

# INTRODUCTION TO CSS

Prof. Andrew Sheehan

Boston University/MET  
Computer Science Dept.



**Bert Bos**  
Co-creator of CSS



**Håkon Wium Lie**  
Co-creator of CSS



Cascading Style Sheets (CSS) is a language used for describing the look and formatting of a document written in HTML

A STANDARD

# BENEFITS

We get faster downloads after 1<sup>st</sup> time with external css files.






Separation of style and structure and greater control of the presentation tier.

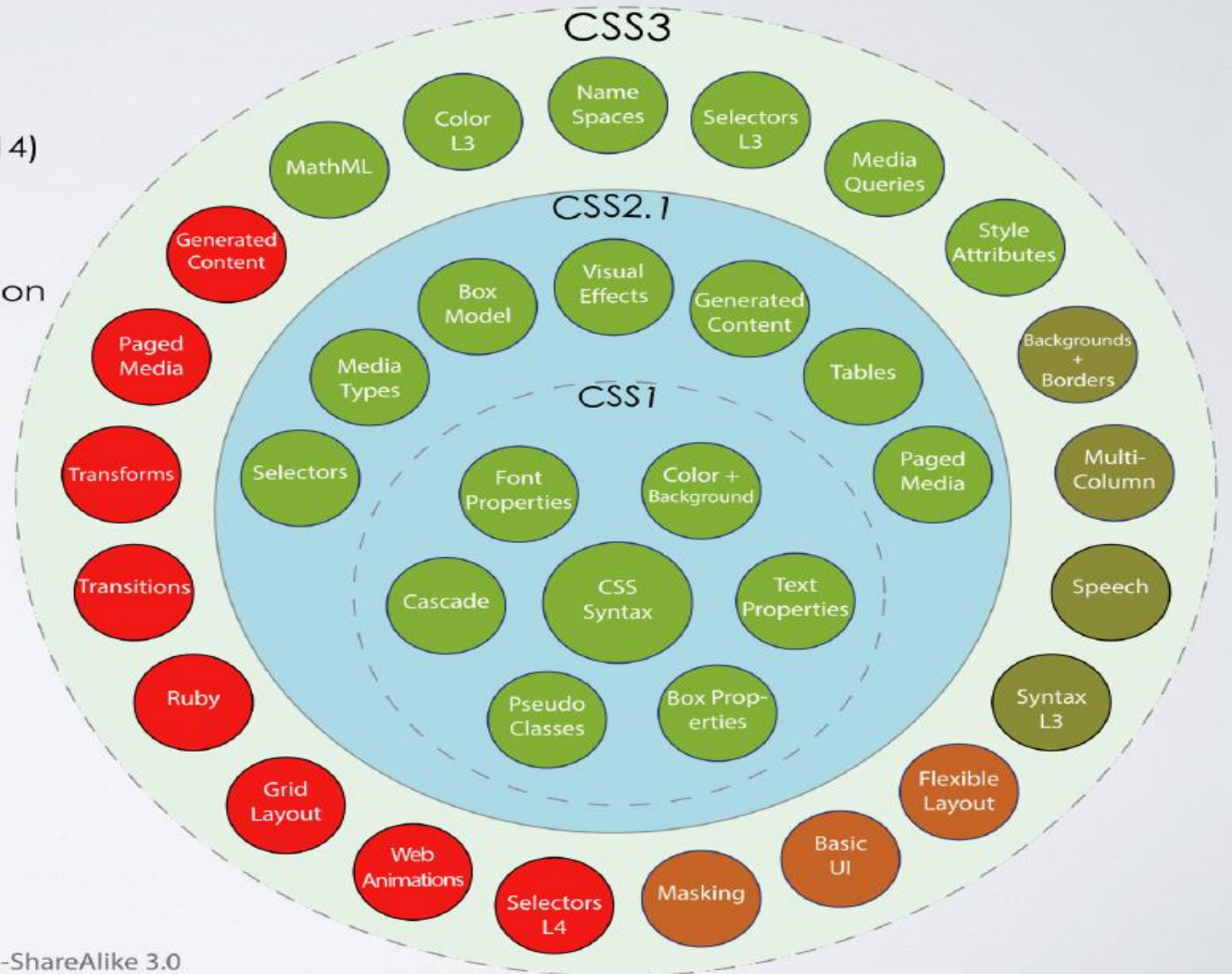
Simpler page means your site will be indexed/categorized better by Bing, Google and other search engines

Each level of CSS builds on the previous, refining definitions and adding more features.

# CSS3

Taxonomy & Status (October 2014)

-  W3C Recommendation
-  Candidate Recommendation
-  Last Call
-  Working Draft
-  Obsolete or inactive



# ***CSS***



NOT FULLY SUPPORTED YET



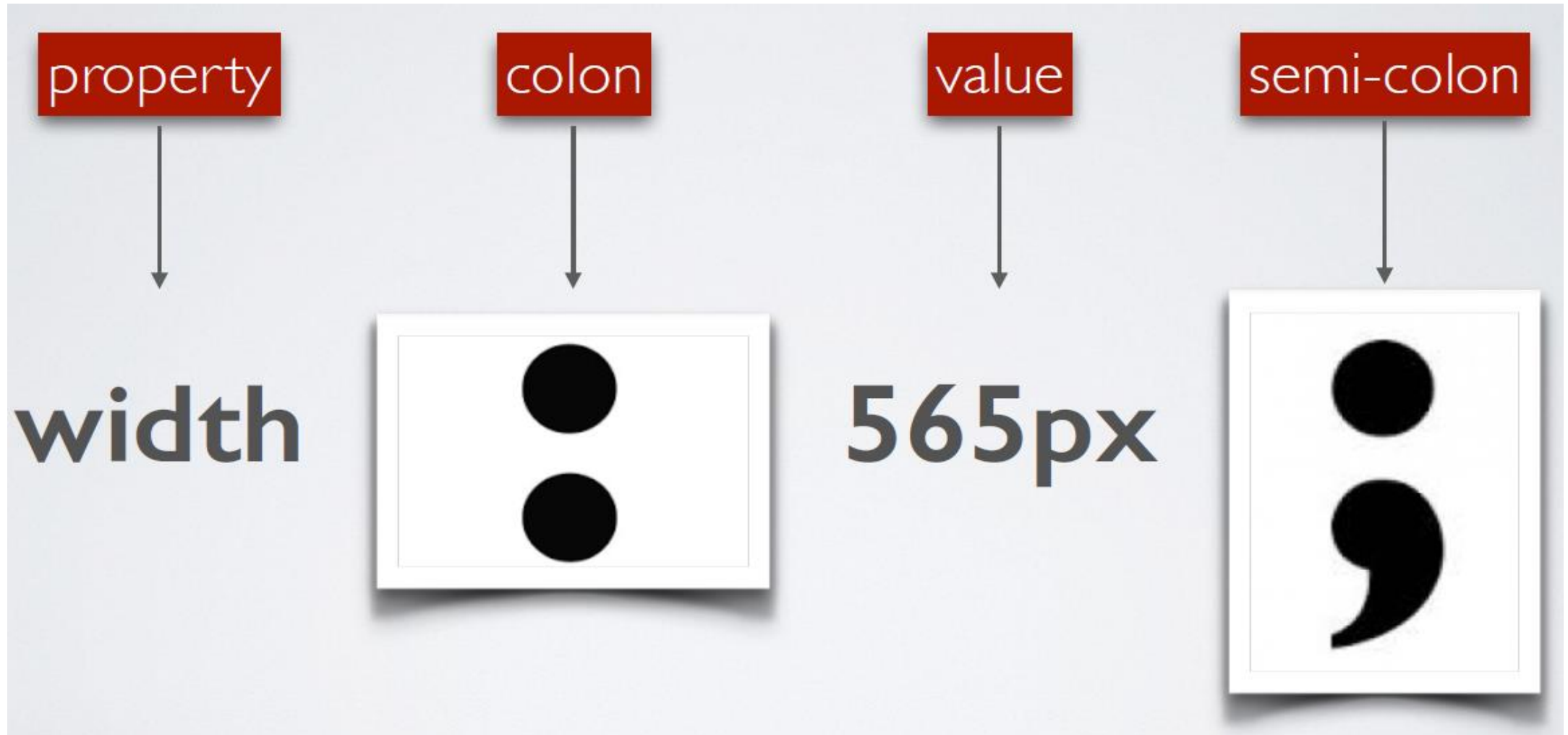


Your clients should not  
find your problems.

Test before you  
release.

TEST. TEST. TEST.

# PROPERTY DECLARATION/SYNTAX





Selector

h1

Declaration

Declaration

{color:blue; font-size:12px;}

Property

Value

Property

Value

SYNTAX



# Linked

`<link href="../../styles/style.css" type="text/css" rel="stylesheet"/>`



# Embedded

`<style>....</style>`



# Inline

`<div style="float: right; width: 545px; height: 455px">...</div>`

# CHANGING

Declaring a rule at the beginning of your CSS can be updated or changed below it's declaration

```
hr {  
  color: blue;  
}
```



```
hr {  
  color: red;  
}
```



# AUGMENTING

Declaring a rule can be added to later down in your CSS

```
hr {  
  color: blue;  
}
```



```
hr {  
  color: blue;  
  width: 80%;  
  letter-spacing: 2px;  
}
```



# USER AGENTS AND CSS

```
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = False  
operation == "MIRROR_Y":  
    mirror_mod.use_x = False  
    mirror_mod.use_y = True  
    mirror_mod.use_z = False  
    operation == "MIRROR_Z":  
    mirror_mod.use_x = False  
    mirror_mod.use_y = False  
    mirror_mod.use_z = True  
  
#selection at the end -add  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier_ob.name))  
mirror_ob.select = 0  
= bpy.context.selected_object  
data.objects[one.name].select  
  
print("please select exactly one mirror")  
  
-- OPERATOR CLASSES --
```

Browsers have defaults. They have to.  
Every property has a value of some type.

# THE RUBRIC USED IN CSS DETERMINATIONS

The CSS specification describes a set of rules to determine which style rules apply if more than one rule matches against a particular element.

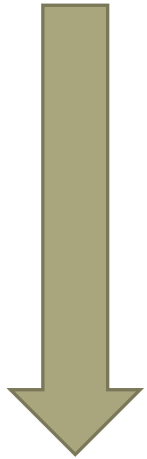
Priorities or weights are calculated and assigned. Results are reliable and predictable.





# ORDER OF IMPORTANCE

highest



lowest

Author inline styles

Author embedded styles

Author external style sheet

User style sheet

Default browser style sheet



# ORDER MATTERS

```
1. p {color: black;}  
2. ul {border: 1px solid pink;}  
3. p.intro {color: brown;}  
4. p {color: red;}
```

In the code above, we have created rules for paragraphs to be three different colors. Clearly, these rules conflict with one another.

Rule #3 is the most specific because it specifies all paragraphs that also have the class attribute value of `intro`.

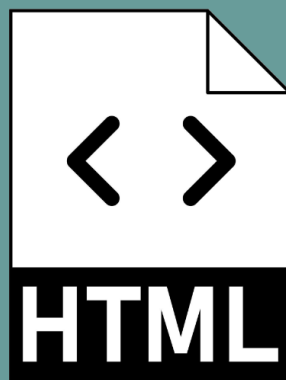
Rules #1 and #4 conflict. They are from the the same style sheet and they have the same level of specificity. If all else is equal in the cascade, we **sort by order specified**.

Rule #4 is declared last in the CSS document and therefore, overrides the previously declared Rule #1.



`<link>`

Linked CSS rules will be trumped by embedded and finally inline CSS trumps them all



`<style>`

`<p style=""></p>`

THE CASCADE

# THE BOX MODEL



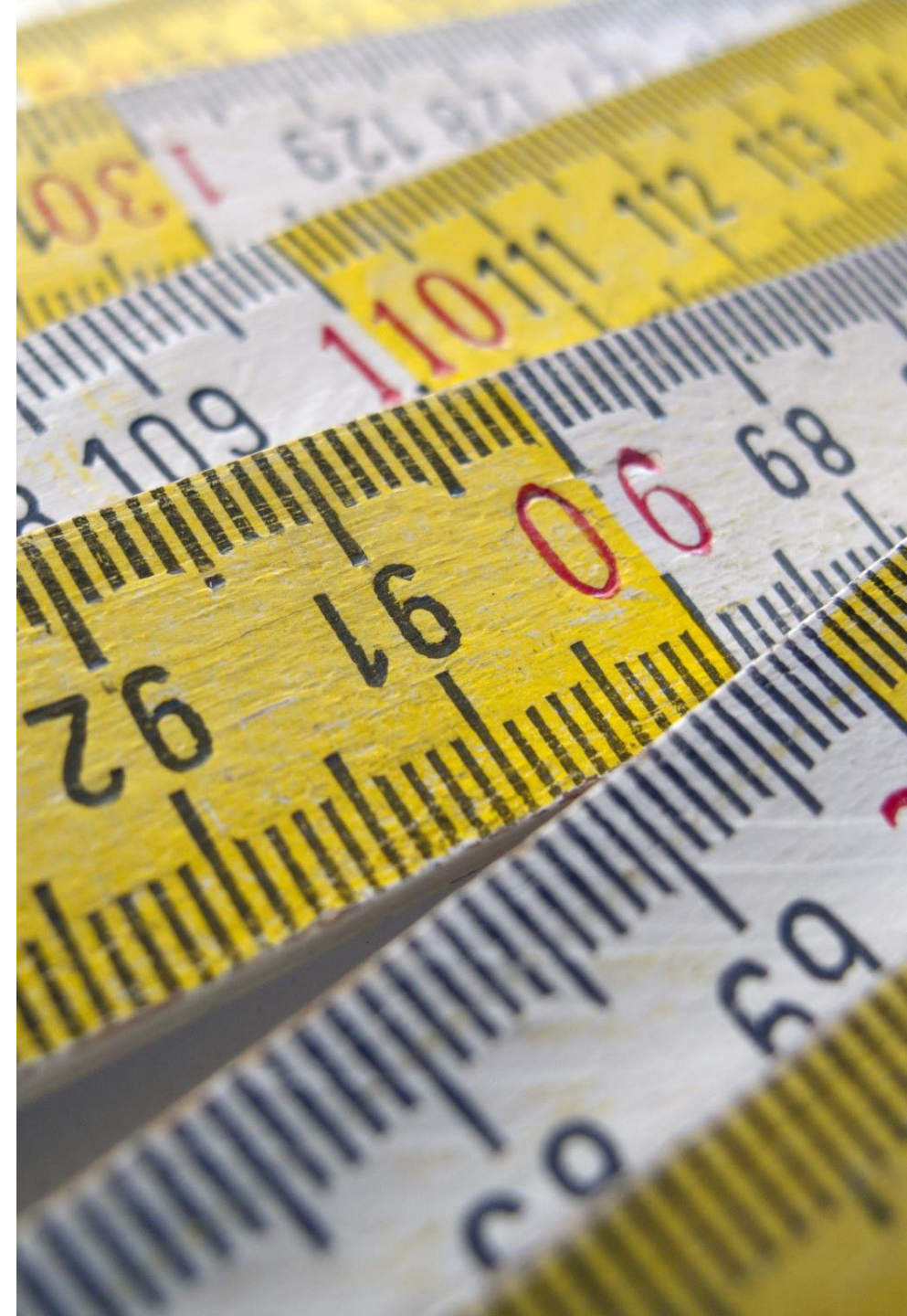
# IMPORTANT

*/\* The important rule can be used to  
override your styling rules. \*/*

```
p > span.odd {  
    font-weight: bold !important;  
}
```

developer.mozilla.org/en-US/docs/Learn/CSS/Building\_blocks/Values\_and\_units

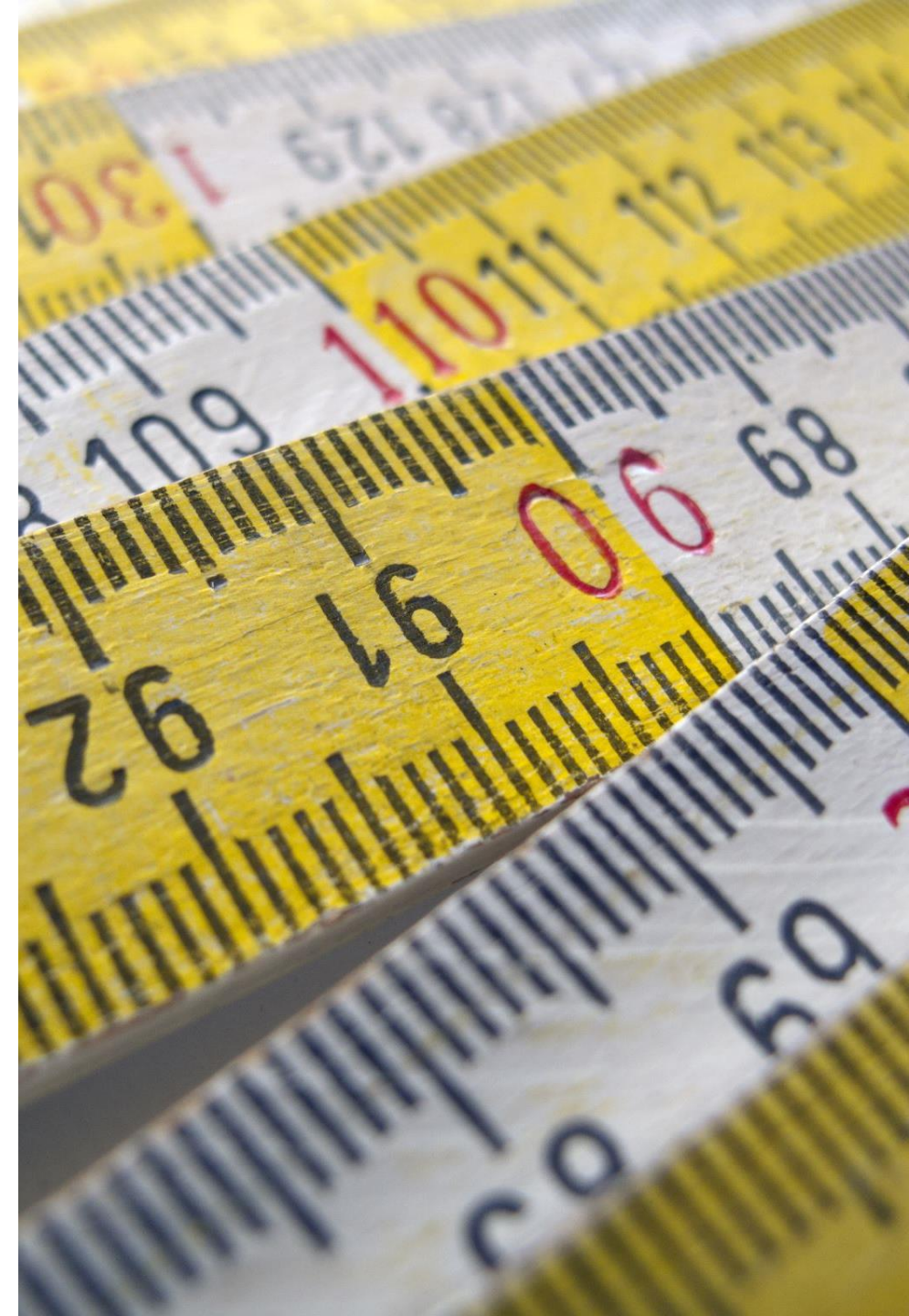
# MEASUREMENTS





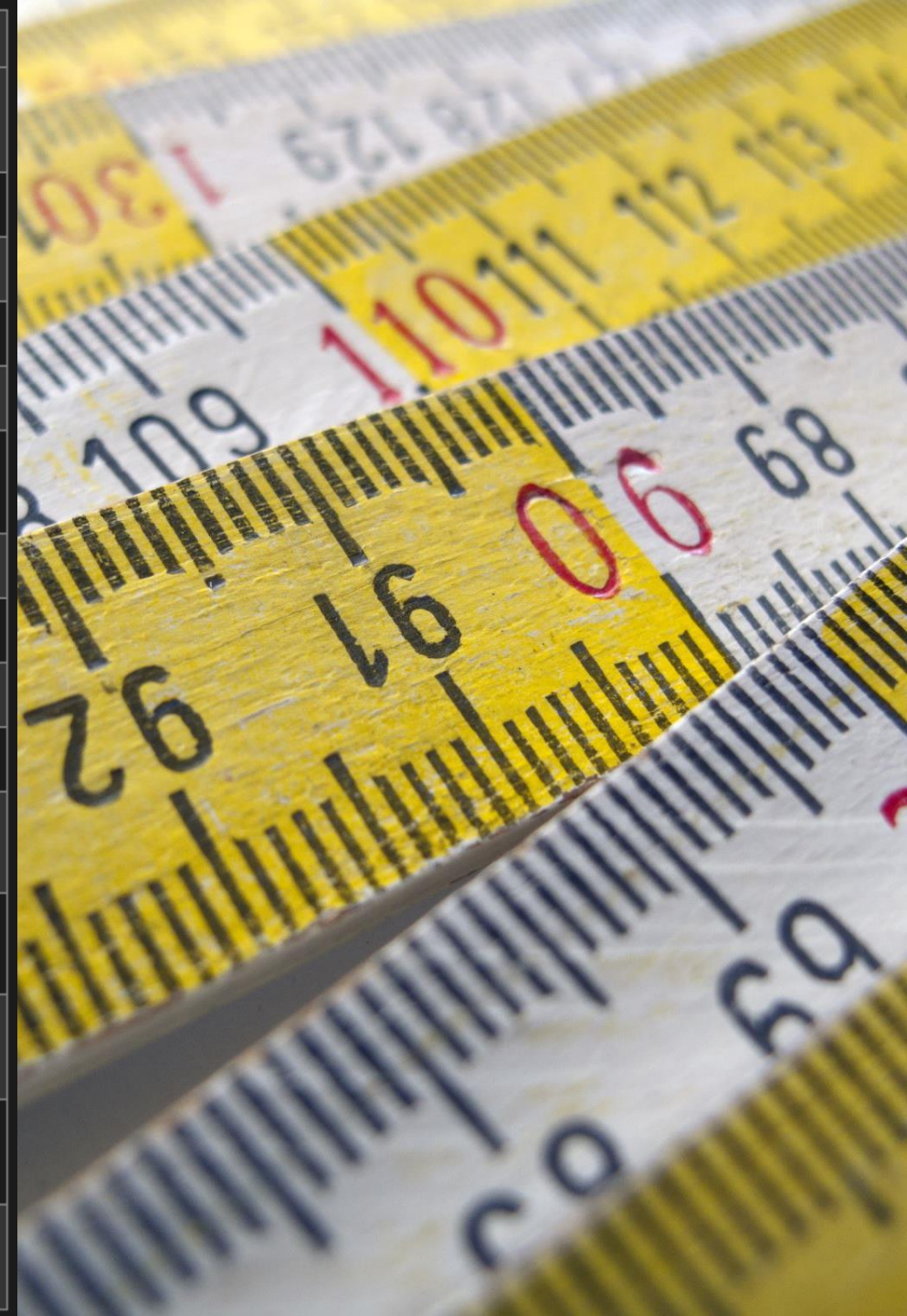
# ABSOLUTE MEASUREMENTS

Unit	Name	Equivalent to
cm	Centimeters	1cm = 37.8px = 25.2/64in
mm	Millimeters	1mm = 1/10th of 1cm
Q	Quarter-millimeters	1Q = 1/40th of 1cm
in	Inches	1in = 2.54cm = 96px
pc	Picas	1pc = 1/6th of 1in
pt	Points	1pt = 1/72nd of 1in
px	Pixels	1px = 1/96th of 1in




# RELATIVE MEASUREMENTS

Unit	Relative to
em	Font size of the parent, in the case of typographical properties like <a href="#">font-size</a> , and font size of the element itself, in the case of other properties like <a href="#">width</a> .
ex	x-height of the element's font.
ch	The advance measure (width) of the glyph "0" of the element's font.
rem	Font size of the root element.
1h	Line height of the element.
rlh	Line height of the root element. When used on the <a href="#">font-size</a> or <a href="#">line-height</a> properties of the root element, it refers to the properties' initial value.
vw	1% of the viewport's width.
vh	1% of the viewport's height.
vmin	1% of the viewport's smaller dimension.
vmax	1% of the viewport's larger dimension.
vb	1% of the size of the initial containing block in the direction of the root element's <a href="#">block axis</a> .
vi	1% of the size of the initial containing block in the direction of the root element's <a href="#">inline axis</a> .
svw, svh	1% of the <a href="#">small viewport</a> 's width and height, respectively.
1vw, 1vh	1% of the <a href="#">large viewport</a> 's width and height, respectively.
dvw, dvh	1% of the <a href="#">dynamic viewport</a> 's width and height, respectively.





# VALIDATOR.W3.ORG

**Markup Validation Service**  
Check the markup (HTML, XHTML, ...) of Web documents

Validate by URI

Validate by File Upload

Validate by Direct Input

Validate by URI

Validate a document online:

Address:

► More Options

Check

# CANIUSE.COM

Can I use

? ⚙ Settings

✕ Feature: CSS Grid Layout (level 1)

🔍

# CSS Grid Layout (level 1) 📄 - CR

Usage % of all users ?

Global 96.32% + 0.49% = 96.81%

unprefixed: 96.32%

☆ Method of using a grid concept to lay out content, providing a mechanism for authors to divide available space for layout into columns and rows using a set of predictable sizing behaviors. Includes support for all `grid-*` properties and the `fr` unit.

Current aligned

Usage relative

Date relative

Filtered

All ⚙

Chrome	Edge	Safari	Firefox	Opera	IE	Chrome for Android	Safari on iOS	Samsung Internet	Opera Mini	Chrome Mobile
4-28			2-39							
1 29-56			3 40-51	10-27						
4 57	2 12-15	3.1-10	4 52-53	1 28-43	6-9		3.2-10.2	4-5.4		
58-103	16-103	10.1-15.5	54-103	44-89	2 10		10.3-15.5	6.2-17.0		12
104	104	15.6	104	90	2 11	104	15.6	18.0	all	
105-107		16.0-TP	105-106				16.0			

Notes

Test on a real browser

Known issues (3)

Resources (8)

Feedback

# UNSUPPORTED CSS?

Your CSS will be ignored if the CSS engine cannot interpret it.

Nothing.