

CSS

FLEXBOX & CSS GRID

# Context



**IN A ROCKET**

Learn front-end development at *rocket speed*



+



# CSS3

# CSS4

CSS?

+80

SPECIFICATIONS

# ALL CSS SPECIFICATIONS

The screenshot shows a web browser window titled "All CSS specifications". The page header includes the W3C logo, the text "Descriptions of all CSS specifications", and a "Languages" section with flags for various countries. A navigation menu contains links for "CSS HOME", "LEARN & USE", "SOFTWARE", and "PARTICIPATE / DRAFTS". A search bar is located on the right side of the header.

**CSS SPECIFICATIONS**

This page contains descriptions of all specifications that the [CSS WG](#) is working on. See the 'current work' page for a compact view and [how to give feedback](#).

**MEDIA QUERIES**

*Media Queries* is an enhancement of the @media rules of CSS and the "media" attribute in HTML. It adds parameters such as size of display, color depth and aspect ratio. This is because within a class of media (such as TV sets) there can still be important variations. It is related to the work on CC/PP, but is a much more light-weight and limited solution.

Editors: Florian Rivoal, Håkon Wium-Lie, Tantek Çelik, Daniel Glazman, Anne van Kesteren

**CSS FLEXIBLE BOX LAYOUT LEVEL 1**

The *Flexible Box Layout Module* defines the 'flex' and 'inline-flex' keywords for the 'display' property, which cause an element to be displayed as either a column or a row of child elements. Additional properties determine the order of the child boxes (left to right, bottom to top, etc.) and how space is distributed over the children and the spaces between them. The module is primarily intended for forcing rows of controls in a GUI to equal height or width.

Editors: Tab Atkins Jr., Erika J. Etemad / fantasai, Rossen Atanassov

**CSS IMAGE VALUES AND REPLACED CONTENT MODULE LEVEL 3**

The *Image Values Module* defines how properties can refer to images by URI. All properties that can take

# MATURITY OF SPECIFICATIONS BY STATUS CODE

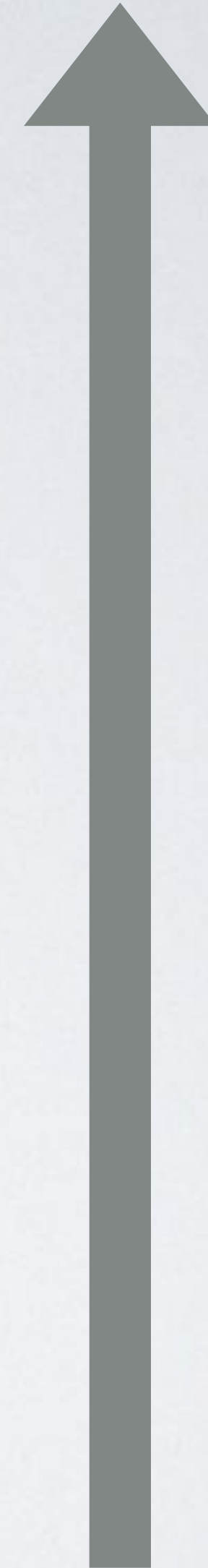
**REC** RECOMMENDATION

**PR** PROPOSED RECOMMENDATION

**CR** CANDIDATE RECOMMENDATION

**WD** WORKING DRAFT

**FPWD** FIRST PUBLIC WORKING DRAFT





Ordered from most to least stable:

<b>Completed</b>	<b>Current</b>	<b>Upcoming</b>	<b>Notes</b>	
CSS Snapshot 2018	NOTE		Latest stable CSS	<input type="checkbox"/>
CSS Snapshot 2017	NOTE			<input type="checkbox"/>
CSS Snapshot 2015	NOTE			<input type="checkbox"/>
CSS Snapshot 2010	NOTE			<input type="checkbox"/>
CSS Snapshot 2007	NOTE			<input type="checkbox"/>
CSS Color Level 3	REC	REC		<input type="checkbox"/>
CSS Namespaces	REC	REC		<input type="checkbox"/>
Selectors Level 3	REC	REC		<input type="checkbox"/>
CSS Level 2 Revision 1	REC	REC	See Errata	<input type="checkbox"/>
Media Queries	REC	REC		<input type="checkbox"/>
CSS Style Attributes	REC	REC		<input type="checkbox"/>
CSS Fonts Level 3	REC	REC		<input type="checkbox"/>
CSS Writing Modes Level 3	REC	REC		<input type="checkbox"/>
CSS Basic User Interface Level 3	REC	REC		<input type="checkbox"/>
CSS Containment Level 1	REC	REC		<input type="checkbox"/>
<b>Stable</b>	<b>Current</b>	<b>Upcoming</b>	<b>Notes</b>	
CSS Backgrounds and Borders Level 3	CR	PR		<input type="checkbox"/>
CSS Conditional Rules Level 3	CR	CR		<input type="checkbox"/>
CSS Multi-column Layout Level 1	WD	CR		<input type="checkbox"/>
CSS Values and Units Level 3	CR	PR		<input type="checkbox"/>
CSS Flexible Box Layout Level 1	CR	PR		<input type="checkbox"/>
CSS Cascading and Inheritance Level 3	CR	PR		<input type="checkbox"/>
CSS Counter Styles Level 3	CR	PR		<input type="checkbox"/>

**FLEXBOX**



**CSS GRID**



## CSS SPECIFICATIONS

This page contains descriptions of all specifications that the [CSS Working Group](#) is currently working on. See the ['current work' page](#) for a compact view and [how to give feedback](#).



## CSS FLEXIBLE BOX LAYOUT LEVEL 1

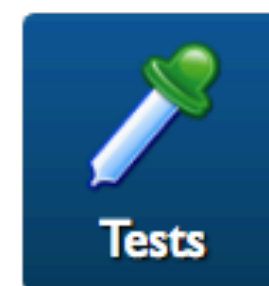
The *Flexible Box Layout Module* defines the 'flex' and 'inline-flex' keywords for the 'display' property, which cause an element to be displayed as either a column or a row of child elements. Additional properties determine the order of the child boxes (left to right, bottom to top, etc.) and how space is distributed over the children and the spaces between them. The module is primarily intended for forcing rows of controls in a GUI to equal height or width.



Editors: Tab Atkins Jr., Erika J. Etemad / fantasai, Rossen Atanassov

## MEDIA QUERIES

*Media Queries* is an enhancement of the @media rules of CSS and the "media" attribute in HTML. It adds parameters such as size of display, color depth and aspect ratio. This is because within a class of media (such as TV sets) there can still be important variations. It is related to the work on CC/PP, but is a much more light-weight and limited solution.

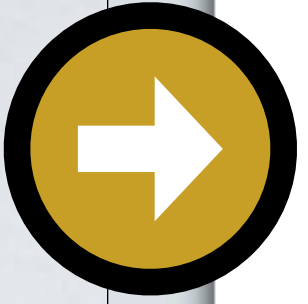


Editors: Florian Rivoal, Håkon Wium Lie, Tantek Çelik, Daniel Glazman, Anne van Kesteren

## CSS IMAGE VALUES AND REPLACED CONTENT MODULE LEVEL 3

The *Image Values Module* defines how properties can refer to images by URL. All properties that can take





Testing	Current	Up-coming	Notes	
CSS Image Values and Replaced Content Level 3	CR	CR		i <input type="radio"/>
CSS Speech	CR	CR		i <input type="radio"/>
CSS Flexible Box Layout Level 1	CR	PR		i <input type="radio"/>
CSS Text Decoration Level 3	CR	CR		i <input type="radio"/>
CSS Shapes Level 1	CR	CR		i <input type="radio"/>
CSS Masking Level 1	CR	CR		i <input type="radio"/>
CSS Fragmentation Level 3	CR	PR		i <input type="radio"/>
CSS Cascading Variables	CR	PR		i <input type="radio"/>
Compositing and Blending Level 1	CR	CR		i <input type="radio"/>
CSS Syntax Level 3	CR	CR		i <input type="radio"/>
CSS Grid Layout Level 1	CR	PR		i <input type="radio"/>
CSS Will Change Level 1	CR	PR		i <input type="radio"/>
Media Queries Level 4	CR	PR		i <input type="radio"/>
Geometry Interfaces Level 1	CR	CR		i <input type="radio"/>
CSS Cascading and Inheritance Level 4	CR	PR		i <input type="radio"/>
CSS Scroll Snap Level 1	CR	PR		i <input type="radio"/>
CSS Painting API Level 1	CR	PR		i <input type="radio"/>
CSS Containment Level 1	CR	PR		i <input type="radio"/>
CSS Writing Modes Level 4	CR	PR		i <input type="radio"/>

W3C indicates the maturity of specifications by a status code. The CSS working group uses the following, from *least* to *most stable*:

Abbreviation	Full name
FPWD	First Public Working Draft
WD	Working Draft
CR	Candidate Recommendation
PR	Proposed Recommendation
REC	Recommendation

The names are defined in [section 6](#) of the W3C process document. A REC is what is normally referred to as a 'standard.' W3C encourages everyday use starting from CR.

The informal stability levels used to group the specs are defined in [this 2007 description of CSS stability levels](#).

## IF YOU WANT TO HELP

Everybody can take part in the discussions on the [archived mailing list `www-style@w3.org`](#). You can [subscribe](#) yourself. This is the preferred place for discussions, because the members of the working group will see them. Please, don't use this list for questions of the type *How do I...* Use [comp.infosystems.www.authoring.stylesheets](#) ("ciwas") or see '[Learning CSS](#)'.

On the mailing list, you will be talking to many people, many very busy people. Before you post, please, search the [archive](#) to see if your great idea has maybe already been discussed. Follow the usual [netiquette](#) and

CSS  
FONTS

**CSS FLEXIBLE BOX  
LAYOUT MODULE**

LEVEL 4



LEVEL 3



LEVEL 2



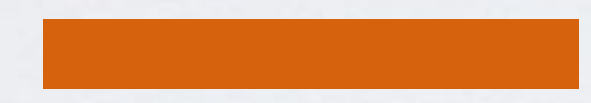
LEVEL 1



# CSS FONTS

# CSS FLEXIBLE BOX LAYOUT MODULE

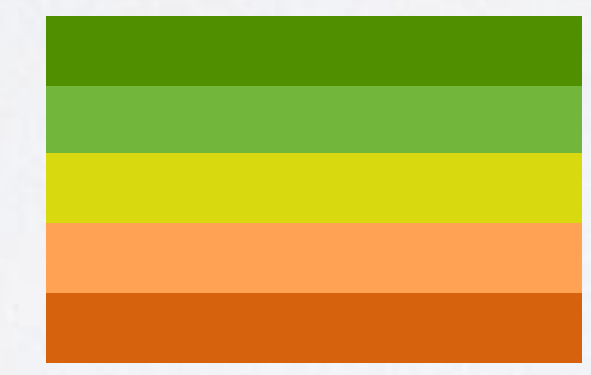
LEVEL 4



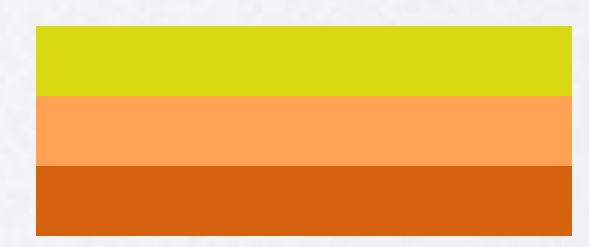
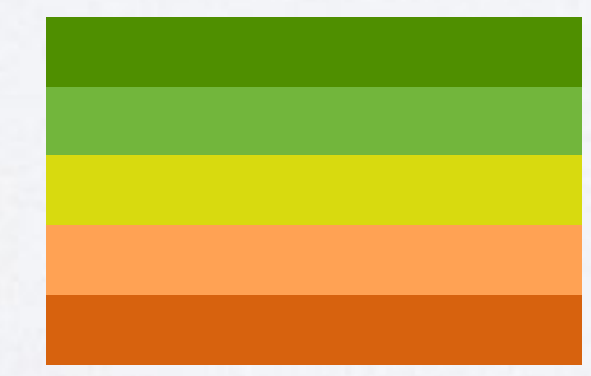
LEVEL 3



LEVEL 2



LEVEL 1



Candidate recommendation

## TABLE OF CONTENTS

<b>1</b>	<b>Introduction</b>
1.1	Overview
1.2	Module interactions
<b>2</b>	<b>Flex Layout Box Model and Terminology</b>
<b>3</b>	<b>Flex Containers: the 'flex' and 'inline-flex' 'display' values</b>
<b>4</b>	<b>Flex Items</b>
4.1	Absolutely-Positioned Flex Children
4.2	Flex Item Margins and Paddings
4.3	Flex Item Z-Ordering
4.4	Collapsed Items
4.5	Automatic Minimum Size of Flex Items
<b>5</b>	<b>Ordering and Orientation</b>
5.1	Flex Flow Direction: the 'flex-direction' property
5.2	Flex Line Wrapping: the 'flex-wrap' property
5.3	Flex Direction and Wrap: the 'flex-flow' shorthand
5.4	Display Order: the 'order' property
5.4.1	Reordering and Accessibility
<b>6</b>	<b>Flex Lines</b>
<b>7</b>	<b>Flexibility</b>
7.1	The 'flex' Shorthand
7.1.1	Basic Values of 'flex'
7.2	Components of Flexibility
7.2.1	The 'flex-grow' property
7.2.2	The 'flex-shrink' property

# CSS Flexible Box Layout Module Level 1

W3C Candidate Recommendation, 19 November 2018



## This version:

<https://www.w3.org/TR/2018/CR-css-flexbox-1-20181119/>

## Latest published version:

<https://www.w3.org/TR/css-flexbox-1/>

## Editor's Draft:

<https://drafts.csswg.org/css-flexbox-1/>

## Previous Versions:

<https://www.w3.org/TR/2018/CR-css-flexbox-1-20181108/>

<https://www.w3.org/TR/2017/CR-css-flexbox-1-20171019/>

<https://www.w3.org/TR/2016/CR-css-flexbox-1-20160526/>

<https://www.w3.org/TR/2016/CR-css-flexbox-1-20160301/>

<https://www.w3.org/TR/2015/WD-css-flexbox-1-20150514/>

<https://www.w3.org/TR/2014/WD-css-flexbox-1-20140925/>

<https://www.w3.org/TR/2014/WD-css-flexbox-1-20140325/>

<https://www.w3.org/TR/2012/CR-css3-flexbox-20120918/>

<https://www.w3.org/TR/2012/WD-css3-flexbox-20120612/>

<https://www.w3.org/TR/2012/WD-css3-flexbox-20120322/>

<https://www.w3.org/TR/2011/WD-css3-flexbox-20111129/>

<https://www.w3.org/TR/2011/WD-css3-flexbox-20110322/>

<https://www.w3.org/TR/2009/WD-css3-flexbox-20090723/>

## Test Suite:

[http://test.csswg.org/suites/css-flexbox-1\\_dev/nightly-unstable/](http://test.csswg.org/suites/css-flexbox-1_dev/nightly-unstable/)

## Editors:

[Tab Atkins Jr.](#) (Google)

[Elika J. Etemad / fantasai](#) (Invited Expert)

[Rossen Atanassov](#) (Microsoft)

## Former Editors:

## TABLE OF CONTENTS

1	Introduction
1.1	Overview
1.2	Module interactions

# CSS Flexible Box Layout Module Level 1

W3C Candidate Recommendation, 19 November 2018



This version:

## 1. Introduction

*This section is not normative.*

CSS 2.1 defined four layout modes — algorithms which determine the size and position of boxes based on their relationships with their sibling and ancestor boxes:

- block layout, designed for laying out documents
- inline layout, designed for laying out text
- table layout, designed for laying out 2D data in a tabular format
- positioned layout, designed for very explicit positioning without much regard for other elements in the document

This module introduces a new layout mode, **flex layout**, which is designed for laying out more complex applications and webpages.

7	Flexibility
7.1	The 'flex' Shorthand
7.1.1	Basic Values of 'flex'
7.2	Components of Flexibility
7.2.1	The 'flex-grow' property
7.2.2	The 'flex-shrink' property

### Editors:

[Tab Atkins Jr.](#) (Google)  
[Elika J. Etemad / fantasai](#) (Invited Expert)  
[Rossen Atanassov](#) (Microsoft)

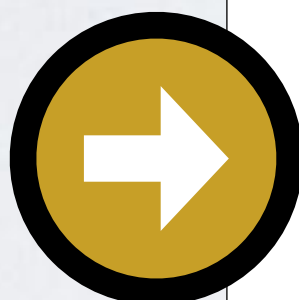
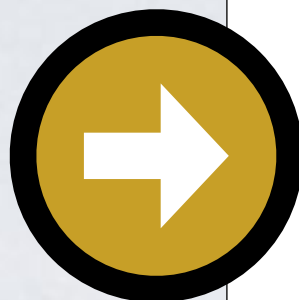
### Former Editors:



**FLEXBOX**



**CSS GRID**



<b>Testing</b>	<b>Current</b>	<b>Upcoming</b>	<b>Notes</b>	
CSS Images Level 3	CR	CR		<input type="checkbox"/> <input type="checkbox"/>
CSS Speech	CR	CR		<input type="checkbox"/> <input type="checkbox"/>
CSS Text Decoration Level 3	CR	CR		<input type="checkbox"/> <input type="checkbox"/>
CSS Shapes Level 1	CR	CR		<input type="checkbox"/> <input type="checkbox"/>
CSS Masking Level 1	CR	CR		<input type="checkbox"/> <input type="checkbox"/>
CSS Fragmentation Level 3	CR	PR		<input type="checkbox"/> <input type="checkbox"/>
CSS Transforms Level 1	CR	PR		<input type="checkbox"/> <input type="checkbox"/>
CSS Cascading Variables	CR	PR		<input type="checkbox"/> <input type="checkbox"/>
Compositing and Blending Level 1	CR	CR		<input type="checkbox"/> <input type="checkbox"/>
CSS Syntax Level 3	CR	CR		<input type="checkbox"/> <input type="checkbox"/>
CSS Grid Layout Level 1	CRD	CR		<input type="checkbox"/> <input type="checkbox"/>
CSS Display Level 3	CR	CR		<input type="checkbox"/> <input type="checkbox"/>
CSS Will Change Level 1	CR	PR		<input type="checkbox"/> <input type="checkbox"/>
Media Queries Level 4	CR	PR		<input type="checkbox"/> <input type="checkbox"/>
Geometry Interfaces Level 1	CR	CR		<input type="checkbox"/> <input type="checkbox"/>
CSS Cascading and Inheritance Level 4	CR	PR		<input type="checkbox"/> <input type="checkbox"/>
CSS Scroll Snap Level 1	CR	PR		<input type="checkbox"/> <input type="checkbox"/>
CSS Painting API Level 1	CR	PR		<input type="checkbox"/> <input type="checkbox"/>
CSS Easing Functions Level 1	CR	WD		<input type="checkbox"/> <input type="checkbox"/>
CSS Writing Modes Level 4	CR	PR		<input type="checkbox"/> <input type="checkbox"/>
CSS Grid Layout Level 2	CRD	CR		<input type="checkbox"/> <input type="checkbox"/>
<b>Refining</b>	<b>Current</b>	<b>Upcoming</b>	<b>Notes</b>	
CSS Animations Level 1	WD	WD		<input type="checkbox"/> <input type="checkbox"/>
Web Animations	WD	WD		<input type="checkbox"/> <input type="checkbox"/>
CSS Text Level 3	WD	CR		<input type="checkbox"/> <input type="checkbox"/>
CSS Transitions	WD	CR		<input type="checkbox"/> <input type="checkbox"/>
CSS Box Alignment Level 3	WD	WD		<input type="checkbox"/> <input type="checkbox"/>
Selectors Level 4	WD	WD		<input type="checkbox"/> <input type="checkbox"/>

## TABLE OF CONTENTS

<b>1</b>	<b>Introduction</b>
1.1	Background and Motivation
1.1.1	Adapting Layouts to Available Space
1.1.2	Source-Order Independence
1.2	Value Definitions
<b>2</b>	<b>Overview</b>
2.1	Declaring the Grid
2.2	Placing Items
2.3	Sizing the Grid
<b>3</b>	<b>Grid Layout Concepts and Terminology</b>
3.1	Grid Lines
3.2	Grid Tracks and Cells
3.3	Grid Areas
<b>4</b>	<b>Reordering and Accessibility</b>
<b>5</b>	<b>Grid Containers</b>
5.1	Establishing Grid Containers: the 'grid' and 'inline-grid' 'display' values
5.2	Sizing Grid Containers
5.3	Scrollable Grid Overflow
5.4	Limiting Large Grids
<b>6</b>	<b>Grid Items</b>
6.1	Grid Item Display
6.2	Grid Item Sizing
6.3	Reordered Grid Items: the 'order' property
6.4	Grid Item Margins and Paddings
6.5	Z-axis Ordering: the 'z-index' property
6.6	Automatic Minimum Size of Grid Items

# CSS Grid Layout Module Level 1

W3C Candidate Recommendation, 21 October 2020



Draft

**This version:**

<https://www.w3.org/TR/2020/CRD-css-grid-1-20201021/>

**Latest published version:**

<https://www.w3.org/TR/css-grid-1/>

**Editor's Draft:**

<https://drafts.csswg.org/css-grid/>

**Previous Versions:**

<https://www.w3.org/TR/2020/CR-css-grid-1-20200818/>

<https://www.w3.org/TR/2017/CR-css-grid-1-20171214/>

<https://www.w3.org/TR/2017/CR-css-grid-1-20170509/>

<https://www.w3.org/TR/2017/CR-css-grid-1-20170209/>

<https://www.w3.org/TR/2016/CR-css-grid-1-20160929/>

<https://www.w3.org/TR/2016/WD-css-grid-1-20160519/>

<https://www.w3.org/TR/2015/WD-css-grid-1-20150917/>

<https://www.w3.org/TR/2015/WD-css-grid-1-20150806/>

<https://www.w3.org/TR/2015/WD-css-grid-1-20150317/>

<https://www.w3.org/TR/2014/WD-css-grid-1-20140513/>

<https://www.w3.org/TR/2014/WD-css-grid-1-20140123/>

<https://www.w3.org/TR/2013/WD-css3-grid-layout-20130910/>

<https://www.w3.org/TR/2013/WD-css3-grid-layout-20130402/>

<https://www.w3.org/TR/2012/WD-css3-grid-layout-20121106/>

<https://www.w3.org/TR/2012/WD-css3-grid-layout-20120322/>

<https://www.w3.org/TR/2011/WD-css3-grid-layout-20110407/>

**Implementation Report:**

<https://wpt.fyi/results/css/css-grid>

**Test Suite:**

## TABLE OF CONTENTS

- 1 Introduction**
- 1.1 Background and Motivation
  - 1.1.1 Adapting Layouts to Available Space
  - 1.1.2 Source-Order Independence
- 1.2 Value Definitions
  
- 2 Overview**
- 2.1 Declaring the Grid
- 2.2 Placing Items

# CSS Grid Layout Module Level 1

W3C Candidate Recommendation, 21 October 2020



Draft

**This version:**

<https://www.w3.org/TR/2020/CRD-css-grid-1-20201021/>

**Latest published version:**

<https://www.w3.org/TR/css-grid-1/>

## 1. Introduction

*This section is not normative.*

Grid Layout is a new layout model for CSS that has powerful abilities to control the sizing and positioning of boxes and their contents. Unlike Flexible Box Layout, which is single-axis-oriented, Grid Layout is optimized for 2-dimensional layouts: those in which alignment of content is desired in both dimensions.

- 5.2 Sizing Grid Containers
- 5.3 Scrollable Grid Overflow
- 5.4 Limiting Large Grids
  
- 6 Grid Items**
- 6.1 Grid Item Display
- 6.2 Grid Item Sizing
- 6.3 Reordered Grid Items: the 'order' property
- 6.4 Grid Item Margins and Paddings
- 6.5 Z-axis Ordering: the 'z-index' property
- 6.6 Automatic Minimum Size of Grid Items

<https://www.w3.org/TR/2015/WD-css-grid-1-20150317/>

<https://www.w3.org/TR/2014/WD-css-grid-1-20140513/>

<https://www.w3.org/TR/2014/WD-css-grid-1-20140123/>

<https://www.w3.org/TR/2013/WD-css3-grid-layout-20130910/>

<https://www.w3.org/TR/2013/WD-css3-grid-layout-20130402/>

<https://www.w3.org/TR/2012/WD-css3-grid-layout-20121106/>

<https://www.w3.org/TR/2012/WD-css3-grid-layout-20120322/>

<https://www.w3.org/TR/2011/WD-css3-grid-layout-20110407/>

**Implementation Report:**

<https://wpt.fyi/results/css/css-grid>

**Test Suite:**

## TABLE OF CONTENTS

<b>1</b>	<b>Introduction</b>
1.1	Background and Motivation
1.1.1	Adapting Layouts to Available Space
1.1.2	Source-Order Independence
1.2	Value Definitions
<b>2</b>	<b>Overview</b>
2.1	Declaring the Grid
2.2	Placing Items
2.3	Sizing the Grid
<b>3</b>	<b>Grid Layout Concepts and Terminology</b>
3.1	Grid Lines
3.2	Grid Tracks and Cells
3.3	Grid Areas
3.4	Nested vs. Subgridded Items
<b>4</b>	<b>Reordering and Accessibility</b>
<b>5</b>	<b>Grid Containers</b>
5.1	Establishing Grid Containers: the 'grid' and 'inline-grid' 'display' values
5.2	Sizing Grid Containers
5.3	Scrollable Grid Overflow
5.4	Limiting Large Grids
<b>6</b>	<b>Grid Items</b>
6.1	Grid Item Display
6.2	Grid Item Sizing
6.3	Reordered Grid Items: the 'order' property
6.4	Grid Item Margins and Paddings
6.5	Z-axis Ordering: the 'z-index' property

# CSS Grid Layout Module Level 2

W3C Candidate Recommendation, 21 October 2020



Draft

**This version:**

<https://www.w3.org/TR/2020/CRD-css-grid-2-20201021/>

**Latest published version:**

<https://www.w3.org/TR/css-grid-2/>

**Editor's Draft:**

<https://drafts.csswg.org/css-grid-2/>

**Previous Versions:**

<https://www.w3.org/TR/2020/CR-css-grid-2-20200818/>

<https://www.w3.org/TR/2019/WD-css-grid-2-20191203/>

<https://www.w3.org/TR/2018/WD-css-grid-2-20180804/>

<https://www.w3.org/TR/2018/WD-css-grid-2-20180628/>

**Implementation Report:**

<https://wpt.fyi/results/css/css-grid/subgrid>

**Issue Tracking:**

[CSSWG GitHub](#)

[Inline In Spec](#)

[GitHub Issues](#)

**Editors:**

[Tab Atkins Jr.](#) (Google)

[Elika J. Etemad / fantasai](#) (Invited Expert)

[Rossen Atanassov](#) (Microsoft)

**Suggest an Edit for this Spec:**

[GitHub Editor](#)

Copyright © 2020 W3C® (MIT, ERCIM, Keio, Beihang). W3C liability, trademark and permissive document license rules apply.

## TABLE OF CONTENTS

<b>1</b>	<b>Introduction</b>
1.1	Background and Motivation
1.2	Value Definitions
<b>2</b>	<b>Masonry Layout</b>
2.1	Line Name Resolution
2.2	Grid Item Placement
2.3	Grid Axis Track Sizing
2.3.1	repeat(auto-fit)
2.4	Masonry Layout Algorithm
2.5	The 'masonry-auto-flow' Property
<b>3</b>	<b>Containing Block</b>
<b>4</b>	<b>The Implicit Grid</b>
<b>5</b>	<b>Sizing Grid Containers</b>
<b>6</b>	<b>Alignment and Spacing</b>
6.1	The 'align-tracks' and 'justify-tracks' Properties
6.2	Stretch Alignment in the Masonry Axis
6.3	Individual Track Alignment in the Masonry Axis
6.4	Baseline Alignment in the Masonry Axis
<b>7</b>	<b>Fragmentation</b>
7.1	Fragmentation in the Masonry Axis
7.2	Fragmentation in the Grid Axis
<b>8</b>	<b>Subgrids</b>
<b>9</b>	<b>Absolute Positioning</b>

# CSS Grid Layout Module Level 3

Editor's Draft, 22 October 2020



## ► Specification Metadata

Copyright © 2020 W3C® (MIT, ERCIM, Keio, Beihang). W3C [liability](#), [trademark](#) and [permissive document license](#) rules apply.

## Abstract

This module introduces masonry layout as an additional layout mode for [CSS Grid](#) containers.

[CSS](#) is a language for describing the rendering of structured documents (such as HTML and XML) on screen, on paper, etc.

## Status of this document

This is a public copy of the editors' draft. It is provided for discussion only and may change at any moment. Its publication here does not imply endorsement of its contents by W3C. Don't cite this document other than as work in progress.

Please send feedback by [filing issues in GitHub](#) (preferred), including the spec code "css-grid" in the title, like this: "[css-grid] ...summary of comment...". All issues and comments are [archived](#). Alternately, feedback can be sent to the [\(archived\)](#) public mailing list [www-style@w3.org](mailto:www-style@w3.org).

This document is governed by the [15 September 2020 W3C Process Document](#).

## § 1. Introduction

*This section is not normative.*

# CSS FLEXIBLE BOX LAYOUT MODULE

# CSS GRID LAYOUT MODULE

LEVEL 4



LEVEL 3



LEVEL 2



LEVEL 1



# CSS FLEXIBLE BOX LAYOUT MODULE

# CSS GRID LAYOUT MODULE

LEVEL 4

LEVEL 3

LEVEL 2

LEVEL 1



Candidate recommendation



Candidate recommendation





+



**FLEXBOX**



**CSS GRID**

## Can I use flexible box

[?](#) [Settings](#)

1 result found

### # Flexible Box Layout Module - CR

 Global 84.07% + 13.6% = 97.68%

 unprefixed: 83.57% + 3.27% = 86.83%

Method of positioning elements in horizontal or vertical stacks. Support includes the support for the all properties prefixed with flex as well as `display: flex`, `display: inline-flex`, `align-content`, `align-items`, `align-self`, `justify-content` and `order`.

[Current aligned](#) [Usage relative](#) [Date relative](#) [Show all](#)

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			49						
			56			9.3		4.4	
	14	52	57	10		10.2		4.4.4	
11	15	53	58	10.1	44	10.3	all	56	57
		54	59	TP	45				
		55	60		46				
		56	61						

[Notes](#)
[Known issues \(8\)](#)
[Resources \(13\)](#)
[Feedback](#)

Home News Nov 30, 2017 - 2 new features added Compare browsers About

## Can I use CSS Flexible Box

1 result found

# CSS Flexible Box Layout Module - CR


Global 85.09% + 12.31% = 97.39%  
unprefixed: 84.77% + 3.25% = 88.02%

Method of positioning elements in horizontal or vertical stacks. Support includes all properties prefixed with `flex`, as well as `display: flex`, `display: inline-flex`, `align-content`, `align-items`, `align-self`, `justify-content` and `order`.

Current aligned Usage relative Date relative Show all

IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
		52	49						
		55	60		10.2				4
	15	56	61	10.1	10.3				5
4 11	16	57	62	11	11.1	all	62	11.4	6.2
	17	58	63	TP					
		59	64						
		60	65						

Notes Known issues (9) Resources (13) Feedback





**statcounter**  
GlobalStats

[Press Releases](#) [FAQ](#) [About](#) [Feedback](#)

Chrome

58.94%

Safari

13.7%

UC Browser

7.46%

Firefox

5.17%

Opera

3.5%

IE

3.12%

Browser Market Share Worldwide - June 2018



UC Browser

## Can I use CSS Flexible Box

[?](#) [Settings](#)

1 result found

### # CSS Flexible Box Layout Module - CR

Method of positioning elements in horizontal or vertical stacks. Support includes all properties prefixed with `flex`, as well as `display: flex`, `display: inline-flex`, `align-content`, `align-items`, `align-self`, `justify-content` and `order`.

Usage % of all users   
 Global 93.2% + 3.47% = 96.68%  
 unprefixed: 92.92% + 2.75% = 95.67%

Current aligned Usage relative Date relative [Show all](#)

IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
			49						
			66		10.3				4
		60	67		11.2				6.2
4 11	17	61	68	11.1	11.4	all	67	11.8	7.2
	18	62	69	12	12				
		63	70	TP					
			71						


[Notes](#)
[Known issues \(9\)](#)
[Resources \(13\)](#)
[Feedback](#)

**FLEXBOX**



**CSS GRID**

Can I use

css grid

? Settings

1 result found

## # CSS Grid Layout WD

Global

0% + 6.8% = 6.8%

unprefixed:

0%

Method of using a grid concept to lay out content, providing a mechanism for authors to divide available space for lay out into columns and rows using a set of predictable sizing behaviors

Current aligned

Usage relative

Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			1 49						
		3 47	1 51					4.4	
8	2 13	3 48	1 52	9.1		9.3		4.4.4	
2 11	2 14	3 49	1 53	10	1 39	10	all	52	51
		3 50	1 54	4 TP	1 40				
		3 51	1 55		1 41				
		3 52	1 56						

Notes

Known issues (0)

Resources (10)

Feedback

Supported in WebKit Nightly with `-webkit-` prefix.

Enabled by default in Firefox nightly and developer editions, but not yet on track to be enabled in beta or stable Firefox.



## Can I use

## css grid

[?](#) [Settings](#)

2 results found

### # CSS Grid Layout - CR

 Global 56.99% + 5.35% = 62.34%

 unprefixed: 56.99%

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

[Current aligned](#) [Usage relative](#) [Date relative](#) [Show all](#)

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			49						
			56			9.3		4.4	
	14	52	57	10		10.2		4.4.4	
11	15	53	58	10.1	44	10.3	all	56	57
		54	59	TP	45				
		55	60		46				
		56	61						

[Notes](#)
[Known issues \(0\)](#)
[Resources \(11\)](#)
[Feedback](#)

 Supported in WebKit Nightly with `-webkit-` prefix.

Home News July 29, 2018 - New feature: createImageBitmap Compare browsers About

## Can I use CSS Grid

? Settings

2 results found

# CSS Grid Layout - CR

Usage Global 83.56% + 3.09% = 86.66%

unprefixed: 83.56%

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 Show all

IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
			49						
			63						
			66		10.3				
			67		11.2				4
11	17	61	68	11.1	11.4	all	67	11.8	7.2
	18	62	69	12	12				
		63	70	TP					
			71						

Notes Known issues (2) Resources (13) Feedback

**FLEXBOX**



**CSS GRID**



CSS

FLEXBOX & CSS GRID

# Context



**IN A ROCKET**

Learn front-end development at *rocket speed*