CSS GRID LAYOUT IMPLEMENTATION DETAILS

MANUEL REGO CASASNOVAS (@regocas)

CSS Day / 14 June 2019 (Amsterdam)

ABOUT THIS TALK

*we are **NOT** looking for a general presentation about Grid*

but about one that explains how you actually ***implemented the specification***

*Our audience **already knows Grid***

but what it doesn't know is how easy or hard it is to actually ***implement Grid in a browser***

ABOUT ME

Igalia Web Platform Team

Web engines hacker working on Chromium/Blink and Safari/WebKit

Member of **CSS Working Group** since 2017

ABOUT ME

- CSS Regions
- CSS Grid Layout
- CSS Containment
- `:focus-within`
- `caret-color`
- …

CSS GRID LAYOUT HISTORY

RECENT HISTORY

RECENT HISTORY

Initial spec by Microsoft 2011

RECENT HISTORY

Initial spec by Microsoft 2011

IE10 released a prefixed version of the initial spec in 2012

RECENT HISTORY

Initial spec by Microsoft 2011

IE10 released a prefixed version of the initial spec in 2012

Google started to work on it by the end of 2011

RECENT HISTORY

Initial spec by Microsoft 2011

IE10 released a prefixed version of the initial spec in 2012

Google started to work on it by the end of 2011

Igalia sponsored by Bloomberg started to work on Blink and WebKit implementation by summer 2013

RECENT HISTORY

Initial spec by Microsoft 2011

IE10 released a prefixed version of the initial spec in 2012

Google started to work on it by the end of 2011

Igalia sponsored by Bloomberg started to work on Blink and WebKit implementation by summer 2013

Mozilla started Firefox implementation around 2014

RECENT HISTORY

Initial spec by Microsoft 2011

IE10 released a prefixed version of the initial spec in 2012

Google started to work on it by the end of 2011

Igalia sponsored by Bloomberg started to work on Blink and WebKit implementation by summer 2013

Mozilla started Firefox implementation around 2014

In 2017 Chrome, Firefox and Safari **shipped CSS Grid Layout together**

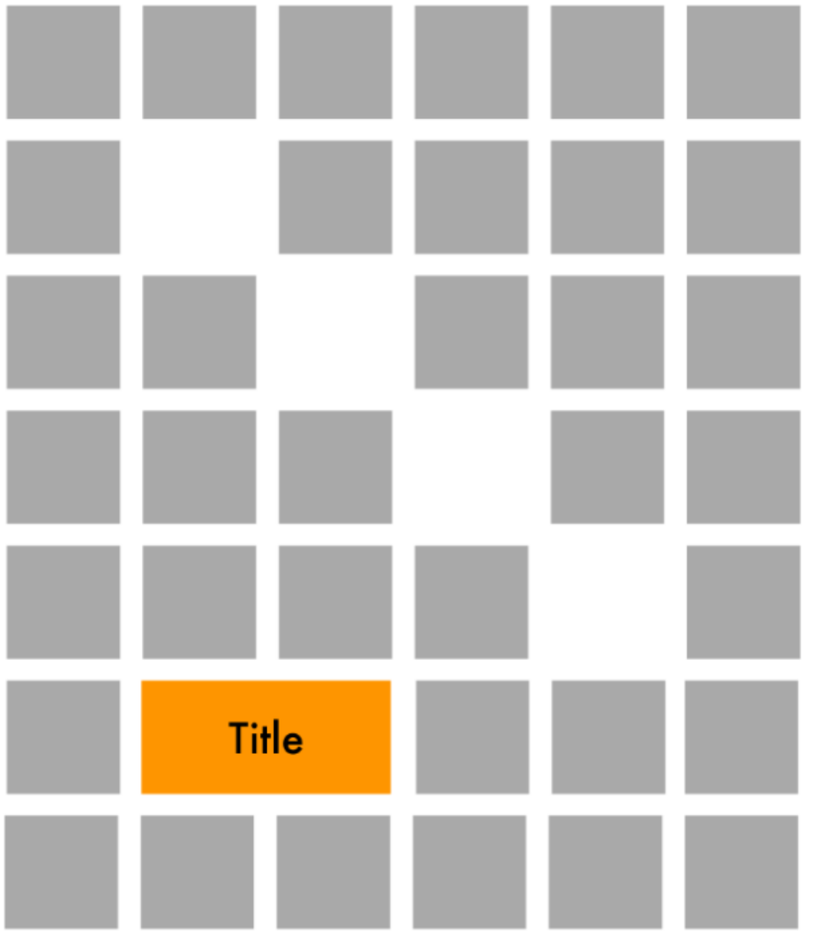
IDEA

"Proposal to CSSWG, Sept 2016" by Jen Simmons

```
@region .container gap1 {
   grid: 2 / 2; }
@region .container gap2 {
   grid: 3 / 3; }
@region .container gap3 {
   grid: 4 / 4; }
@region .container gap4 {
   grid: 5 / 5; }

gap1, gap2, gap3, gap4, {
   grid-auto-flow-self: skip;
}
```

or: none; empty; nope;



IDEA

Explain the process to implement a new CSS property for Grid Layout:

IDEA

Explain the process to implement a new CSS property for Grid Layout:

- CSS Working Group

IDEA

Explain the process to implement a new CSS property for Grid Layout:

- CSS Working Group
- Intent to implement

IDEA

Explain the process to implement a new CSS property for Grid Layout:

- CSS Working Group
- Intent to implement
- Implementation behind a runtime flag

IDEA

Explain the process to implement a new CSS property for Grid Layout:

- CSS Working Group
- Intent to implement
- Implementation behind a runtime flag
- Test suite in Web Platform Tests repository

IDEA

Explain the process to implement a new CSS property for Grid Layout:

- CSS Working Group
- Intent to implement
- Implementation behind a runtime flag
- Test suite in Web Platform Tests repository
- Ship feature!

CSS WORKING GROUP (CSSWG)

GitHub repository:

https://github.com/w3c/csswg-drafts/

NEW CSS PROPERTY

[css-grid] Mark some cells as busy so they are skip in auto-placement #4012

Open **mrego** opened this issue 7 days ago · 0 comments

mrego commented 7 days ago — Member

This idea was already introduced by **@jensimmons** long time ago at TPAC 2016:
https://speakerdeck.com/jensimmons/proposal-to-csswg-sept-2016?slide=51
But I'm just reporting the issue here to track it for future work/proposals around CSS Grid Layout.

I guess it'd be nice to have some kind of syntax that allows us to mark some cells as busy (even without the need to place there empty dummy elements), and the auto-placement algorithm would skip them.

Of course we need to think also in a syntax, so we can refer to a list of cells not a single one.

NAME BIKESHEDDING

NAME BIKESHEDDING

`grid-busy-cells`

NAME BIKESHEDDING

`grid-busy-cells`

`grid-emtpy-cells`

NAME BIKESHEDDING

`grid-busy-cells`

`grid-emtpy-cells`

`grid-taken-areas`

NAME BIKESHEDDING

`grid-busy-cells`

`grid-emtpy-cells`

`grid-taken-areas`

`grid-skip-areas`

NAME BIKESHEDDING

~~`grid-busy-cells`~~

~~`grid-empty-cells`~~

~~`grid-taken-areas`~~

`grid-skip-areas`

GRAMMAR

```
<grid-area> =
<grid-line> [ / <grid-line> ]{0,3}
```

```
<grid-skip-areas> =
none | <grid-area>#
```

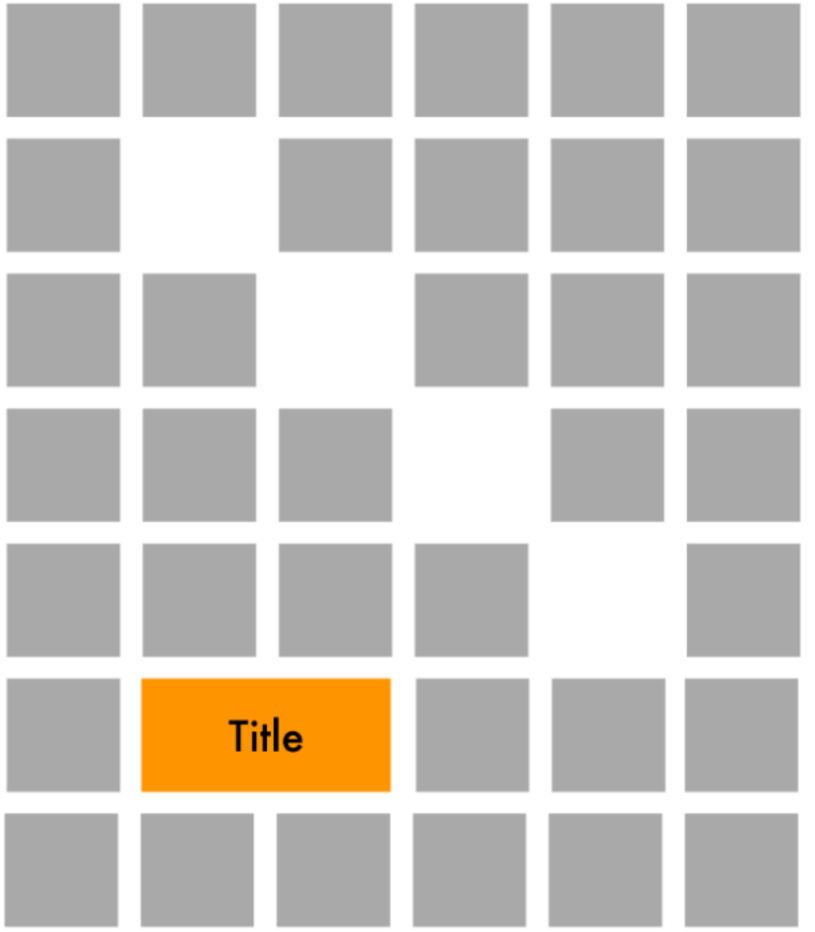
EXAMPLE

`grid-skip-areas: 2/2, 3/3, 4/4, 5/5`

```
@region .container gap1 {
   grid: 2 / 2; }
@region .container gap2 {
   grid: 3 / 3; }
@region .container gap3 {
   grid: 4 / 4; }
@region .container gap4 {
   grid: 5 / 5; }

gap1, gap2, gap3, gap4, {
   grid-auto-flow-self: skip;
}

      or: none; empty; nope;
```



MORE QUESTIONS

MORE QUESTIONS

Can fixed positioned items use a *skipped* area?

MORE QUESTIONS

Can fixed positioned items use a *skipped* area?

How does it interact with `repeat(auto-fit)`?

MORE QUESTIONS

Can fixed positioned items use a *skipped* area?

How does it interact with `repeat(auto-fit)`?

Is there a way to specify skipped areas in `grid-template-areas`?

MORE QUESTIONS

Can fixed positioned items use a *skipped* area?

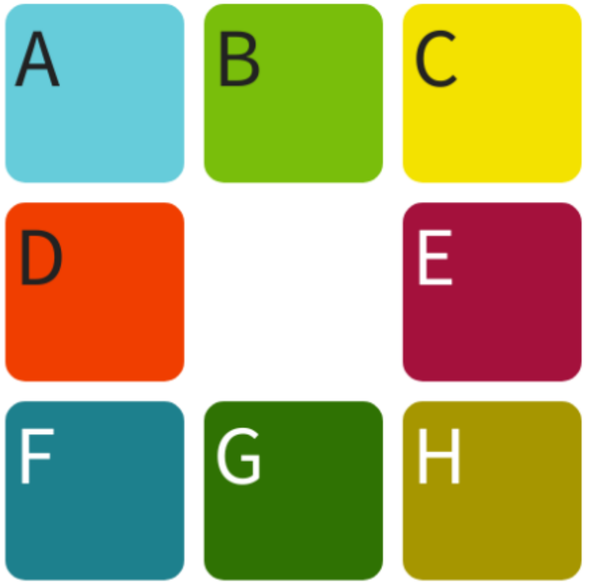
How does it interact with `repeat(auto-fit)`?

Is there a way to specify skipped areas in `grid-template-areas`?

...

NOW

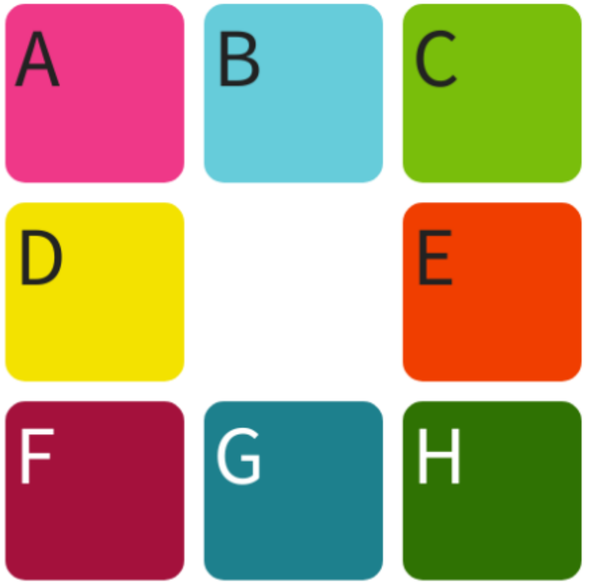
```
<div class="grid">
  <div class="dummy"></div>
  <div>A</div>
  <div>B</div>
  <div>C</div>
  <div>D</div>
  <div>E</div>
  <div>F</div>
  <div>G</div>
  <div>H</div>
</div>
```




```
.grid { display: grid;
        grid: repeat(3, 100px) / repeat(3, 100px); }
.dummy { grid-area: 2 / 2; }
```

FUTURE?

```
<div class="grid">
  <div>A</div>
  <div>B</div>
  <div>C</div>
  <div>D</div>
  <div>E</div>
  <div>F</div>
  <div>G</div>
  <div>H</div>
</div>
```




```
.grid { display: grid;
        grid: repeat(3, 100px) / repeat(3, 100px);
        grid-skip-areas: 2 / 2; }
```

INTENT TO IMPLEMENT

INTENT TO IMPLEMENT

Content: description, motivation, design, specification status, interoperability and compatibility risks, etc.

Mail templates: Chromium, Firefox

W3C Technical Architecture Group (TAG) review

INTENT TO IMPLEMENT

Project mailing list:

- `blink-dev@chromium.org`
- `dev-platform@lists.mozilla.org`
- `webkit-dev@lists.webkit.org`

IMPLEMENTATION

SETUP

Get the Code: Checkout, Build, & Run Chromium

... SOME HOURS LATER ...

SOURCE CODE

`chromium/src/third_party/blink/renderer/core/`

RUNTIME FLAG

`../platform/runtime_enabled_features.json5`

```
1       {
2         name: "CSSDayGridSkipAreas",
3         status: "experimental",
4       },
```

BUILDING PATCH

```
$ time ninja -C src/out/Default blink_tests
ninja: Entering directory `src/out/Default'
[721/721] STAMP obj/blink_tests.stamp

real    10m27.546s
user    55m40.446s
sys     1m54.131s
```

PARSER

```
<grid-skip-areas> =
none | <grid-area>#
```

PARSER

```
<grid-skip-area> = none | <grid-area>
```

REFACTORING

`css/properties/shorthands/shorthands_custom.c`

`GridArea::ParseShorthand()`

```
1  bool GridArea::ParseShorthand(
2      bool important,
3      CSSParserTokenRange& range,
4      const CSSParserContext& context,
5      const CSSParserLocalContext&,
6      HeapVector<CSSPropertyValue, 256>& properties) const {
7    DCHECK_EQ(gridAreaShorthand().length(), 4u);
8
9    CSSValue* row_start_value =
10       css_parsing_utils::ConsumeGridLine(range, context);
11   if (!row_start_value)
12     return false;
```

GridArea::ParseShorthand()

```
1   CSSValue* column_start_value = nullptr;
2   CSSValue* row_end_value = nullptr;
3   CSSValue* column_end_value = nullptr;
4   if (css_property_parser_helpers::ConsumeSlashIncludingWh
5     column_start_value = css_parsing_utils::ConsumeGridLir
6     if (!column_start_value)
7       return false;
8     if (css_property_parser_helpers::ConsumeSlashIncluding
9       row_end_value = css_parsing_utils::ConsumeGridLine(r
10      if (!row_end_value)
11        return false;
12      if (css_property_parser_helpers::ConsumeSlashIncludi
13        column_end_value = css_parsing_utils::ConsumeGridL
14        if (!column_end_value)
```

NEW METHOD

`css/properties/css_parsing_utils.cc`:
`css_parsing_utils::ConsumeGridArea()`

GridArea::ParseShorthand()

```
1  bool GridArea::ParseShorthand(
2      bool important,
3      CSSParserTokenRange& range,
4      const CSSParserContext& context,
5      const CSSParserLocalContext&,
6      HeapVector<CSSPropertyValue, 256>& properties) const {
7    DCHECK_EQ(gridAreaShorthand().length(), 4u);
8
9    CSSValueList* grid_area = css_parsing_utils::ConsumeGridA
10   if (!grid_area)
11     return false;
```

BUILDING PATCH

```
$ time ninja -C src/out/Default blink_tests
ninja: Entering directory `src/out/Default'
[12/12] STAMP obj/blink_tests.stamp

real    0m28.963s
user    1m25.154s
sys     0m4.390s
```

NEW PROPERTY

`css/css_properties.json5`

```
{
    name: "grid-skip-area",
    property_methods: ["ParseSingleValue", "CSSValueFromC
    runtime_flag: "CSSDayGridSkipAreas",
    layout_dependent: true,
    field_group: "*",
    field_template: "external",
    include_paths: ["third_party/blink/renderer/core/styl
    default_value: "GridAreaPositions()",
    type_name: "GridAreaPositions",
    keywords: ["none"],
    typedom_types: ["Keyword"],
    converter: "ConvertGridAreaPositions",
  },
```

NEW CLASS

`style/grid_area_positions.h`

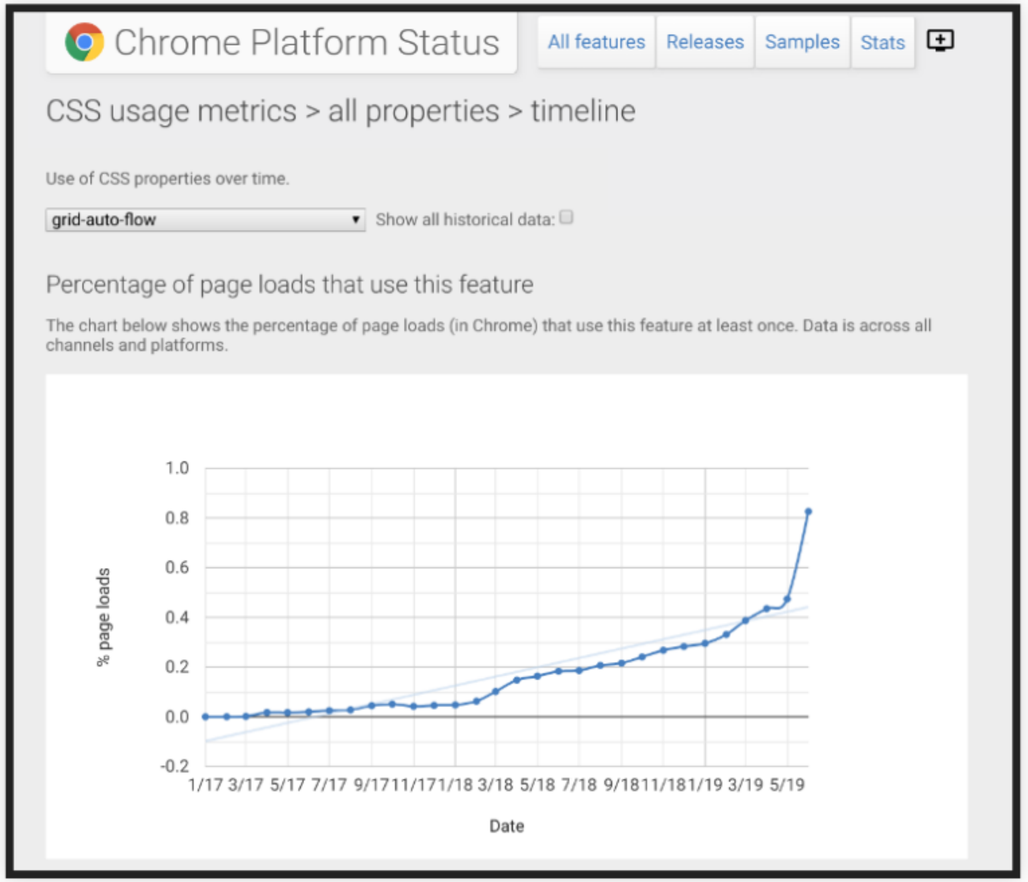
```
1 class GridAreaPositions {
2   ...
3  private:
4   GridPosition row_start_;
5   GridPosition row_end_;
6   GridPosition column_start_;
7   GridPosition column_end_;
8 };
```

USE COUNTERS

`frame/use_counter_helper.cc`

```
1 int usecounterhelper::mapcsspropertyidtocsssampleidforhistog
2     CSSPropertyID unresolved_property) {
3   switch (unresolved_property) {
4     ...
5     case CSSPropertyID::kGridSkipArea:
6       return 641;
```

CHROME PLATFORM STATUS



Chrome Platform Status CSS Metrics

PARSING

css/properties/longhands/longhands_custom.cc

```
1 const CSSValue* GridSkipArea::ParseSingleValue(
2     CSSParserTokenRange& range,
3     const CSSParserContext& context,
4     const CSSParserLocalContext&) const {
5   if (range.Peek().Id() == CSSValueID::kNone)
6     return css_property_parser_helpers::ConsumeIdent(range);
7   return css_parsing_utils::ConsumeGridArea(range, context);
8 }
```

STYLE BUILDER

css/resolver/style_builder_converter.cc

```
1  GridAreaPositions StyleBuilderConverter::ConvertGridAreaPos
2      StyleResolverState& state,
3      const CSSValue& value) {
4    const auto& values = To<CSSValueList>(value);
5    DCHECK_EQ(values.length(), 4u);
6
7    GridPosition row_start = ConvertGridPosition(state, value
8    GridPosition column_start = ConvertGridPosition(state, va
9    GridPosition row_end = ConvertGridPosition(state, values.
10   GridPosition column_end = ConvertGridPosition(state, valu
11
12   return GridAreaPositions(row_start, row_end, column_start
13 }
```

BUILDING PATCH

```
$ time ninja -C src/out/Default blink_tests
ninja: Entering directory `src/out/Default'
[789/789] STAMP obj/blink_tests.stamp
real    13m33.084s
user    94m28.150s
sys     3m5.134s
```

STATUS

Now the browser accepts:

- `grid-skip-area: 3 / 5;`
- `grid-skip-area: 1 / 1 / span 2 / span 2;`
- `grid-skip-area: none;`

But it doesn't do anything with that property

STYLE RESOLVER REFACTORING

style/grid_positions_resolver.cc

```
GridSpan GridPositionsResolver::ResolveGridAreaPositionsFr
    const ComputedStyle& grid_container_style,
    GridTrackSizingDirection direction,
    size_t auto_repeat_tracks_count) {
  GridPosition initial_position =
      direction == kForColumns
          ? grid_container_style.GridSkipArea().ColumnStar
          : grid_container_style.GridSkipArea().RowStart()
  GridPosition final_position =
      direction == kForColumns ? grid_container_style.Gric
                               : grid_container_style.Gric

  return ResolveInitialAndFinalPositionsFromStyle(
      grid_container_style, direction, auto_repeat_tracks_
```

LAYOUT CODE

`layout/layout_grid.cc`

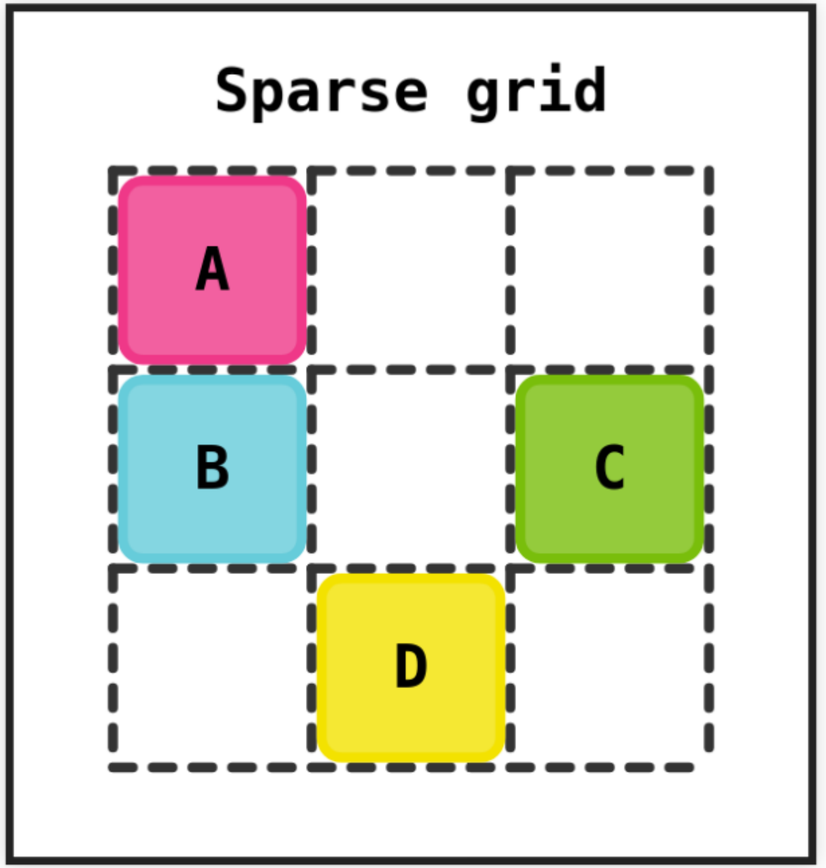
```
1  void LayoutGrid::PlaceItemsOnGrid(
2    ...
3    GridSpan grid_skip_rows =
4        GridPositionsResolver::ResolveGridAreaPositionsFromS
5            *Style(), kForRows, AutoRepeatCountForDirection(
6    GridSpan grid_skip_columns =
7        GridPositionsResolver::ResolveGridAreaPositionsFromS
8            *Style(), kForColumns, AutoRepeatCountForDirecti
9    const GridArea* grid_skip_area = nullptr;
10   if (!grid_skip_rows.IsIndefinite() && !grid_skip_columns
11     grid_skip_rows.Translate(abs(grid.SmallestTrackStart(k
12     grid_skip_columns.Translate(abs(grid.SmallestTrackStar
13     grid_skip_area = new GridArea(grid_skip_rows, grid_ski
14   }
```

MEMORY STRUCTURE

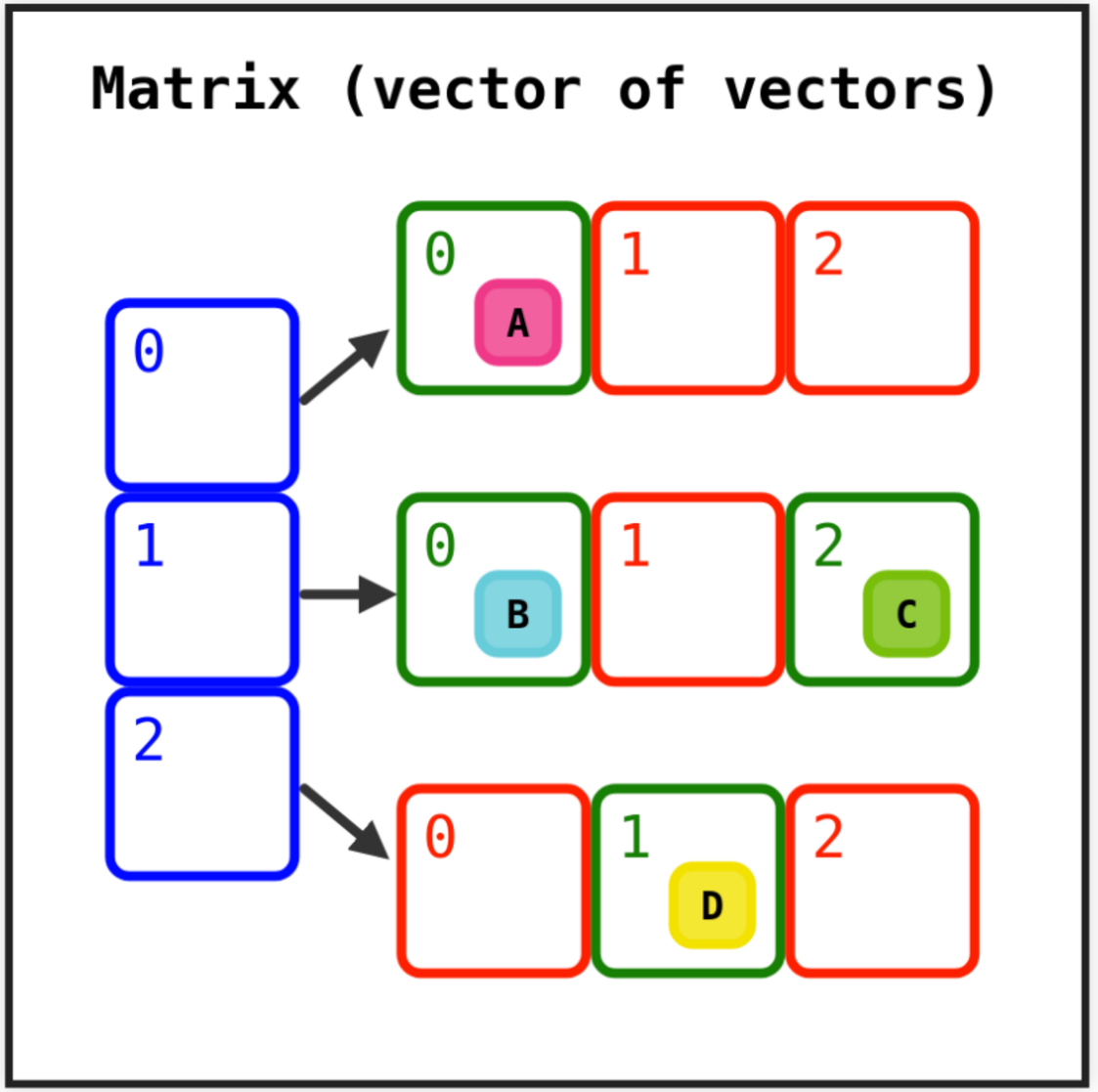
Chromium implementation used to have a matrix (vector of vectors), each grid cell was created even if empty

Now it uses a doubly linked lists, only occupied cells are created

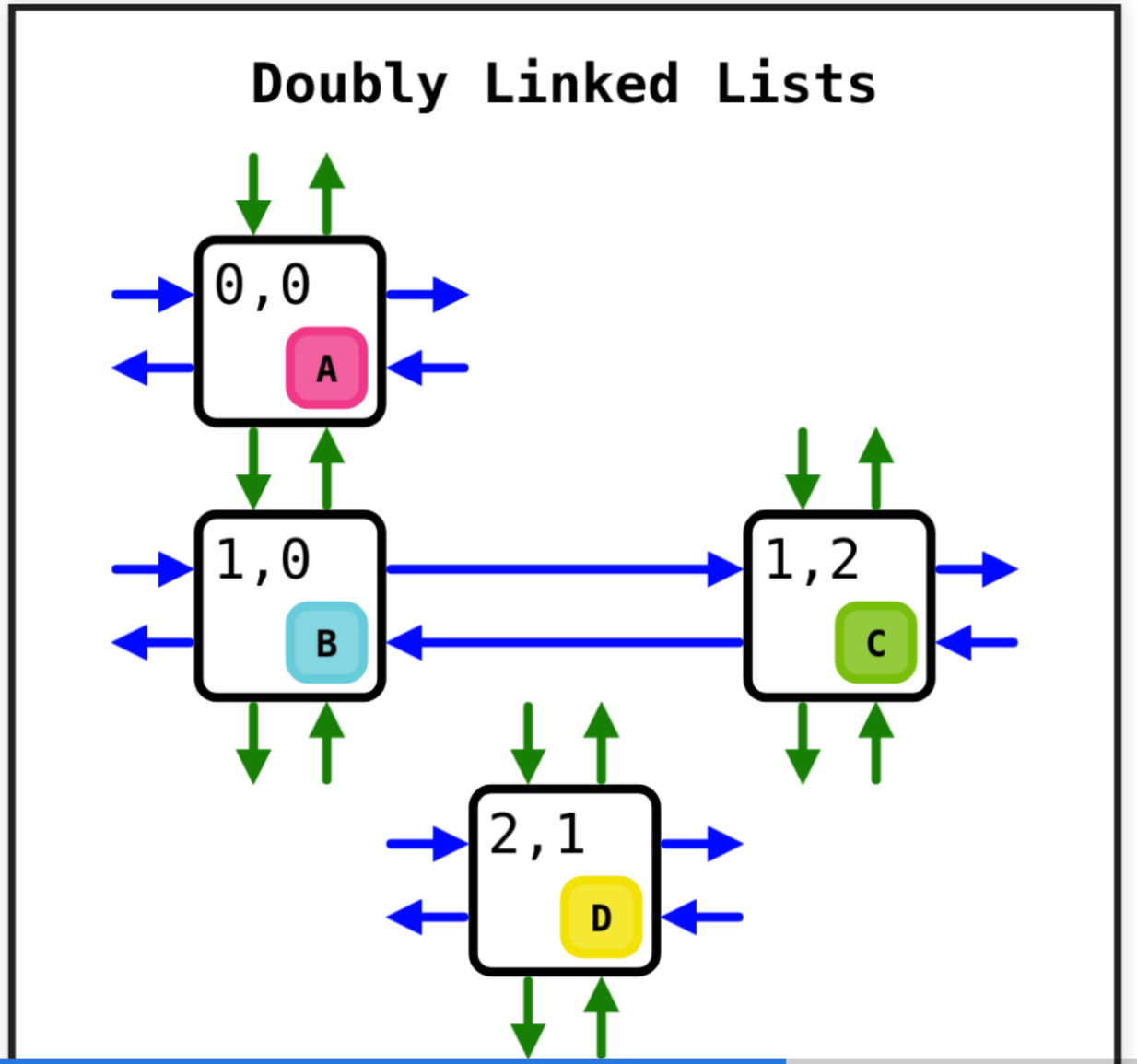
MEMORY STRUCTURE



MEMORY STRUCTURE



MEMORY STRUCTURE



AUTO-PLACEMENT ALGORITHM

`layout/grid.cc`

```
std::unique_ptr<GridArea> ListGridIterator::NextEmptyGridA
    size_t fixed_track_span,
    size_t varying_track_span,
    const GridArea* grid_skip_area) {
  ...
  auto OverlapsGridSkipArea = [this, fixed_track_span, var
                               grid_skip_area]() {
    bool is_row_axis = direction_ == kForColumns;
    size_t row_span = is_row_axis ? varying_track_span : 1
    size_t column_span = is_row_axis ? fixed_track_span :
    return grid_skip_area->Overlaps(GridArea(
        GridSpan::TranslatedDefiniteGridSpan(row_index_, r
        GridSpan::TranslatedDefiniteGridSpan(column_index_
                                             column_index_
```

AUTO-PLACEMENT ALGORITHM

`layout/grid.cc`

```
1 std::unique_ptr<GridArea> ListGridIterator::NextEmptyGridAre
2   ...
3   while (OverlapsGridSkipArea())
4     varying_index++;
5   ...
```

BUILDING PATCH

```
$ time ninja -C src/out/Default blink_tests
ninja: Entering directory `src/out/Default'
[138/138] STAMP obj/blink_tests.stamp

real    5m42.757s
user    40m18.312s
sys     1m13.215s
```

STATUS

`grid-skip-area` is already working, but it only accepts one area

CSS Grid Layout Implementation Details

Back Forward Refresh Stop file:///home/rego/Documents/201906-css-day/example.html#/

EXAMPLE

```
<div class="grid">
  <div>A</div>
  <div>B</div>
  <div>C</div>
  <div>D</div>
  <div>E</div>
  <div>F</div>
  <div>G</div>
  <div>H</div>
</div>
```


```
.grid { display: grid;
        grid: repeat(3, 100px) / repeat(3, 100px);
        grid-skip-area: 2 / 2; }
```

MULTIPLE AREAS

`css/css_properties.json5`

```
@@ -1872,11 +1872,11 @@
        field_group: "*",
        field_template: "external",
        include_paths: ["third_party/blink/renderer/core/sty
-       default_value: "GridAreaPositions()",
-       type_name: "GridAreaPositions",
+       default_value: "Vector<GridAreaPositions>()",
+       type_name: "Vector<GridAreaPositions>",
        keywords: ["none"],
        typedom_types: ["Keyword"],
-       converter: "ConvertGridAreaPositions",
+       converter: "ConvertGridAreaPositionsList",
      },
```

PARSING

css/properties/longhands/longhands_custom.cc

```
const CSSValue* GridSkipArea::ParseSingleValue(
    CSSParserTokenRange& range,
    const CSSParserContext& context,
    const CSSParserLocalContext&) const {
  if (range.Peek().Id() == CSSValueID::kNone)
    return css_property_parser_helpers::ConsumeIdent(range
  CSSValueList* grid_areas = CSSValueList::CreateCommaSepa
  do {
    CSSValue* grid_area =
        css_parsing_utils::ConsumeGridArea(range, context,
    if (!grid_area)
      return nullptr;
    grid_areas->Append(*grid_area);
  } while (css_property_parser_helpers::ConsumeCommaInclud
```

STYLE BUILDER

css/resolver/style_builder_converter.cc

```
1 Vector<GridAreaPositions> StyleBuilderConverter::ConvertGrid
2     StyleResolverState& state,
3     const CSSValue& value) {
4   Vector<GridAreaPositions> list;
5   for (auto& curr_value : To<CSSValueList>(value)) {
6     list.push_back(ConvertGridAreaPositions(state, *curr_val
7   }
8   return list;
9 }
```

AUTO-PLACEMENT ALGORITHM

`layout/grid.cc`

```
std::unique_ptr<GridArea> ListGridIterator::NextEmptyGridA
    size_t fixed_track_span,
    size_t varying_track_span,
    const Vector<GridArea*>& grid_skip_areas) {
  ...
  auto OverlapsGridSkipAreas = [this, fixed_track_span, va
                                grid_skip_areas]() {
    bool is_row_axis = direction_ == kForColumns;
    size_t row_span = is_row_axis ? varying_track_span : 1
    size_t column_span = is_row_axis ? fixed_track_span :
    for (auto* grid_skip_area : grid_skip_areas) {
      if (grid_skip_area->Overlaps(
              GridArea(GridSpan::TranslatedDefiniteGridSpa
                       row_index_, row_index_ + row_sp
```

BUILDING PATCH

```
$ time ninja -C src/out/Default blink_tests
ninja: Entering directory `src/out/Default'
[373/373] STAMP obj/blink_tests.stamp

real    5m54.719s
user    40m34.151s
sys     1m16.931s
```

STATUS

`grid-skip-area` now accepts a list of grid areas and skip them as expected

- `grid-skip-area: 1 / 1, 2 / 2, 3 / 3, 4 / 4;`
- `grid-skip-area: 3 / 5 / span 2 / span 3, 9 / 12;`

CSS Grid Layout Implementation Details

Back Forward Refresh Stop file:///home/rego/Documents/201906-css-day/example.html#/

EXAMPLE

```
<div class="grid">
  <div></div>
  <div></div>
  <div></div>
  ...
  <div></div>
</div>
```

```
.grid { display: grid;
        grid: repeat(4, 100px) / repeat(4, 100px);
        grid-skip-area: 1/1, 2/2, 3/3, 4/4; }
```

RENAME TO

grid-skip-areas

```
$ git grep -l grid-skip-area | xargs sed -i 's/grid-skip-area/
$ git grep -l GridSkipArea | xargs sed -i 's/GridSkipArea/Grid
```

BUILDING PATCH

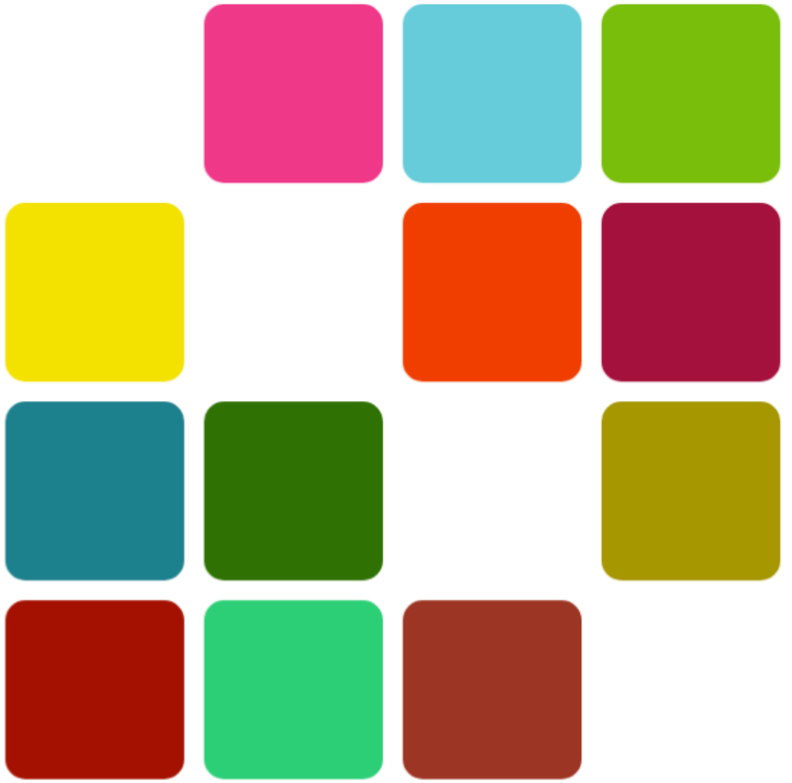
```
$ time ninja -C src/out/Default blink_tests
ninja: Entering directory `src/out/Default'

[550/550] STAMP obj/blink_tests.stamp

real    9m45.056s
user    69m15.646s
sys     2m19.425s
```

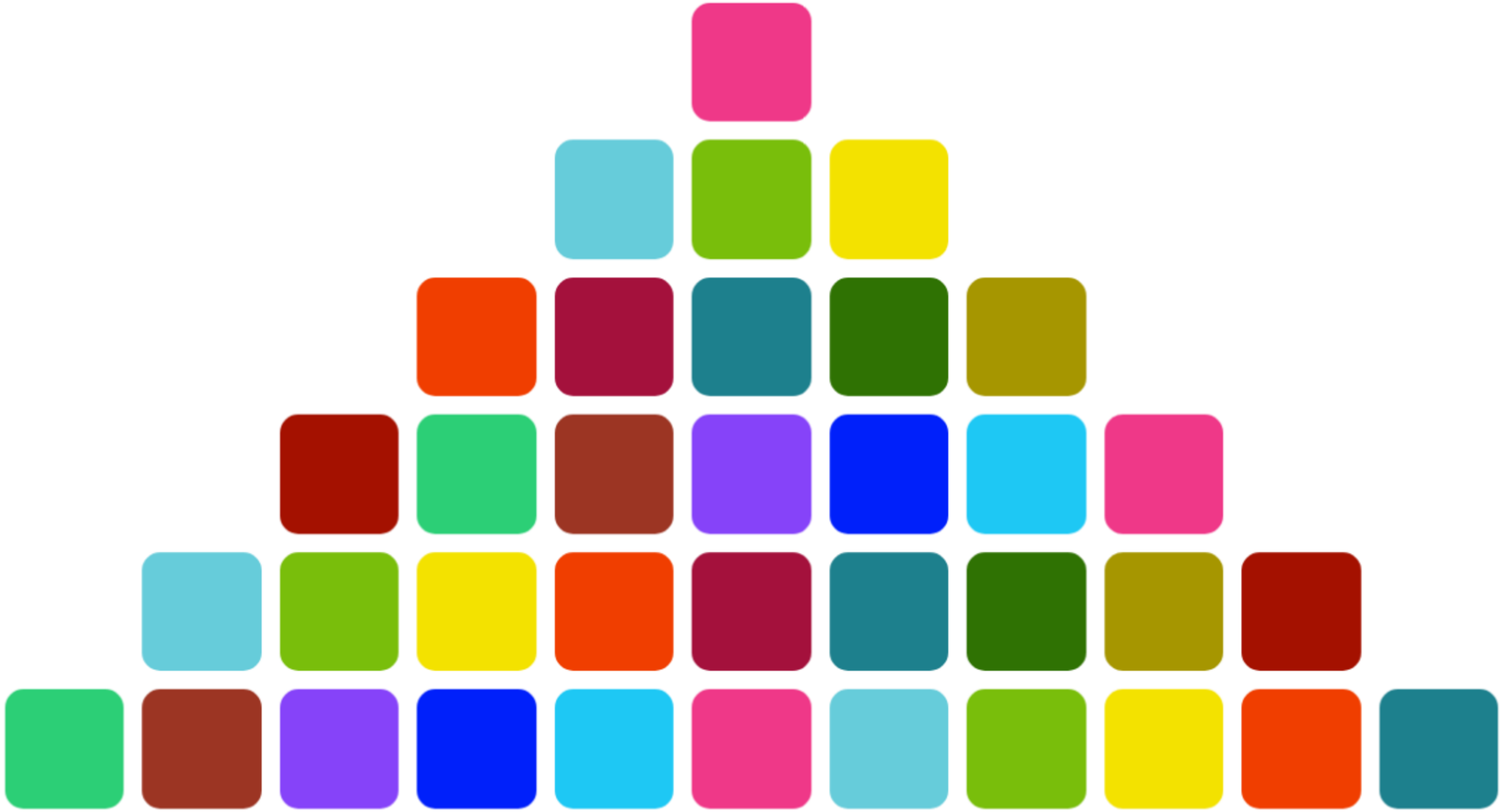
EXAMPLE

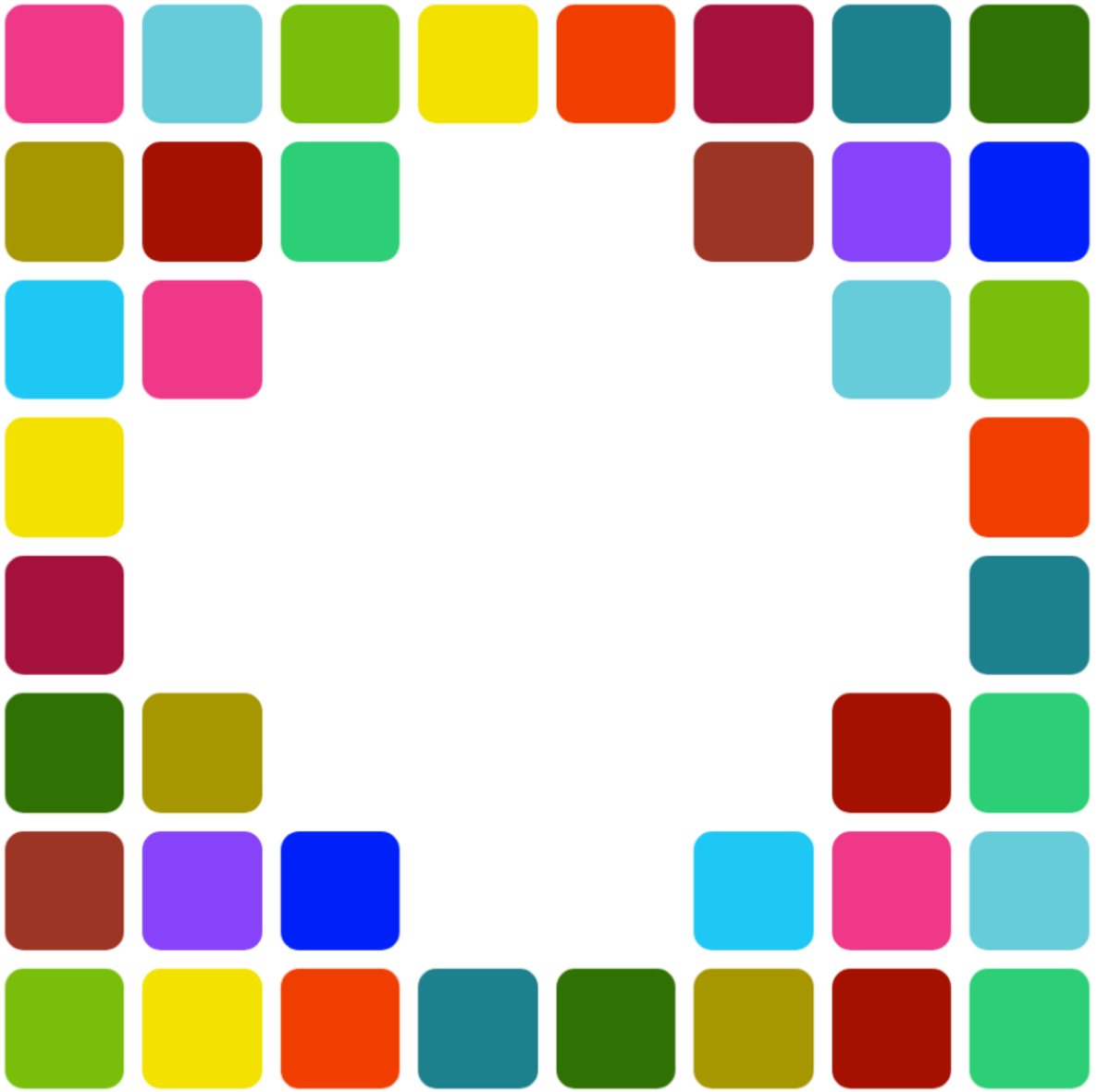
```
<div class="grid">
  <div></div>
  <div></div>
  <div></div>
  ...
  <div></div>
</div>
```



```
.grid { display: grid;
        grid: repeat(4, 100px) / repeat(4, 100px);
        grid-skip-areas: 1/1, 2/2, 3/3, 4/4; }
```

```
grid-skip-areas: 1/4/-1, 1/8/-1,
2/2/span 3/span 2, 2/6/auto/span 2,
4/5/auto/span 2, 2/10/auto/span 2,
4/9/auto/span 2;
```





TESTS

RUNNING "LAYOUT" TESTS

```
$ cd chromium/src/third_party/blink/
$ tools/run_web_tests.py -t Default/
Found 86793 tests; running 81273, skipping 5520.

$ tools/run_web_tests.py -t Default/ web_tests/external/wpt/cs
Found 916 tests; running 916, skipping 0.

All 916 tests ran as expected (908 passed, 8 didn't).
```

WEB PLATFORM TESTS (WPT)

GitHub repository:
https://github.com/web-platform-tests/wpt/

Shared tests repository for all browser vendors

Helps to improve interoperability between different implementations and also specs

src/third_party/blink/web_tests/external/wpt/

PARSING TEST

```
<!DOCTYPE html>
<meta charset="utf-8">
<title>CSS Grid Layout Test: parsing grid-skip-areas with
<link rel="author" title="Manuel Rego Casasnovas" href="ma
<link rel="help" href="https://drafts.csswg.org/css-grid-1
<meta name="assert" content="grid-skip-areas supports the
<script src="/resources/testharness.js"></script>
<script src="/resources/testharnessreport.js"></script>
<script src="/css/support/parsing-testcommon.js"></script>
<script>
// none
test_valid_value("grid-skip-areas", "none", "none");

// <grid-areas>
```

Summary

Harness status: OK

Found 6 tests

6 Pass

Details

Result	Test Name	Message
Pass	e.style['grid-skip-areas'] = "none" should set the property value	
Pass	e.style['grid-skip-areas'] = "1" should set the property value	
Pass	e.style['grid-skip-areas'] = "1 / 2" should set the property value	
Pass	e.style['grid-skip-areas'] = "1 / 2 / 3 / 4" should set the property value	
Pass	e.style['grid-skip-areas'] = "3 / 1 / span 2 / span 3" should set the property value	
Pass	e.style['grid-skip-areas'] = "1, 2" should set the property value	

JAVASCRIPT TEST

```
<!DOCTYPE html>
<meta charset="utf-8">
<title>CSS Grid Layout Test: grid-skip-areas</title>
<link rel="author" title="Manuel Rego Casasnovas" href="ma
<link rel="help" href="http://www.w3.org/TR/css-grid-1/#gr
<meta name="assert" content="This test checks that grid-sk
<link href="support/grid.css" rel="stylesheet">
<style>
.grid {
  position: relative;
  grid: repeat(3, 50px) / repeat(4, 100px);
}
.grid > div {
  background: magenta;
```

Summary

Harness status: OK

Found 2 tests

2 Pass

Details

Result	Test Name	Message
Pass	.grid 1	
Pass	.grid 2	

REFERENCE TEST

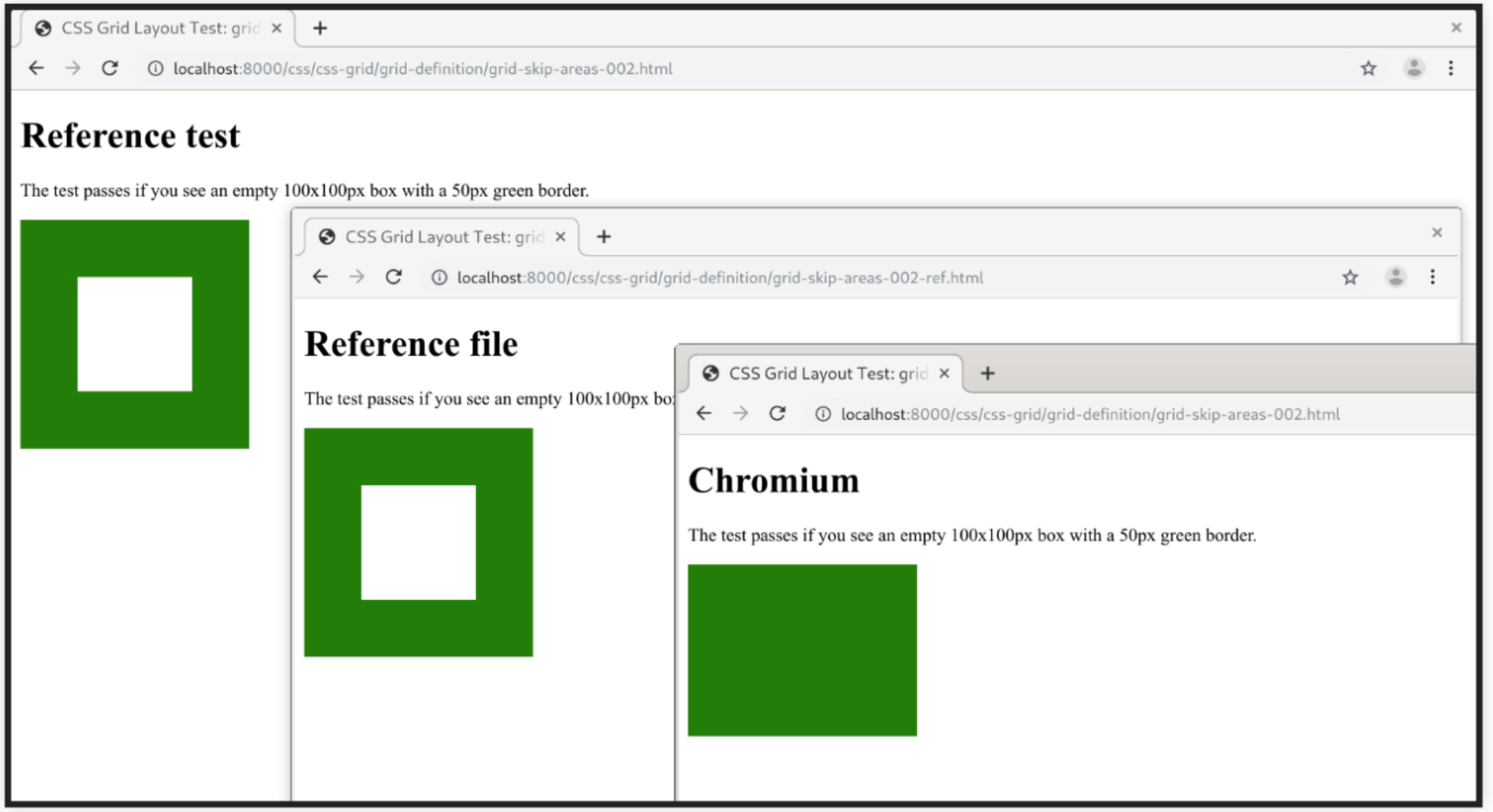
```
<!DOCTYPE html>
<meta charset="utf-8">
<title>CSS Grid Layout Test: grid-skip-areas</title>
<link rel="author" title="Manuel Rego Casasnovas" href="ma
<link rel="help" href="http://www.w3.org/TR/css-grid-1/#gr
<link rel="match" href="grid-skip-areas-002-ref.html">
<meta name="assert" content="This test checks that grid-sk
<style>
#grid {
  display: grid;
  grid: repeat(4, 50px) / repeat(4, 50px);
  grid-skip-areas: 2 / 2 / span 2 / span 2;
}
#grid > div {
```

REFERENCE FILE

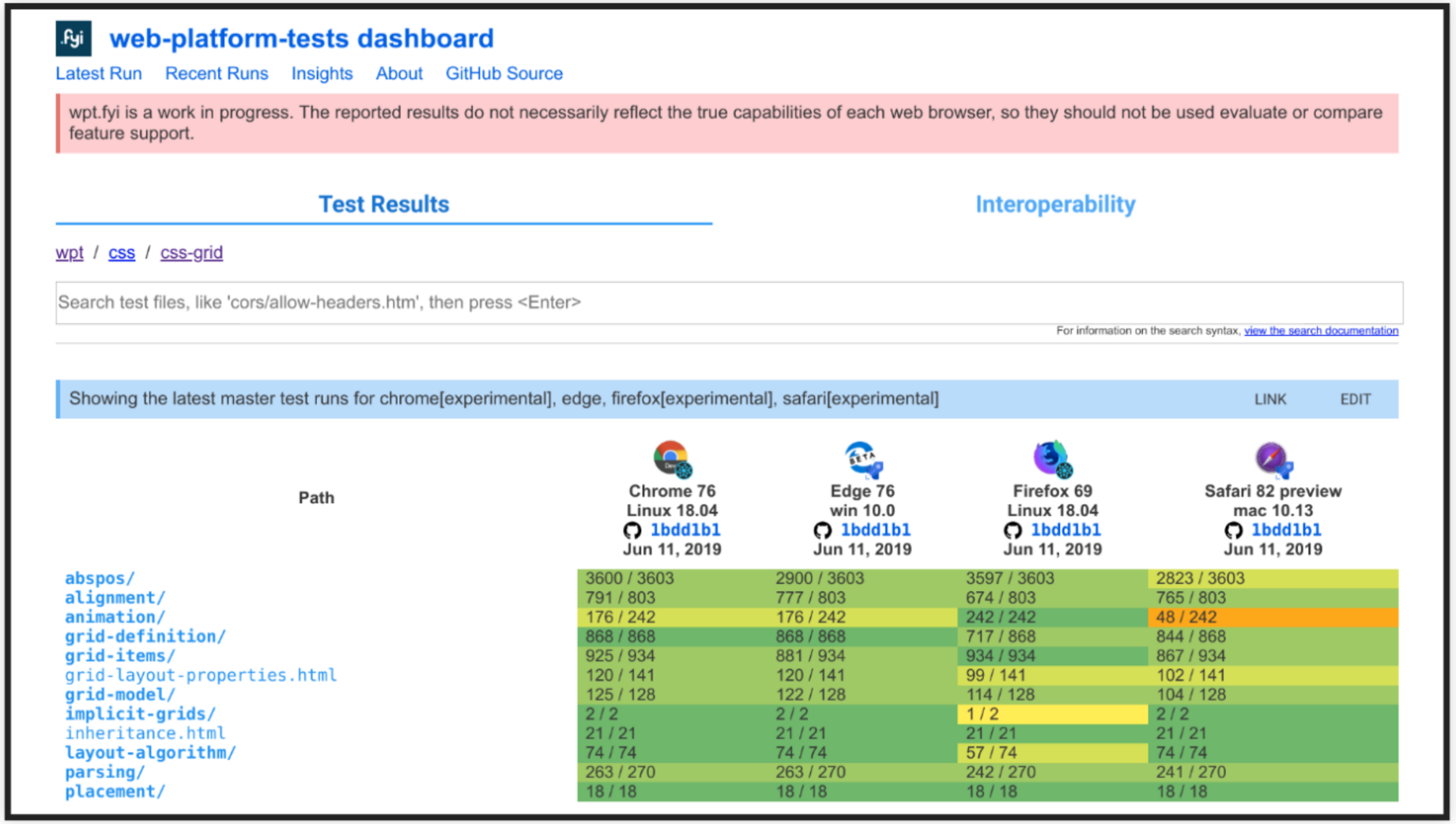
```
<!DOCTYPE html>
<html lang=en>
<meta charset="utf-8">
<title>CSS Grid Layout Test: grid-skip-areas reference fil
<link rel="author" title="Manuel Rego Casasnovas" href="ma
<style>
div {
  width: 100px;
  height: 100px;
  border: 50px solid green;
}
</style>

<p>The test passes if you see an empty 100x100px box with
```

REFERENCE TEST



WPT.FYI



web-platform-tests dashboard

Latest Run Recent Runs Insights About GitHub Source

wpt.fyi is a work in progress. The reported results do not necessarily reflect the true capabilities of each web browser, so they should not be used evaluate or compare feature support.

Test Results Interoperability

wpt / css / css-grid

Search test files, like 'cors/allow-headers.htm', then press <Enter>

For information on the search syntax, view the search documentation

Showing the latest master test runs for chrome[experimental], edge, firefox[experimental], safari[experimental] LINK EDIT

Path	Chrome 76 Linux 18.04 1bdd1b1 Jun 11, 2019	Edge 76 win 10.0 1bdd1b1 Jun 11, 2019	Firefox 69 Linux 18.04 1bdd1b1 Jun 11, 2019	Safari 82 preview mac 10.13 1bdd1b1 Jun 11, 2019
abspos/	3600 / 3603	2900 / 3603	3597 / 3603	2823 / 3603
alignment/	791 / 803	777 / 803	674 / 803	765 / 803
animation/	176 / 242	176 / 242	242 / 242	48 / 242
grid-definition/	868 / 868	868 / 868	717 / 868	844 / 868
grid-items/	925 / 934	881 / 934	934 / 934	867 / 934
grid-layout-properties.html	120 / 141	120 / 141	99 / 141	102 / 141
grid-model/	125 / 128	122 / 128	114 / 128	104 / 128
implicit-grids/	2 / 2	2 / 2	1 / 2	2 / 2
inheritance.html	21 / 21	21 / 21	21 / 21	21 / 21
layout-algorithm/	74 / 74	74 / 74	57 / 74	74 / 74
parsing/	263 / 270	263 / 270	242 / 270	241 / 270
placement/	18 / 18	18 / 18	18 / 18	18 / 18

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

SHIP IT!

Intent to ship mail

Change runtime flag to `stable`

`../platform/runtime_enabled_features.json5`

```
1       {
2         name: "CSSDayGridSkipAreas",
3         status: "stable",
4       },
```

After a few cycles the runtime flag will be removed

REVIEW PROCESS

All changes need to be approved by an owner/reviewer

Tests should pass in all platforms

E.g. https://chromium-review.googlesource.com/c/chromium/src/+/1659448/1

CONTRIBUTING TO WEB PLATFORM

Open process, all browser engines now are open source

Shipping some feature has an **huge impact** (interoperability and backwards compatibility)

OPEN WEB PLATFORM

Externals individuals and/or companies (other than browser vendors) **can contribute**

- Report/Star/Vote bugs
- Use new features and provide feedback
- Write/Tweet/Speak about the new features and missing use cases
- Participate in CSS WG discussions
- Implement the feature yourself or sponsoring someone else

ACKNOWLEDGEMENTS

igalia Bloomberg

Igalia and Bloomberg working together
to build a better web

THANKS

- Twitter: @regocas
- Mail: rego@igalia.com
- Blog: http://blogs.igalia.com/mrego/