style.css' and contains an '@import' statement for 'common.css'. Another orange arrow points from this '@import' statement to the top of the third box. The third box is titled 'common.css' and contains a CSS rule for the 'p' selector.

style.css

@import
"common.css"

common.css

p { color:blue; }

Cascade Rules of Thumb

DOM inheritance *for some properties*

Sheet importance *author > user > browser*

Specificity *id > class > element*

*Count**




**union elements count separately*

Conflict Resolution

Source order

*latest wins,
unless important*



```
p { color: blue; }  
p { color: red; }
```

```
p { color: blue !important; }  
p { color: red; }
```



Cascade Rules of Thumb

Source order *latest wins*

Note:

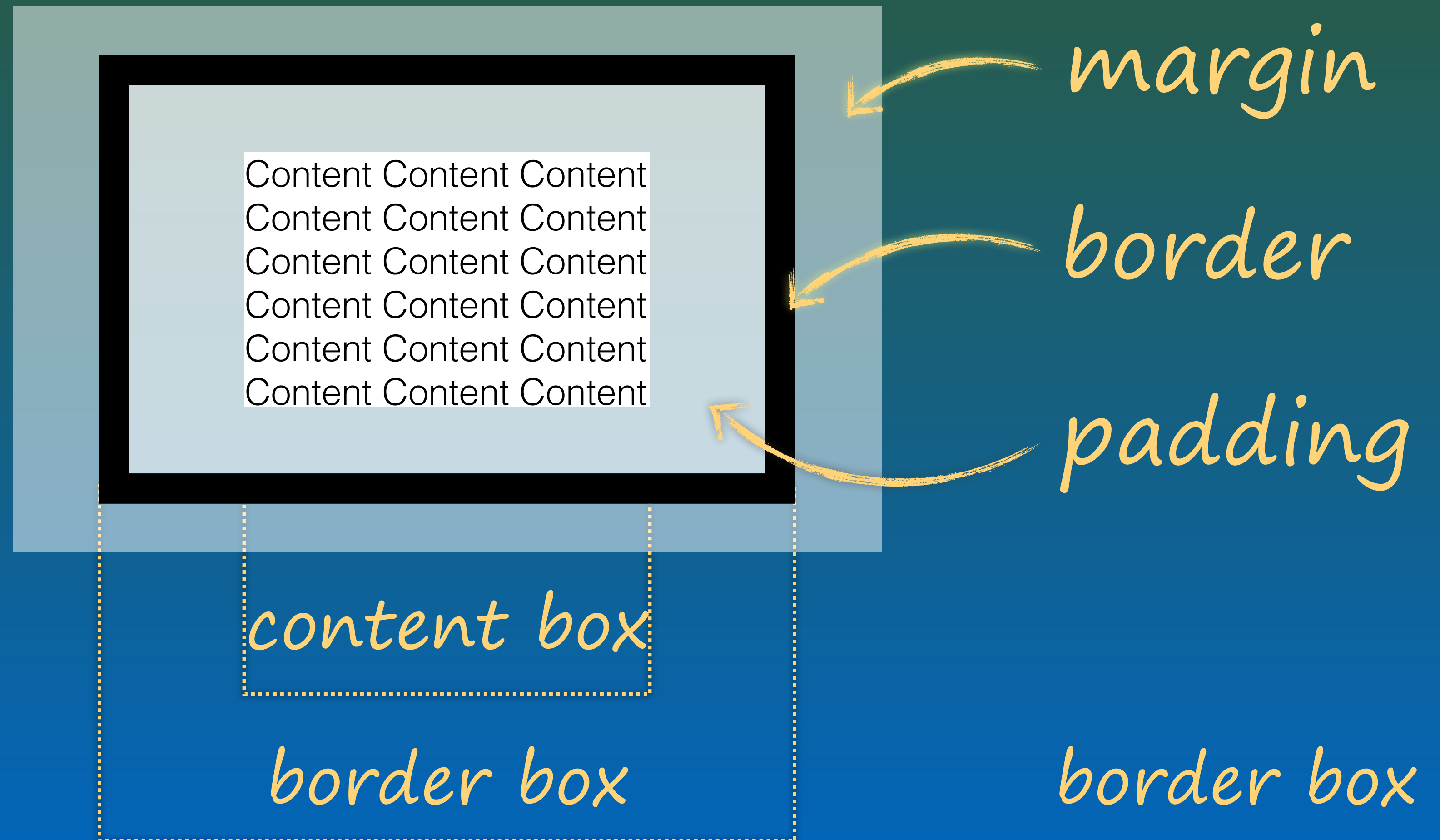
`p { color: blue !important; }` *overrides all the above!*

Properties

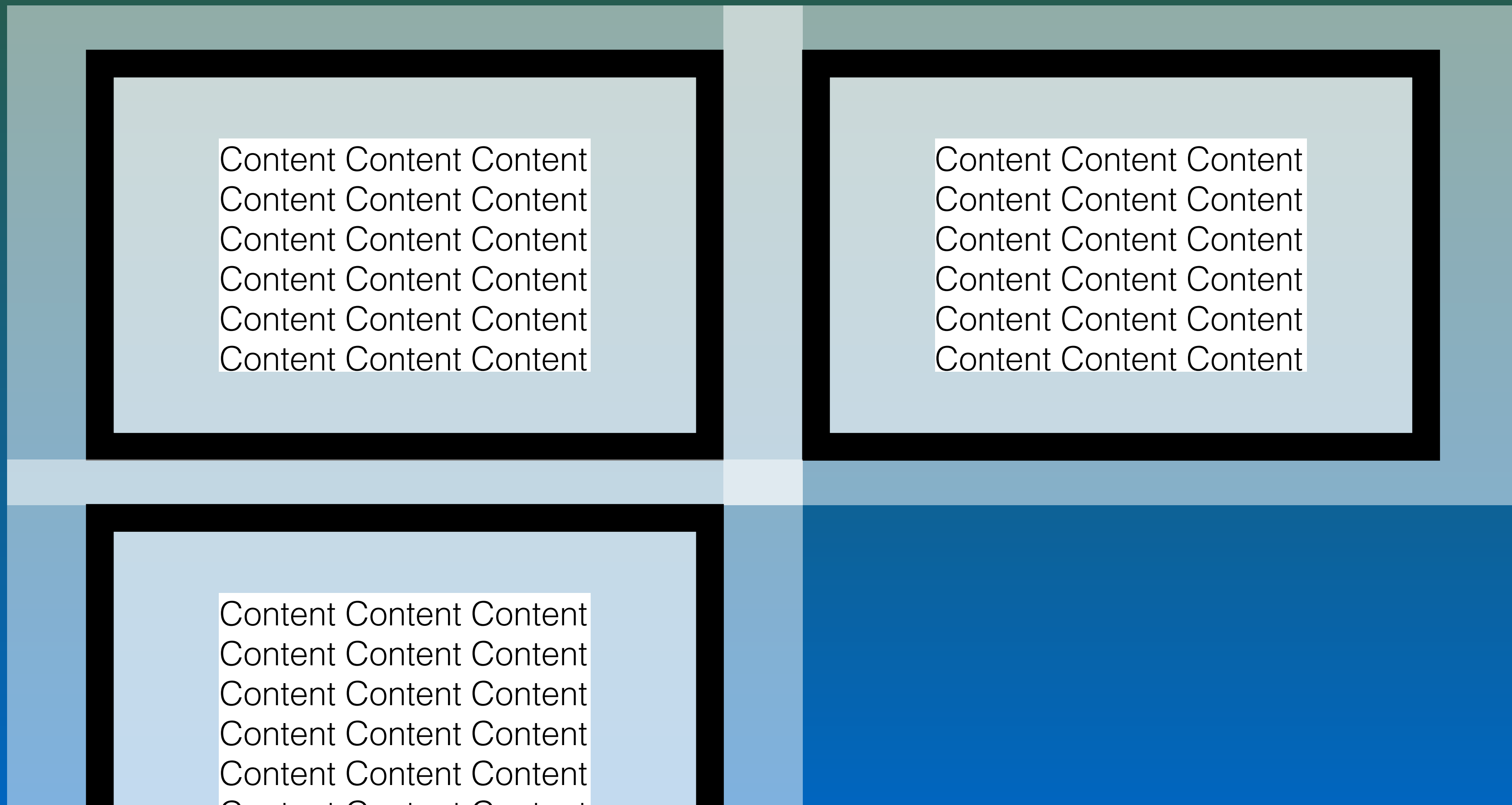
There are too many properties
and possible values to list here.

We can only cover how it works
in principle.

Box Model



Margins *might* collapse



Units

px

pixel

cm, mm, in, pt

absolute length

em, rem

size of "M"

%

relative to parent size

vw, vh

relative to viewport

Functions

`calc(100% / 12)`

formula

`attr(name)`

attribute value

`url("backg.jpg")`

external reference

`var(--my-var)`

variable

Further Topics: layout

display

inline, block, none,...

position

static, relative,

absolute, fixed, sticky

float, clear

left, right

Further Topics: layout

flexbox *<http://flexboxfroggy.com>*

grid *https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout*
<https://codepip.com/games/grid-garden/>

Further Topics: color

red, blue, ...

color names

#FF0000

hex values

rgba(255,0,0,0.5)

opacity (alpha)

linear-gradient()

direction, color stops

radial-gradient()

shape, color, position

Further Topics

Fonts, Web Fonts, Entities

Animations, Transitions

Media Query (*comes later*)

transform, clip, z-index, svg

flex-box, grid, table-layout

Case Study

<https://dierk.github.io/Home/>