



# Working with grids

Lesson 8

# Lesson plan

1

What is Grid, and how does it differ from Flexbox?

2

Working with Columns and Rows

3

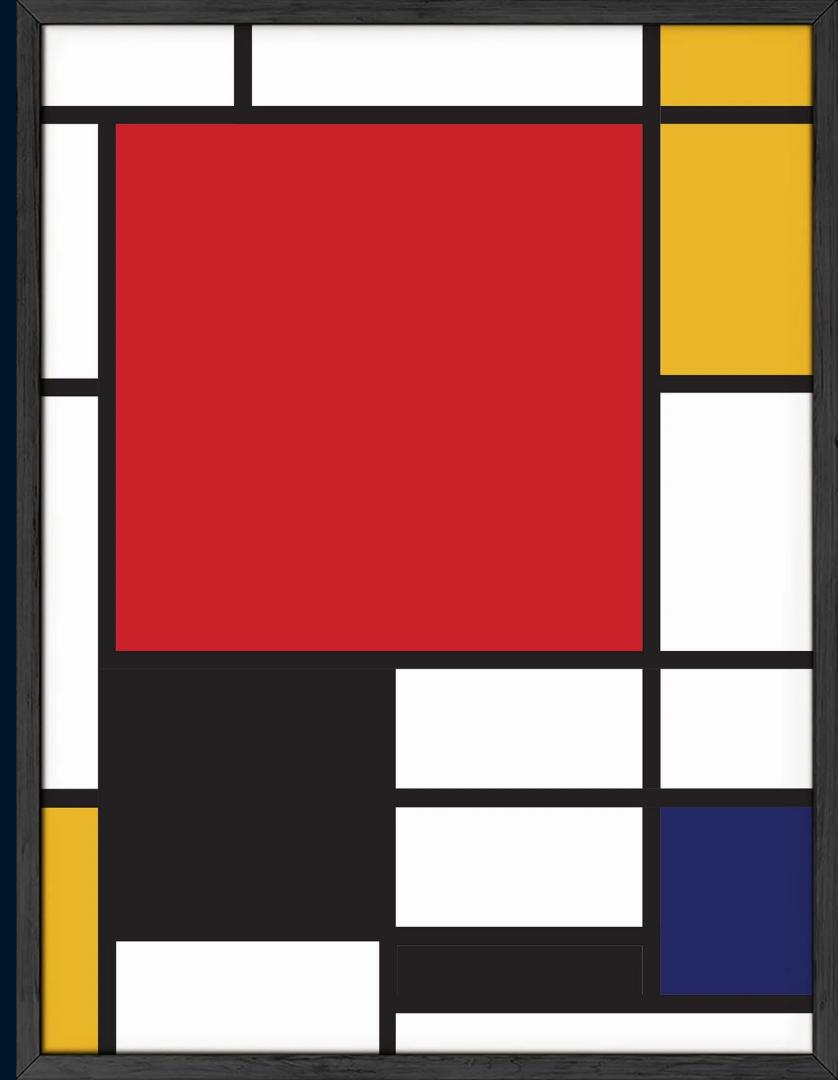
Named Areas

4

Automatic Filling and Rearrangement of Elements

# Grid

- is a two-dimensional layout system for organizing elements in a grid, ensuring clarity about the structure and size of content, and allowing us to define where everything is located.



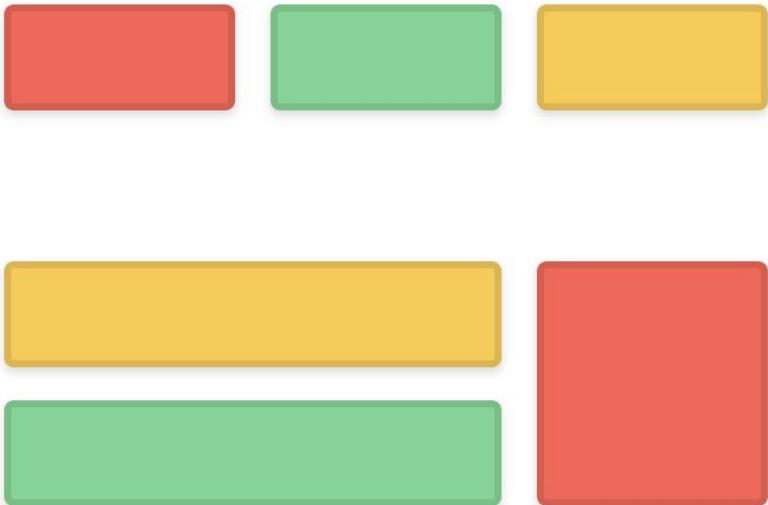
*"Composition"* Piet Mondrian

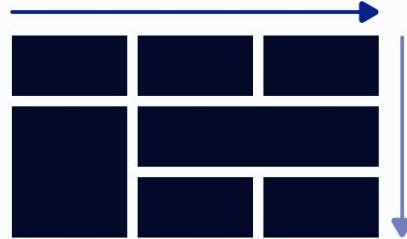
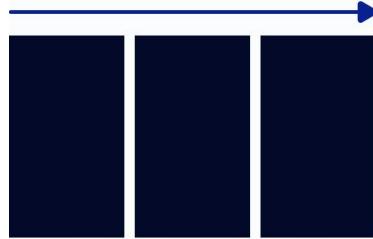
# FlexBox vs Grid

Does CSS Grid Replace Flexbox?

**No. Well. Mostly No.**

- Grid can do things Flexbox can't do.
- Flexbox can do things Grid can't do.
- They can work together: a grid item can be a flexbox container. A flex item can be a grid container.





<b>FlexBox</b>	<b>Grid</b>
<b>One-dimensional</b> layouts	<b>Two-dimensional</b> layouts
Flexbox will wrap down onto another row when the flex items fill a row	Grid will fall along the same grid lines all the other elements do
In flexbox, most of the layout (beyond the very basics) happen <b>on the children</b>	Grid is mostly defined <b>on the parent element</b>
Flexbox <b>can push things away</b>	Grid is <b>better at overlapping</b>

# FlexBox

## One-dimensional layout

Either vertically or horizontally

- Lists
- Navigation
- Aligning elements along one axis
- Flexible responsiveness
- Micro layouts

# Grid

## Two-dimensional layout

Simultaneous control over horizontal and vertical directions

- General page structure
- Complex layouts
- Grid with strict size control of elements
- Repeating and template layouts

# FlexBox

START NOW  
**Your plan, your choice.**

**MOST POPULAR**

**Starter**  
\$49

No setup fee  
Unlimited products  
Mobile inventory  
Mobile app

**Personal**  
\$99

No setup fee  
Unlimited products  
Mobile inventory  
Mobile app

**Team**  
\$229

No setup fee  
Unlimited products  
Mobile inventory  
Mobile app

**Get Started**

# Menu

Fusion menu collaboration between Master Chef Robert Smith and Jennifer Morisette

**First course**  
Fresh steamed lobster with asparagus and lemon juice

**Main course**  
Roast cauliflower with hazelnut, mint, parsley and herb cream cheese

**Dessert**  
Lemon pudding and raspberry panna cotta with shaved dark chocolate

**Book a Table!**

# Grid

Eat 5 fruits and...  
**The Benefits**  
Clean, wholesome eating is the first step to a new way of living and being.

The Values  
You'll start to enjoy the benefits within just a few days.

**GOOD MENS GOODS**  
New Arrivals | Designers | Clothing | Footwear | Accessories | The Journal

**DETACHED HOUSE • 5Y OLD**  
\$750,000  
742 Evergreen Terrace  
3 Bedrooms | 2 Bathrooms  
REALTOR  
Tiffany Heffner  
(555) 555-4321

**FALL MIXTAPE**  
Saturdays Surf NYC's Co-Founders Talk About the Brand

**CRAZY WEEKEND SALE**

**PHOTOGRAPHY**

**the JOURNAL**

**STAFF & PROJECT WITHOUT WORRYING ABOUT THE END PRODUCT**

**Logo**

**Hero Heading**  
Lorem ipsum dolor sit amet consectetur adipisicing elit. Tempora, accusamus.

**Read More**

FREE SHIPPING ON ALL ORDERS OVER \$50

**LOVE**

**Pure Coconut Oil**  
\$28.00

8 fl.oz.

100% pure organic & cruelty free. 100% MCT. This is a type of healthy polyunsaturated fat. It's loaded specifically for your hair and skin. Say goodbye to dryness that comes up from coconut cooking Oils & Butters.

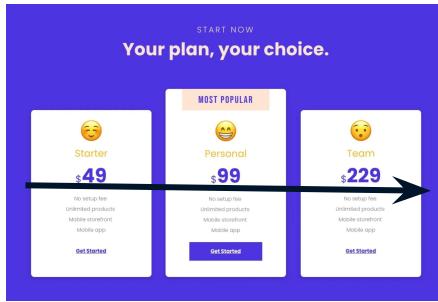
Using our natural cold-pressed and certificate extraction method, this product is packed with healthy fatty acids, non-greasy, and absorb quickly into your hair and skin leaving it nourished, healthy, and robbery smooth.

**Quantity**  
  
**ADD TO CART**

**Ingredients**  
**Watch Video**  
**Benefits & How To Use**  
**Why Love Coconut Oil**

**SHARE** **TWEET** **PIN**

# FlexBox



Menu

Fusion menu collaboration between Master Chef Robert Smith and Jennifer Morissette

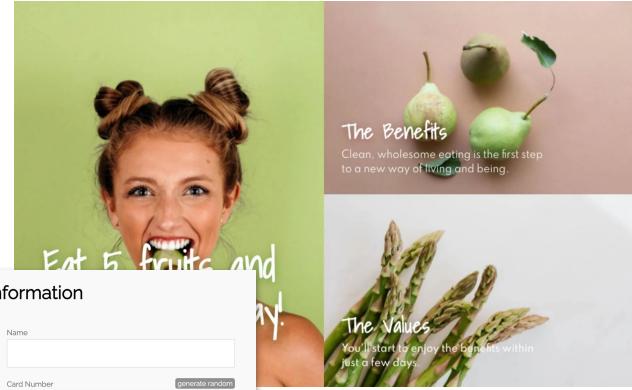
**First course**  
Fresh steamed lobster with asparagus and lemon juice

**Main course**  
Roast cauliflower with hazelnut, mint, parsley and herb cream cheese

**Dessert**  
Lemon pudding with raspberry panna cotta with shaved dark chocolate

Book a Table!

# Grid



## Payment Information



## cells

GOOD MENS GOODS

PHOTOGRAPHY

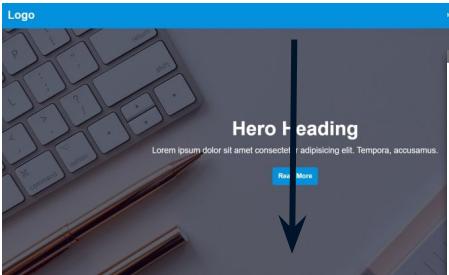
CRAZY WEEKEND SALE

the JOURNAL

NEW ARRIVALS

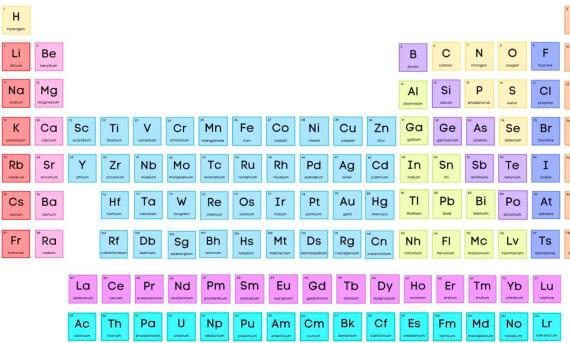
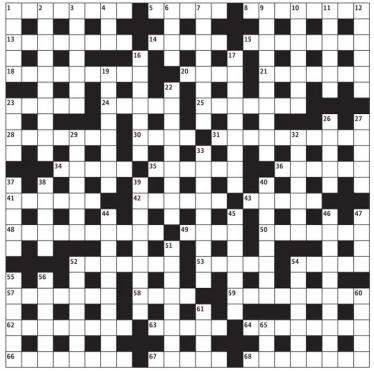
Over 200 PRODUCTS ADDED

SEARCH



# axis

# Grid in real life



**Steak Burgers**

(All sandwiches served with choice of French Fries, side dish or potato salad)

P.O.B. (plus 1/2 pound cheeseburger)	4.49	Mushrooms & Onions	4.49
Bacon Cheeseburger	4.49		
Santa Fe Burger (plus cheese, onions, & jalapeños)	4.49		

**Salads**

Dinner Salad	1.99	Side Salad	1.99
Fiesta Salad	1.99	Grilled Chicken Salad	4.99
Basket 1.99		Blackened Chicken Salad	4.99

**Soups & Chili**

Soup of the Day (low fat 2.99)	
Chili or Jalapeño Chili (low fat 2.99)	

**Beverages**

Soft Drinks (Milk, Iced Tea)	2.99	Hot Chocolate	2.99
	2.99	Hot Tea	2.99

**Hand Dipped Malts & Shakes**

2.99
------

**The Hound Dog**

(Served in a long 1/2 pound bun)

Pickle Plow	2.79
Chili Cheese Dog	3.99
Bacon	3.49

**Breakfast**

(Rocky Road, low carb bacon, hash browns, & toast)

Rocky Road	2.79
Hash Browns	3.49



**The Good Approach:**  
Assume that a colleague speaks up to you, even if it's just once. This "good approach" approach has been described as "the most effective method of creating positive energy and social capital among employees." It's also a strategy that's more likely to bring people together, rather than divide them.

For example, let's say that in your group, you have a problematic colleague who tends to talk a lot, but it's not about the work. Instead, he's just really good at it. And he's a bit of a team player, too. You might think that you should ignore him, as he's not contributing much to the group. But that's not the case. In fact, it's important to address his behavior and let him know that it's not acceptable. This will help him understand that his behavior is not welcome and encourage him to change his behavior for the better.

It's also important to remember that the employee is not the problem. It's their behavior that's causing issues.

**Using Caution:**  
It's also important to remember that the people you're working with may have different backgrounds and experiences. Some may be more experienced than others, while others may be less experienced. It's important to be aware of this when addressing behavior.

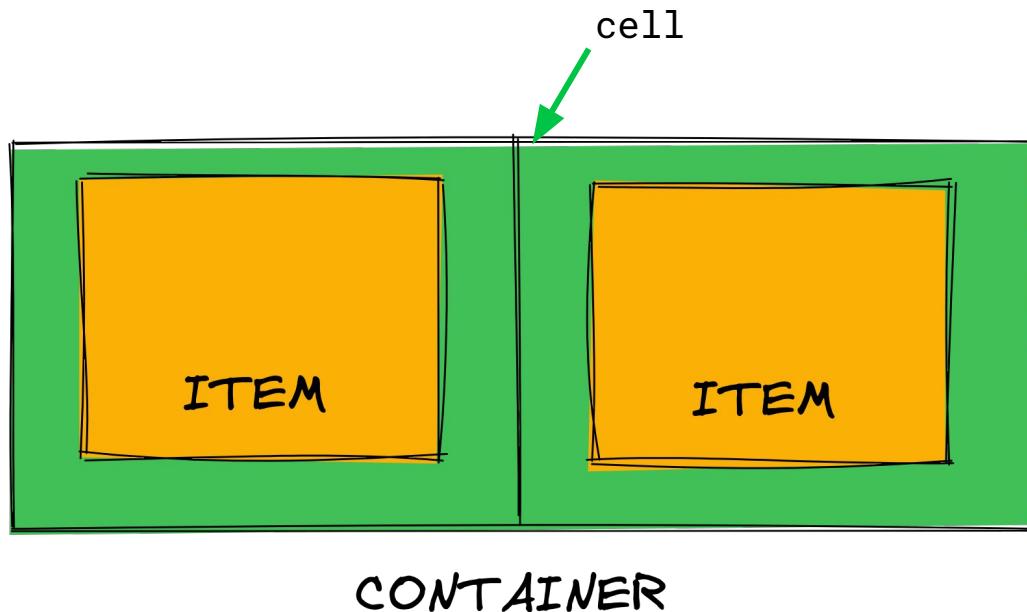
**Creating Affinity:**  
To create a positive work culture, it's important to focus on creating affinity and trust. This means creating environments where people feel comfortable and supported.

**Dealing with Broken Promises:**  
What we're talking about here is making promises that we don't keep. It's important to be transparent and honest when making promises.



HOGWARTS EXPRESS				
PASSENGER		DEPART FROM PLATFORM		
HERMIONE J. GRANGER		HOGMEEDE		
DEPART		ARRIVE	TIME	
KING'S CROSS		HOGMEEDE	DEPART	ARRIVE
DATE	TIME	CAR	SEAT	
1 SEP 2018	11:00AM	4	BB	9 3/4
BOARDING BEGINS 30 MINUTES BEFORE DEPARTURE. SNACKS AVAILABLE FOR PURCHASE FROM THE HONEYDUKE'S EXPRESS.				
THIS TICKET IS NON-REFUNDABLE - HOGWARTS EXPRESS RAILWAY AUTHORITY				
 001256733 				

# Grid container and grid item



```
.container {  
    display: grid;  
    display: inline-grid;  
}
```

# Grid lines

Left to right / top to bottom – **positive** numbers

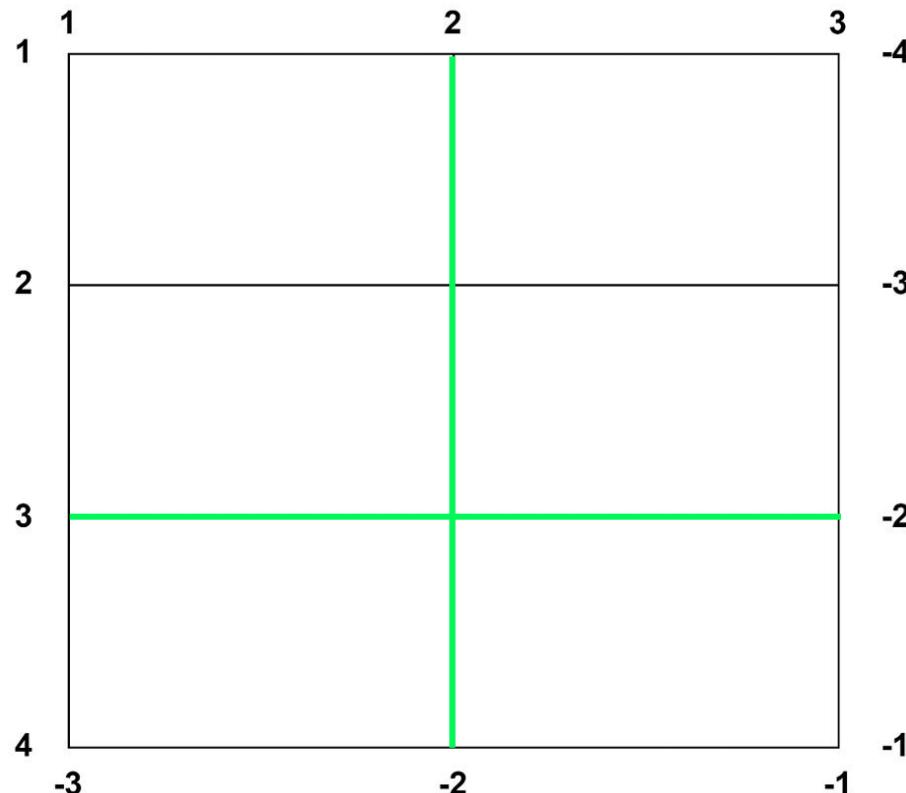
Right to left / bottom to top – **negative** numbers

There are always **one more** line than the **number of columns or rows**

You can give custom names to lines using:

**grid-template-rows**

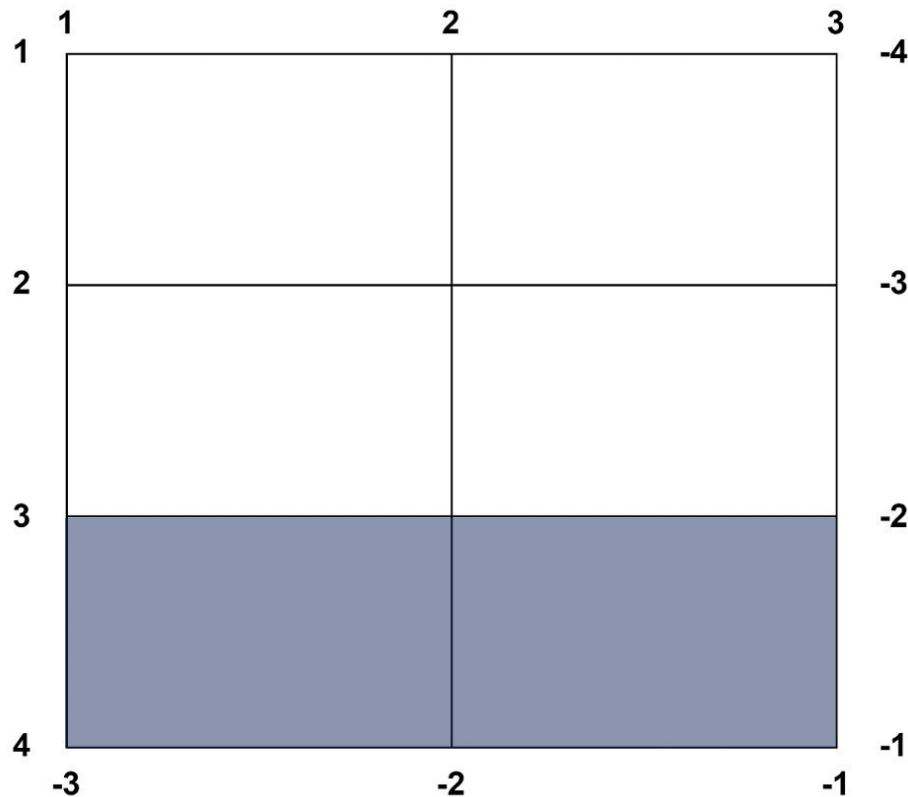
**grid-template-columns**



# Grid track

The area **between two lines** in one direction

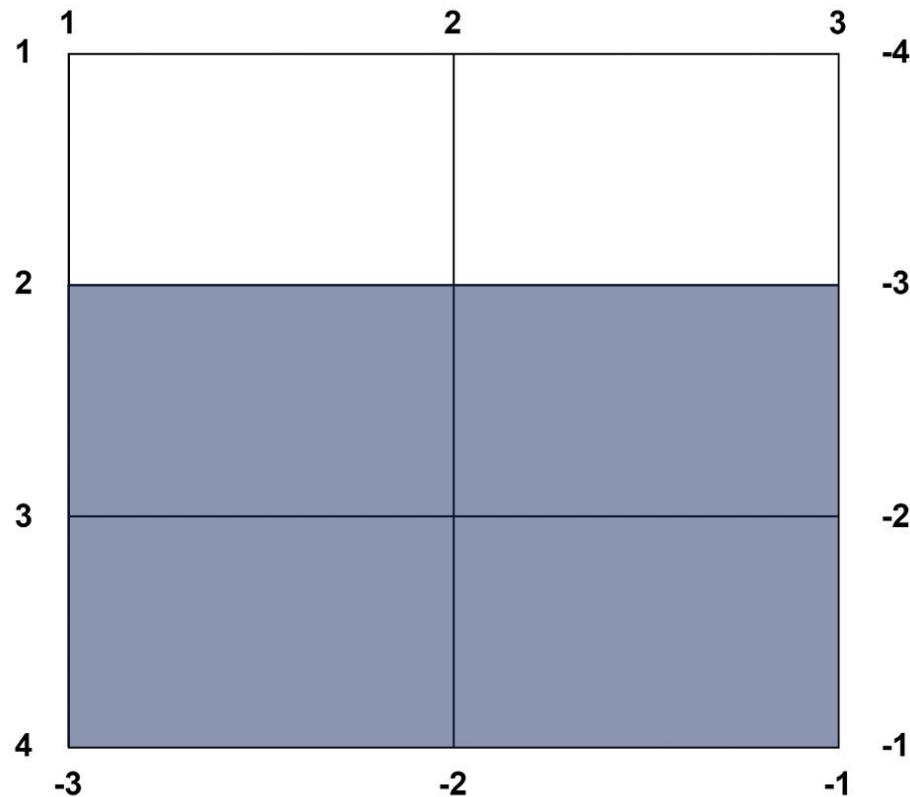
Either horizontally or vertically  
columns / rows



# Grid area

An area surrounded by  
**four lines**

Can contain any number  
of cells



# Grid fr

- fraction

Allows distributing free space between elements according to proportion

Works only with **grid**

$$1\text{fr} + 2\text{fr} = 3\text{fr}$$

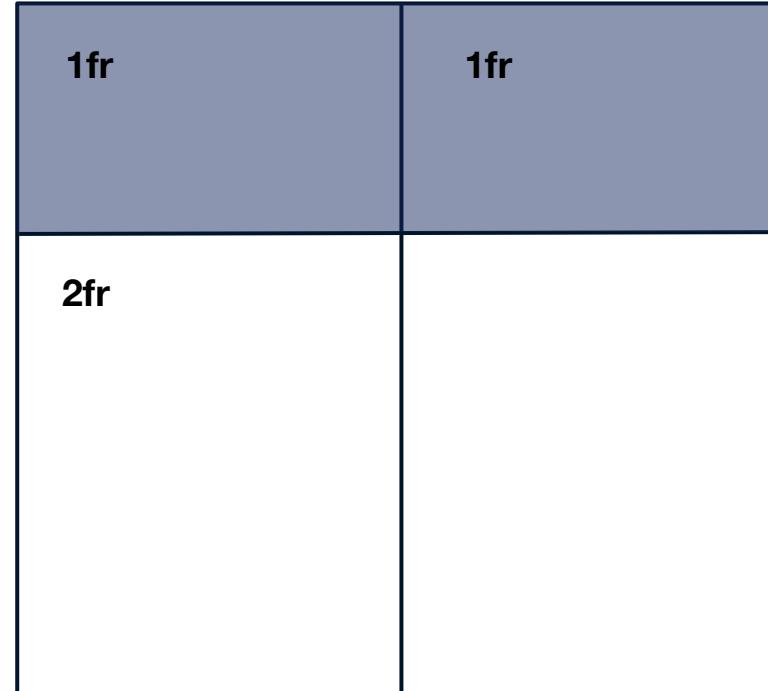
$\frac{1}{3}$

**1fr**

**1fr**

$\frac{2}{3}$

**2fr**



# Grid gap

- Determines the distance between grid cells

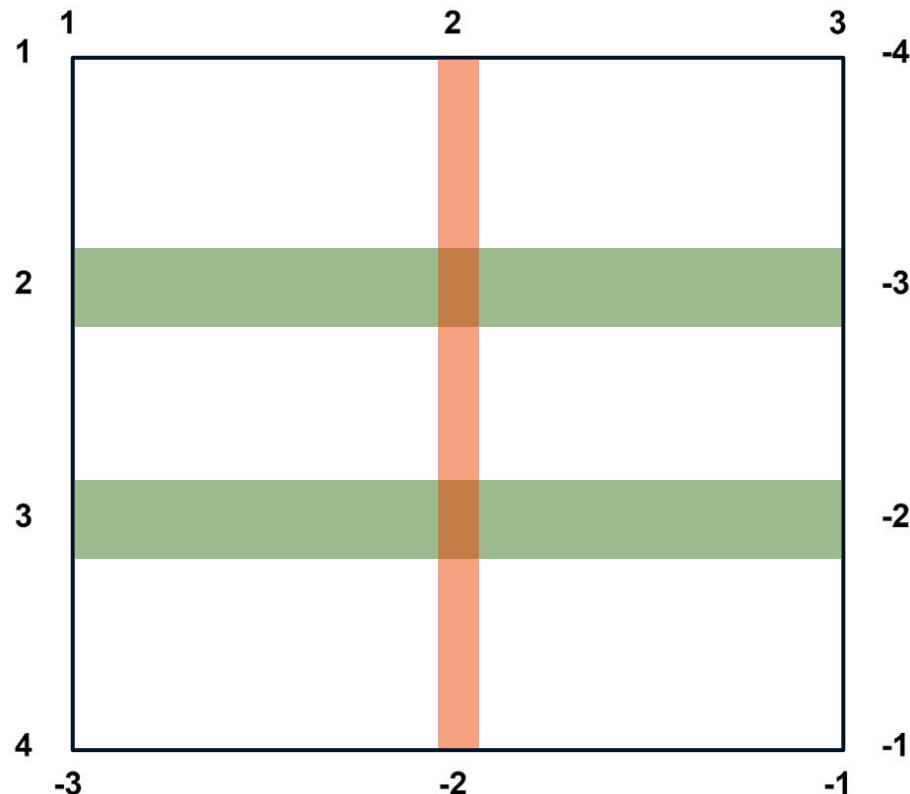
Applied to a **grid container**

**gap: 10px 20px;**

**gap-row: 10px;**

**gap-column: 20px;**

**px, rem, em, %**

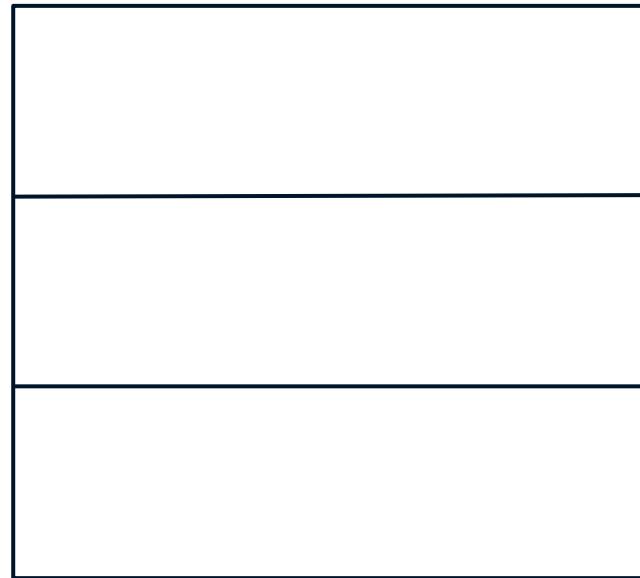


# Creating a Grid

# Grid layout

```
<section class="grid-container">
  <div class="grid-cell"></div>
  <div class="grid-cell"></div>
  <div class="grid-cell"></div>
</section>

.grid-container {
  display: grid;
}
```



# Grid cell

for grid cell don't work:

- **display: inline-block**
- **display: table-cell**
- **float**
- **vertical-align**

However, you can use **position** for cells.



# Grid columns/rows (base properties)

Managing row sizes:

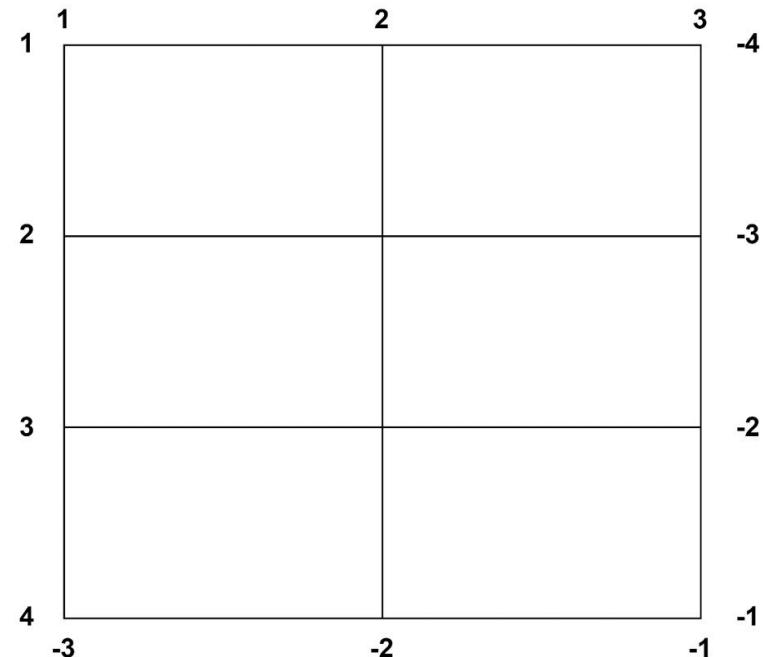
```
grid-template-rows: 1fr 1fr 1fr;  
grid-template-rows: 200px auto 1fr;
```

Managing column sizes:

```
grid-template-columns: 1fr 1fr;  
grid-template-columns: 50% 20px auto;
```

Values in units:

- px
- %
- fr
- auto



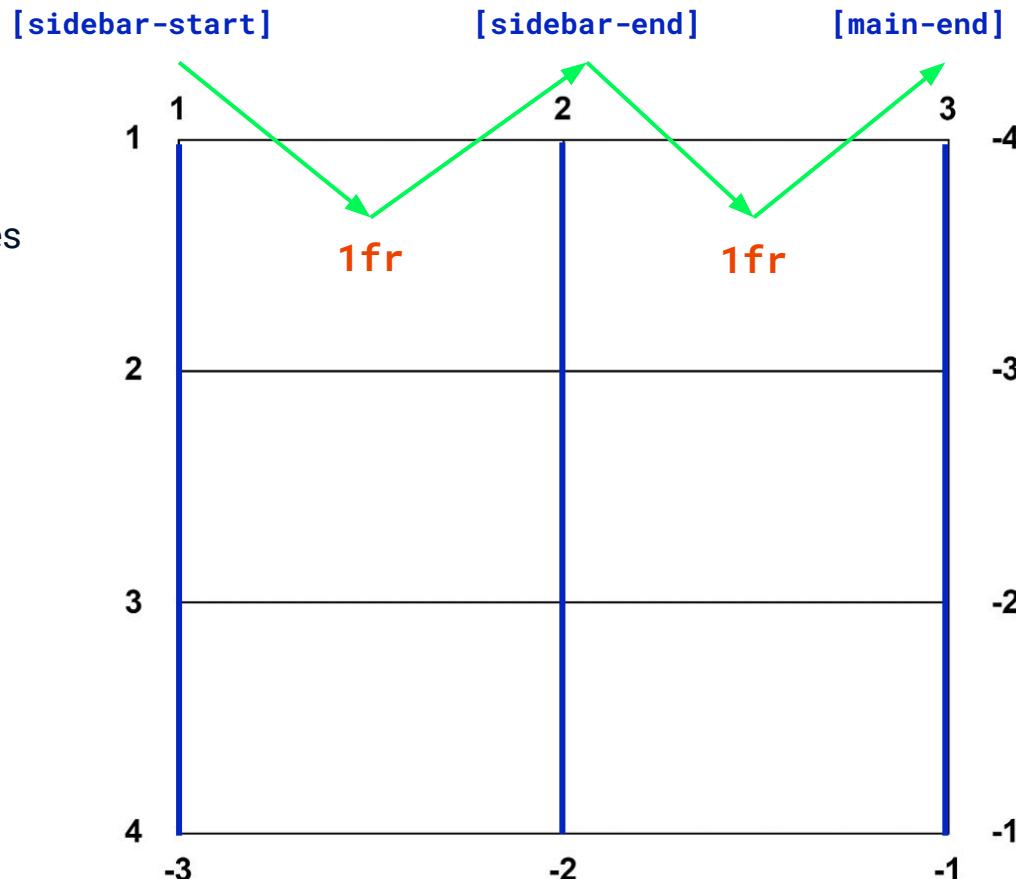
# Grid lines

You can give custom names to lines using properties:

`grid-template-rows`  
`grid-template-columns`

Naming goes from left to right:

`grid-template-columns:`  
  `[sidebar-start]`  
  `1fr`  
  `[sidebar-end]`  
  `1fr`  
  `[main-end];`



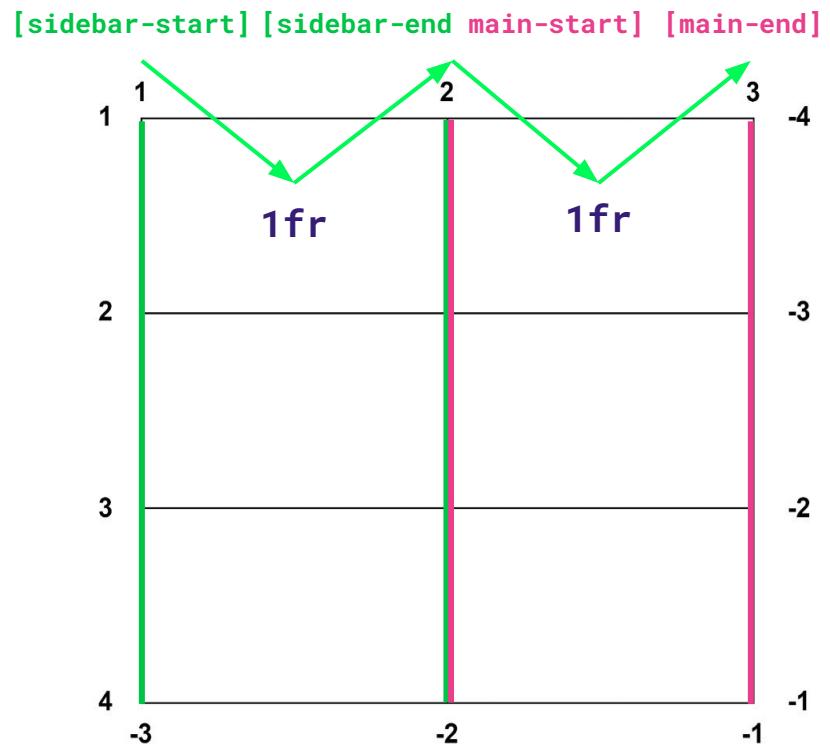
# Grid lines with Two Names

A line can have two names:

```
grid-template-columns:  
  [sidebar-start]  
  1fr  
  [sidebar-end main-start]  
  1fr  
  [main-end];
```

```
header {  
  grid-column-start: sidebar-start;  
  grid-column-end: main-end;  
}
```

Code becomes more readable



# Building a Grid

Practice

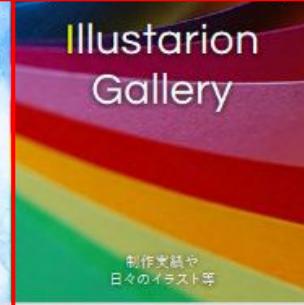


The collage includes:

- A large central image of a pearl necklace with a diamond and gemstone pendant.
- To the left, a green-skinned alien-like character wearing a purple zip-up hoodie.
- To the right, two orange, bird-like characters with wings and small yellow spheres on their heads.
- At the bottom left, a girl with long blonde hair in a black dress.
- At the bottom right, a colorful graphic with the text "Illustrarion Gallery" and "制作実績や日々のイラスト等" (Past works and daily illustrations).
- On the far left, a blue background featuring the PATAPATA WORKS logo and a paragraph of Japanese text describing the studio's services.



PATAPATA WORKS(パタパタワークス)は、福岡県福岡市でWEBデザインを始めとしたデザイン業務、イラスト制作・キャラクターデザインなどを行っているフリーランスデザイナーです。  
ホームページ制作やイラスト制作などでお困りでしたらお気軽にお問い合わせください。



```
<main class="grid-container">
    <article class="grid-cell"></article> x6
</main>

.grid-container {
    display: grid;
    grid-template-columns: 1fr 1fr 1fr 1fr;
    grid-template-rows: 1fr 1fr 1fr;
}
```

You can insert images of your favorite characters

[Demo](#)

# Grid functions

# Grid repeat()

- number of repetitions for rows or columns

**grid-template-rows:**

**repeat(3, 1fr);**

=

**grid-template-rows: 1fr 1fr 1fr;**

**repeat(3, 50px 100px)**

6 rows: 3 odd **50px**, 3 even - **100px**

~~**repeat(3 100px repeat(3, 1fr)  
100px)**~~

<b>1fr</b>	
<b>1fr</b>	
<b>1fr</b>	

# Creating a Grid – Practice

```
<main class="grid-container">
  <article class="grid-cell"></article> x6
</main>

.grid-container {
  display: grid;
  grid-template-columns: repeat(4, 1fr);
  grid-template-rows: repeat(3, 1fr);
}
```

[Demo](#)

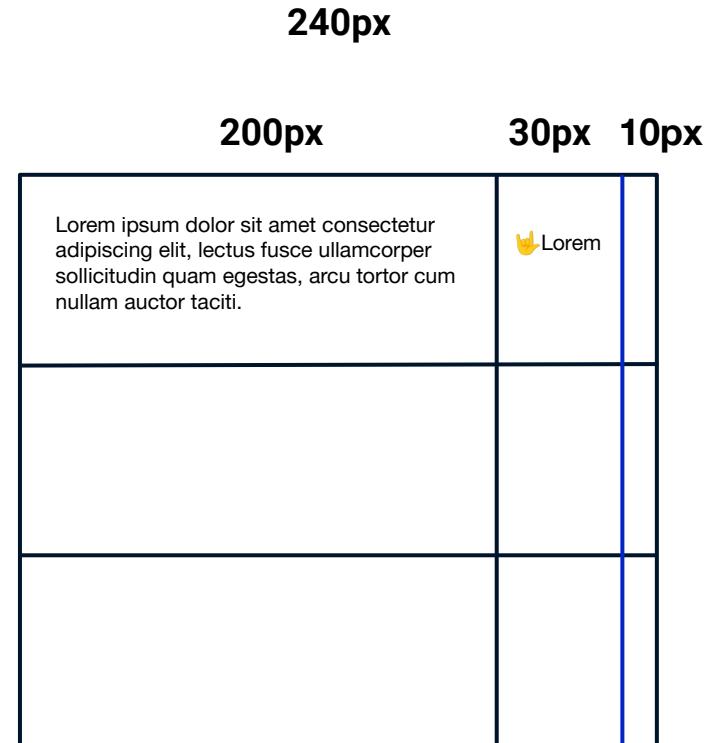
# Grid `fit-content()`

- a cell takes up as much space as its content, but is limited by a maximum size

`grid-template-columns: fit-content(200px);`

*The first column will stretch until its content reaches **200px**, then it will stop stretching.*

*The second column takes up as much space as its content requires, but if the content increases, it can stretch up to **200px** in width.*



# Grid max-content

240px

200px

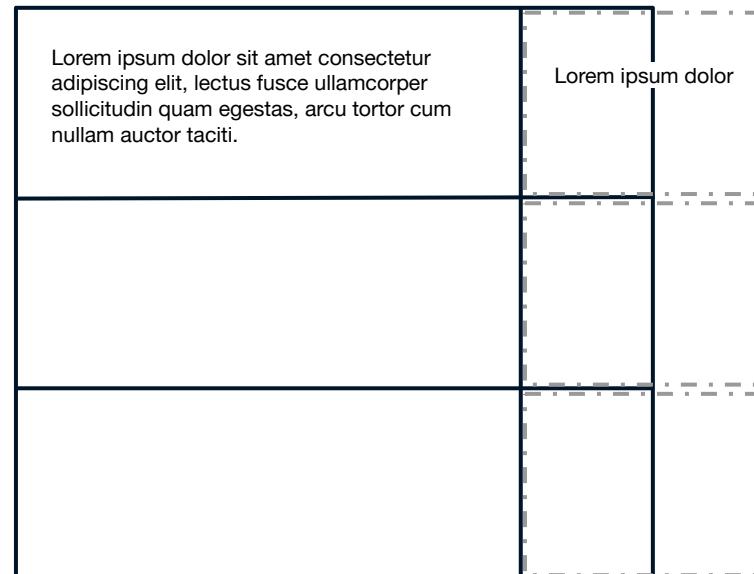
40px

- forces an element to take up as much space as it needs to display all content.

If there is no content, the cell will collapse.

If there is a lot of content, it might overflow the container.

**grid-template-columns:**  
**fit-content(200px) max-content;**



# Grid min-content

- the content will take all soft-wrapping opportunities, becoming as small as the longest word.

```
grid-template-columns:  
min-content min-content 1fr;
```

240px

100px

40px

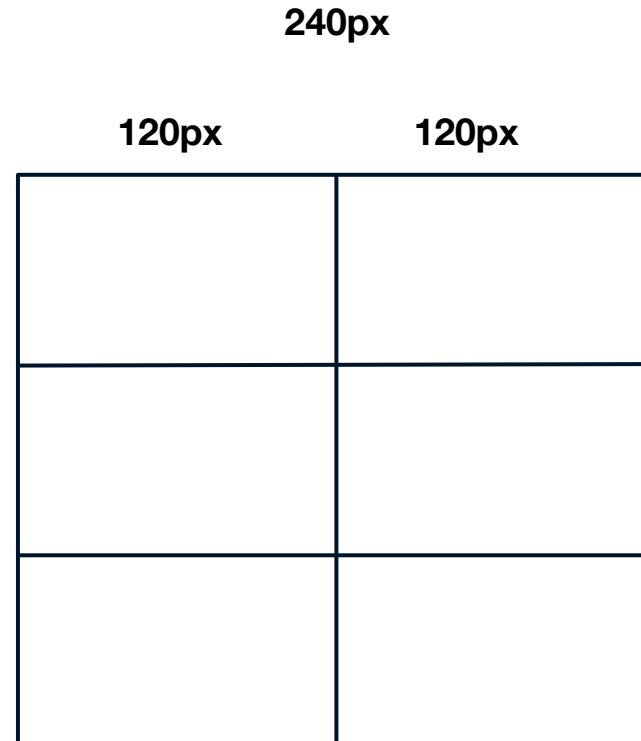
Lorem ipsum dolor sit amet consectetur adipiscing elit, lectus fusce ullamcorper sollicitudin quam egestas, arcu tortor cum nullam auctor taciti.	Lorem ipsum dolor	

# Grid auto

Auto sets the size for rows or columns depending on their content.

If there is a lot of content, it will take up all the available space.

```
grid-template-columns: auto auto;  
=   
grid-template-columns: 1fr 1fr;
```



# Grid auto

The cell stretches to the full available width/height of the container.

**grid-template-columns: auto 1fr  
30px;**

240px	120px	1fr	30px
	Lorem ipsum dolor sit amet consectetur adipiscing elit, lectus fusce ullamcorper sollicitudin quam egestas, arcu tortor cum nullam auctor taciti.	Lorem	

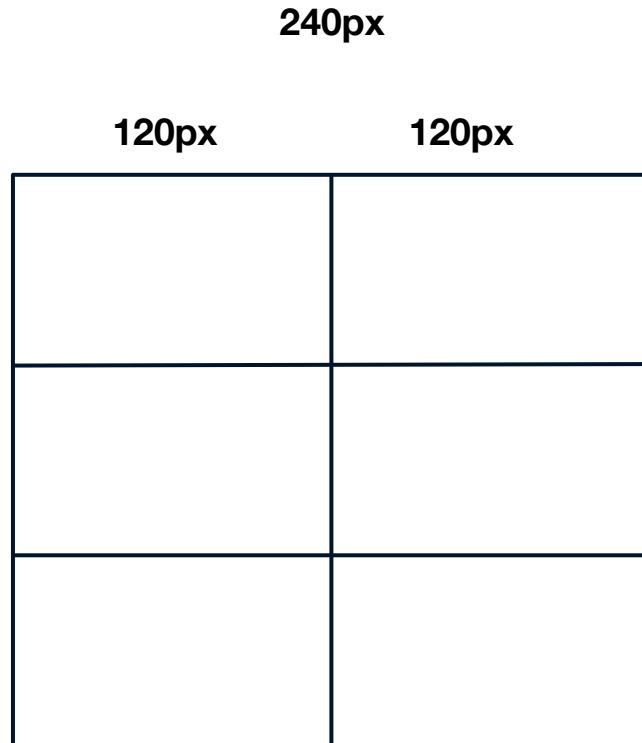
# Grid minmax()

- sets the minimum and maximum size for rows or columns.

Allows creating fluid and responsive layouts.

**grid-template-columns:**  
**minmax(100px, 1fr);**

*The column width should be at least 100px, but if possible, additional space can be allocated proportionally from the available space.*



# Positioning Elements in the Grid

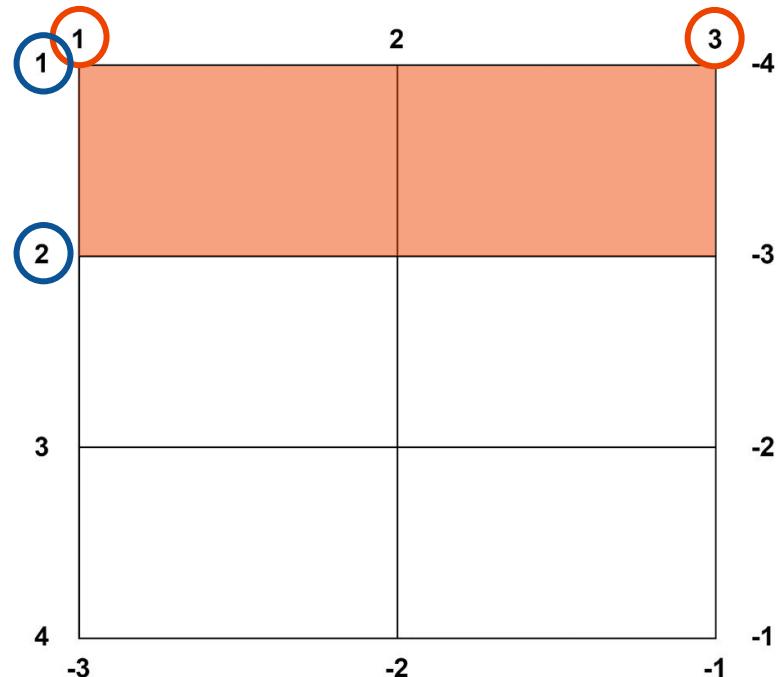
# Grid positioning

Managing the size of a cell by specifying the lines between which the element spans

Applied to a `grid cell`

```
grid-column-start: 1;  
grid-column-end: 3;
```

```
grid-row-start: 1;  
grid-row-end: 2;
```

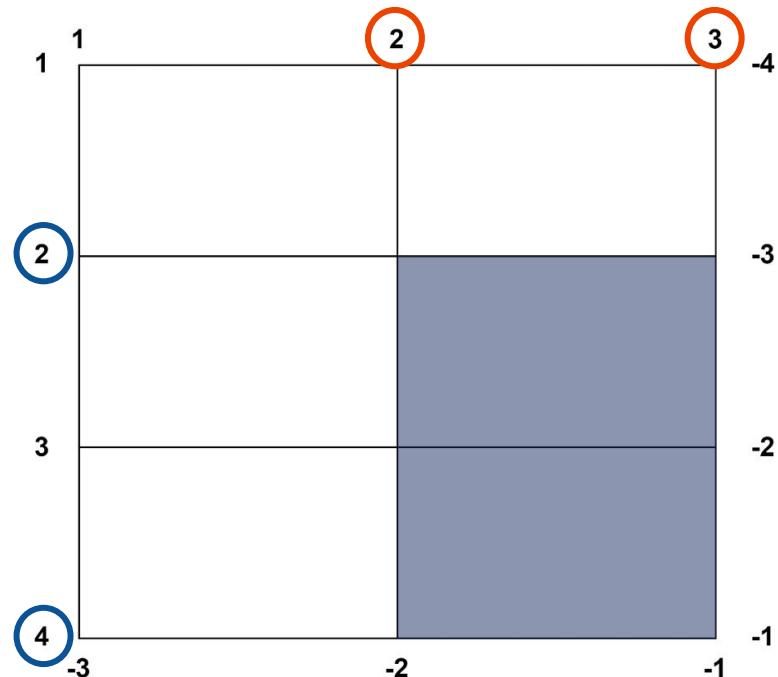


# Grid positioning

Shorthand notation:

`grid-column: 2 / 3;`

`grid-row: 2 / 4;`



# Value types:

- **Line number**

```
grid-column: 1 / 3;
```

- **span - the number of occupied cells**

```
grid-column: 1 / span 2;
```

- **line-name**

```
grid-column: start / end;  
grid-column: [start] / [middle];
```

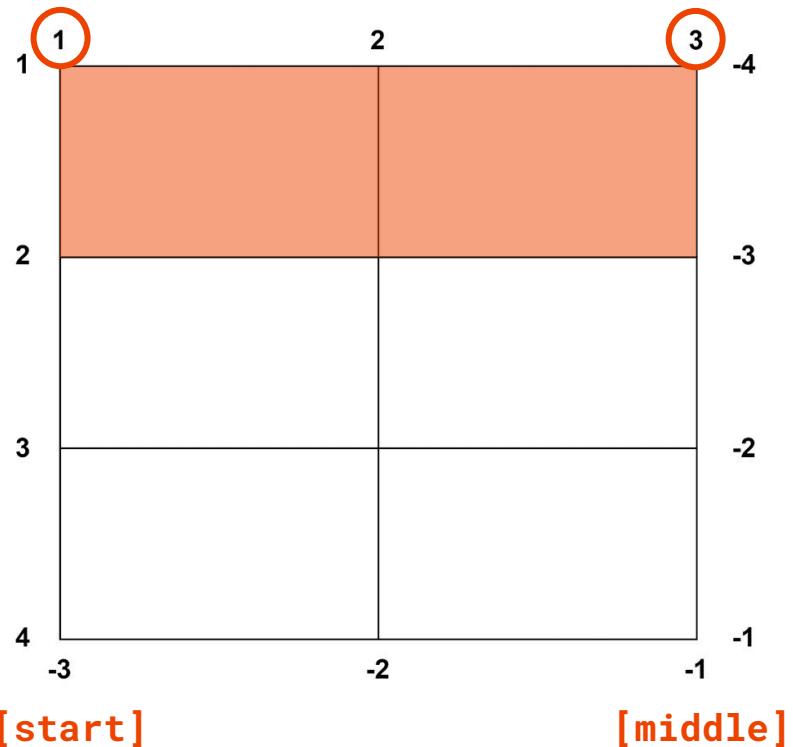
- **auto**

```
grid-column: 1 / auto;
```

It will occupy one cell, starting from the first line.

- If an element's position is not defined in the grid, it will be placed in the first available cell.

- The element will automatically determine where to start and end.



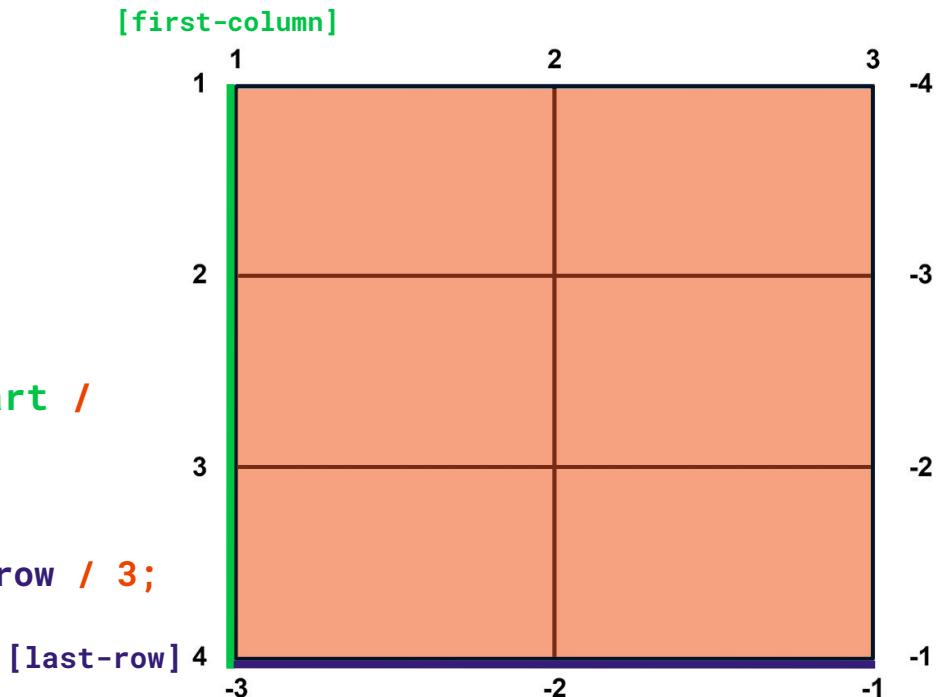
# Grid-area

**grid-area** Shorthand property for:

**grid-row-start** +  
**grid-column-start** +  
**grid-row-end** +  
**grid-column-end**

**grid-area: row-start / column-start /  
row-end / column-end;**

**grid-area: 1 / first-column / last-row / 3;**



# Positioning Elements in the Grid

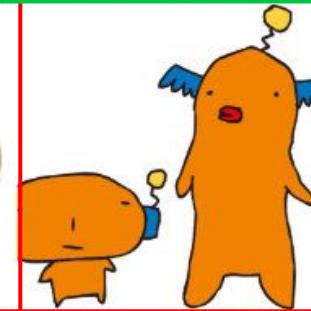
Practice



PATAPATA WORKS(パタパタワークス)は、福岡県福岡市でWEBデザインを始めとしたデザイン業務、イラスト制作・キャラクターデザインなどを行っているフリーランスデザイナーです。  
ホームページ制作やイラスト制作などでお困りでしたらお気軽にお問い合わせください。



Design & Illustration  
**PATAPATA**  
WORKS



```
<main class="grid-container">
    <article class="grid-cell-description"></article>
    <article class="grid-cell-jewelry"></article>
    <article class="grid-cell-fox"></article>
    <article class="grid-cell-dukes"></article>
    <article class="grid-cell-girl"></article>
    <article class="grid-cell-rainbow"></article>
</main>
```

```
.grid-cell-description {
    grid-column: 1 / 3;
    grid-row: 1 / 4;
}
```

```
.grid-cell-jewelry {
    grid-column: 3 / span 2;
}
```

Demo

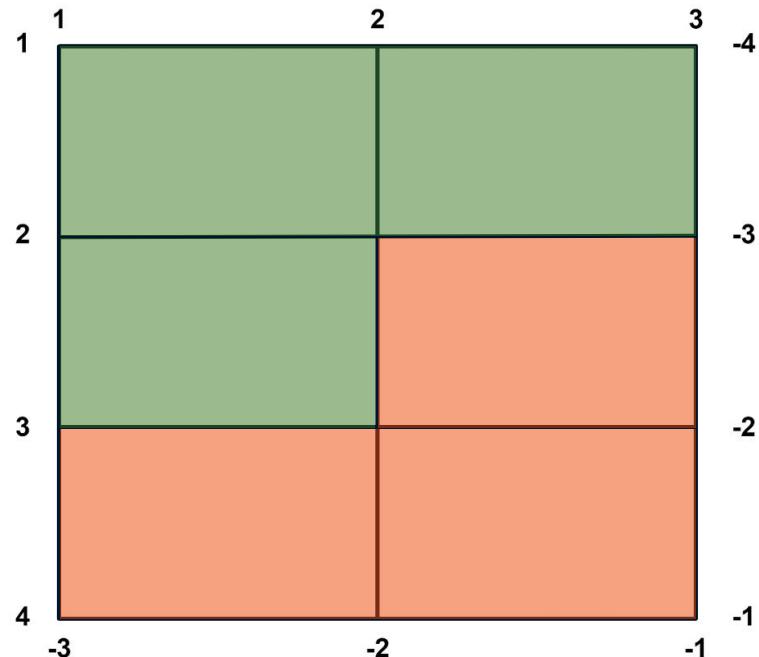
# Grouping Elements in the Grid

# display: contents

Instructs the browser to ignore this element and instead use its child elements as grid items.

Applied to a **direct descendant** of the grid.

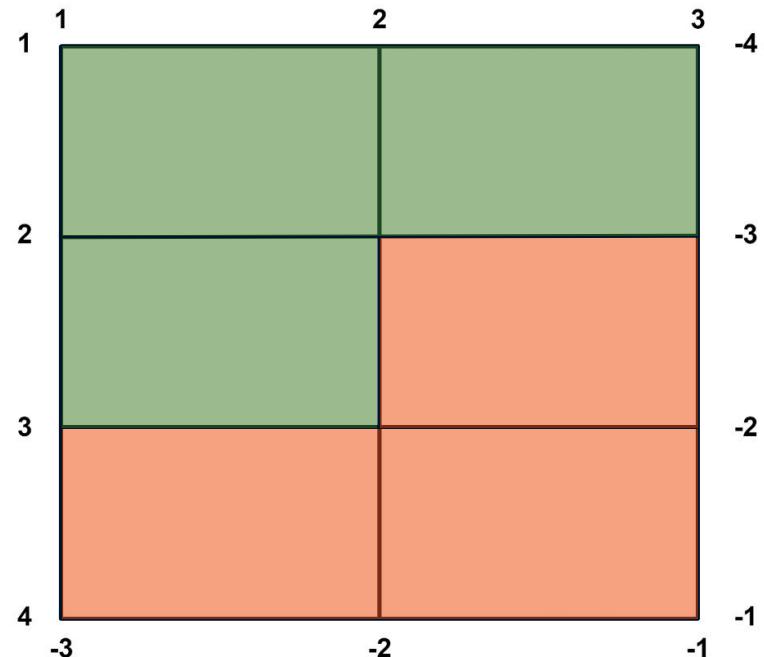
Used when it is necessary to **group** grid elements or **apply different styles** to such groups.



# display: contents

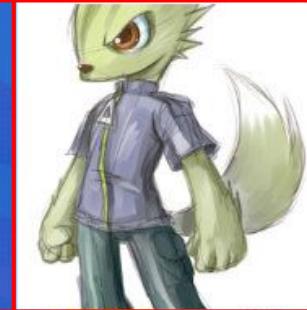
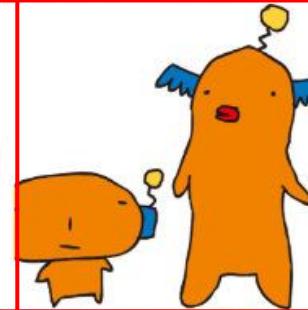
```
<div class="grid-container">
  <section class="new-articles">
    <article></article> x3
  </section>
  <section class="old-articles">
    <article></article> x3
  </section>
</div>
```

```
section {
  display: contents;
}
```



# Grouping Elements in the Grid

Practice

1	2	3	4	5
1				
2	<p>PATAPATA WORKS(パタパタワークス)は、福岡県福岡市でWEBデザインを始めとしたデザイン業務、イラスト制作・キャラクターデザインなどを行っているフリーランスデザイナーです。</p> <p>ホームページ制作やイラスト制作などでお困りでしたらお気軽にお問い合わせください。</p>			
3				
4				

```
<main class="grid-container">
  <article class="grid-cell-description"></article>
  <section class="articles">
    <article class="grid-cell-jewelry"></article>
    <article class="grid-cell-fox"></article>
    <article class="grid-cell-dukes"></article>
    <article class="grid-cell-girl"></article>
    <article class="grid-cell-rainbow"></article>
  </section>
</main>
```

```
.articles{
  display: contents;
}
```

[Demo](#)

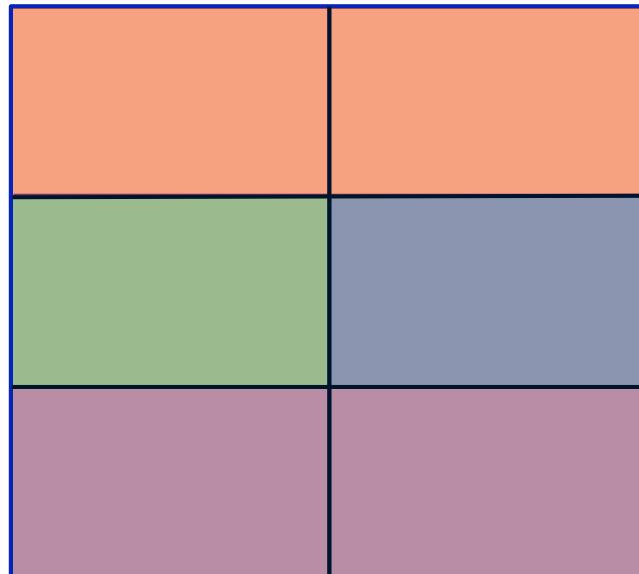
# Grid **areas**

# Grid template

**grid-template-areas** define the grid visually

Applied to a grid container

**grid-template-areas:**  
“header header”  
“sidebar content”  
“footer footer”;



# Grid area

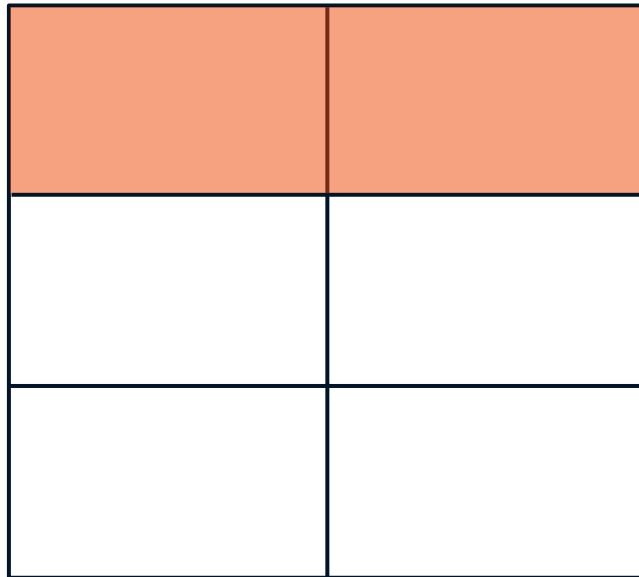
The area name is set for the grid cell

```
.header {  
  grid-area: header;  
}
```

It's better to name based on the function the cell performs:

Not recommended  
Names:  
`a`  
`item1`  
`block-c`

Recommended  
Names:  
`catalog-item`  
`article-header`  
`footer-logo`



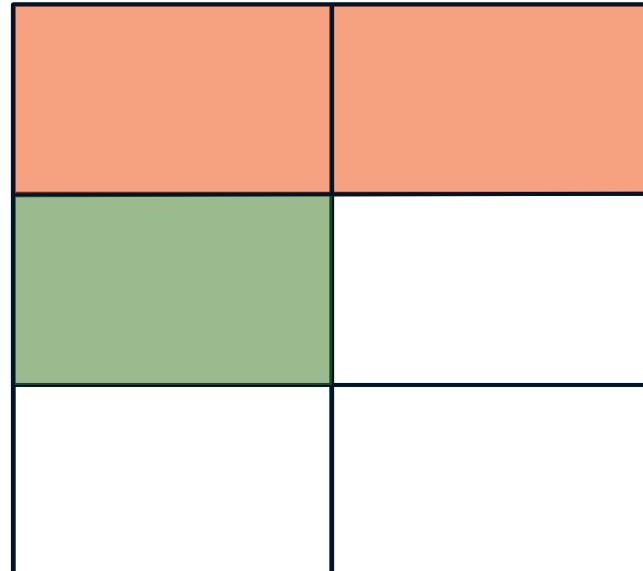
# Empty Grid Cells

An empty area or an area without a name is denoted by a ''.

```
grid-template-areas:  
  "header header"  
  "sidebar .";
```

can use **multiple** full-stop:  
"sidebar ..."; 

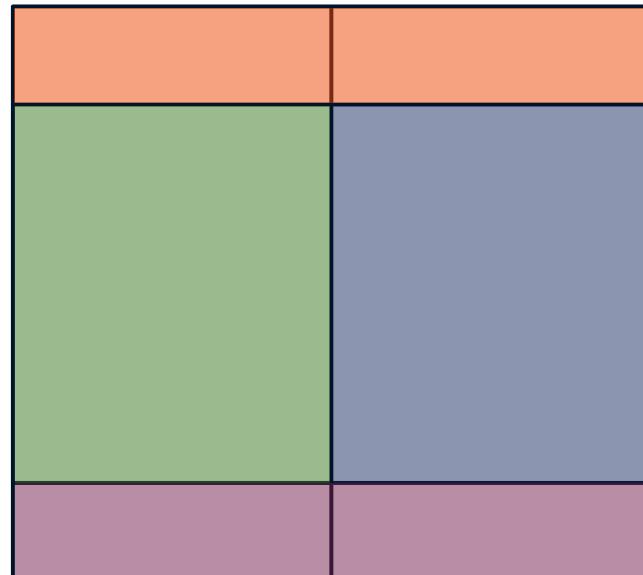
**don't use just space:**  
"sidebar "; 



# Grid areas shorthand

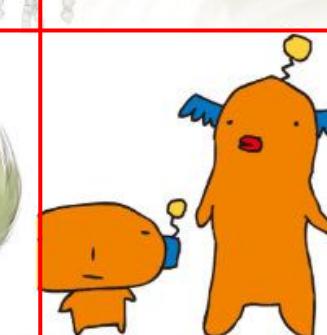
Shorthand for – grid-template  
= **grid-template-areas** +  
**grid-template-rows** /  
**grid-template-columns**;

```
grid-template:  
“header header” 100px  
“sidebar content” 1fr  
“footer footer” 100px / 50% 1fr;
```



# Rebuild everything using grid-area

**Practice**

1	2	3	4	5
1				
2	<p>PATAPATA WORKS(パタパタワークス)は、福岡県福岡市でWEBデザインを始めとしたデザイン業務、イラスト制作・キャラクターデザインなどを行っているフリーランスデザイナーです。</p> <p>ホームページ制作やイラスト制作などでお困りでしたらお気軽にお問い合わせください。</p>			
3				
4				

```
<main class="grid-container">
  <article class="grid-cell-description"></article>
  <article class="grid-cell-jewelry"></article>
  <article class="grid-cell-fox"></article>
  <article class="grid-cell-dukes"></article>
  <article class="grid-cell-girl"></article>
  <article class="grid-cell-rainbow"></article>
</main>
```

```
.grid-container {
  grid-template-areas:
    "description description jewelry jewelry"
    "description description fox dudes"
    "description description girl rainbow";
}

.grid-cell-description {
  grid-area: description;
}
```

[Demo](#)

# Implicit grids

# Implicit grids

- are automatically formed when there are more elements than the cells defined by:

**grid-template-columns /  
grid-template-rows**

**grid-template-columns: 1fr 1fr;  
grid-template-rows: 1fr 1fr;**

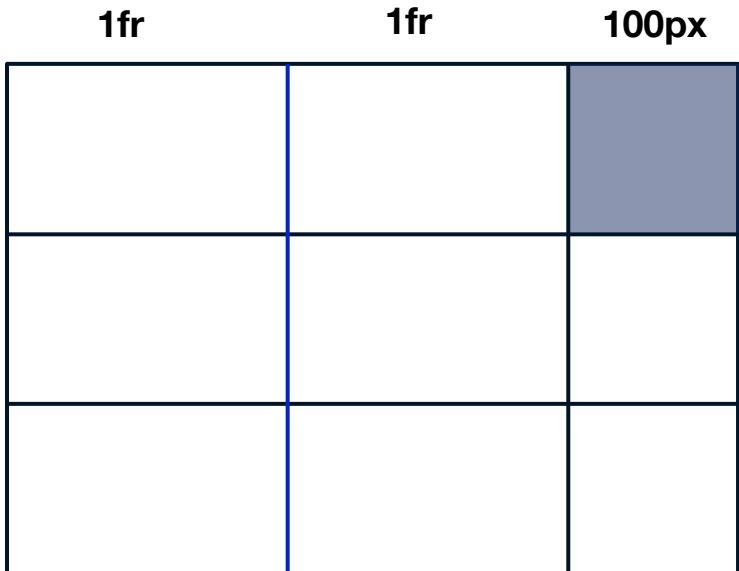
*If there are 4 elements in the container,  
they will fill it completely,  
but if there are 5, a new row will be created.*


# Managing Implicit Grids

```
grid-auto-columns /  
grid-auto-rows
```

```
grid-template-columns: repeat(2,  
1fr);  
grid-auto-columns: 100px;
```

*Each automatically created column will have a width of 100px*



# Auto Placement in the Grid

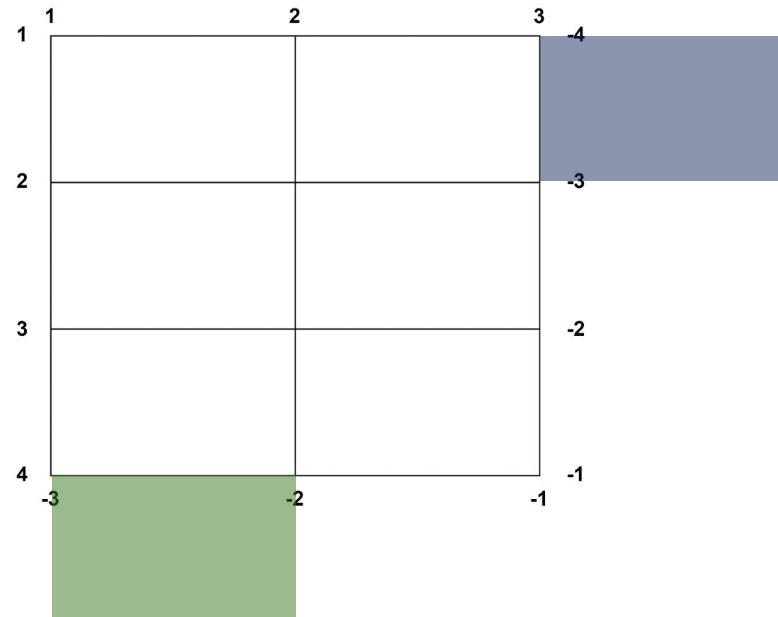
# Grid auto flow

- Controls the automatic placement of elements within the grid

Applied to **grid container**

```
grid-auto-flow: row;  
                  column;
```

A new column/row will be created, even if it was not specified in the original template.

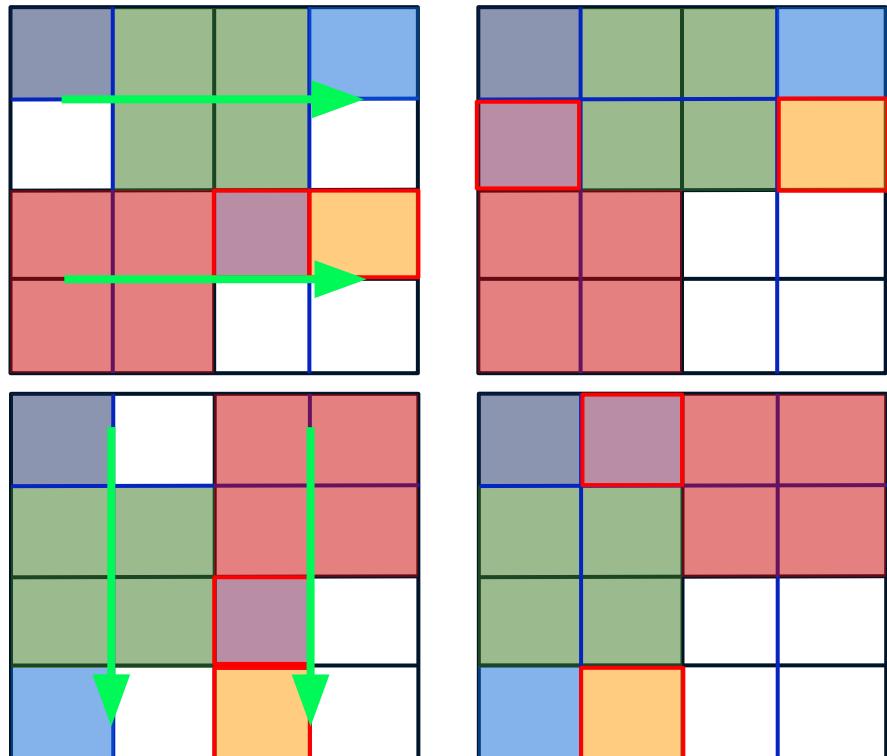


# Grid auto flow

**grid-auto-flow: dense;** – tries to fill any gaps in the grid, regardless of their location.

**row dense;**  
**column dense;**

– they first attempt to fill rows and columns, respectively, and then fill any remaining gaps.



# Grid auto-fill

- Automatically creating columns or rows based on available space

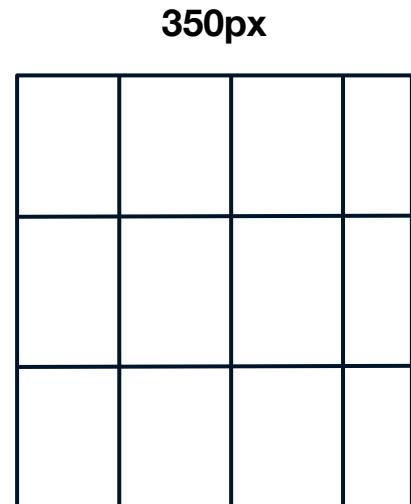
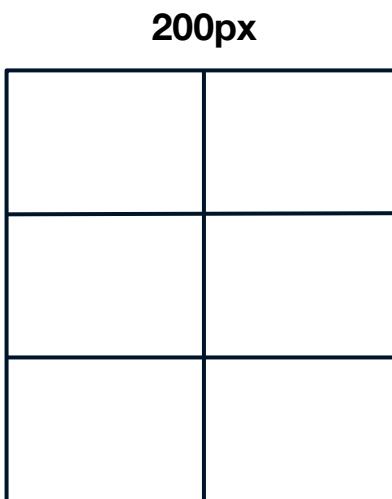
The remaining space stays empty.

Used with:

`grid-template-columns`  
`grid-template-rows`

`grid-template-columns:`  
`repeat(auto-fill, 100px);`

*will create columns of 100px based on the container's width*



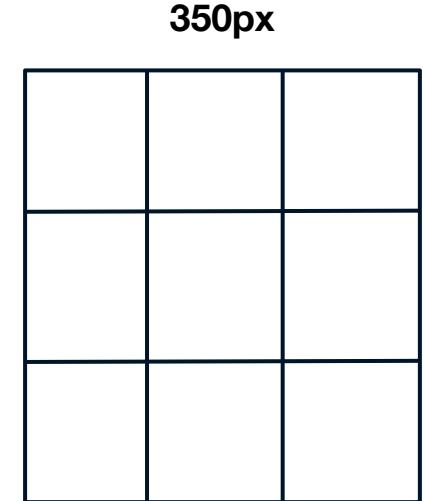
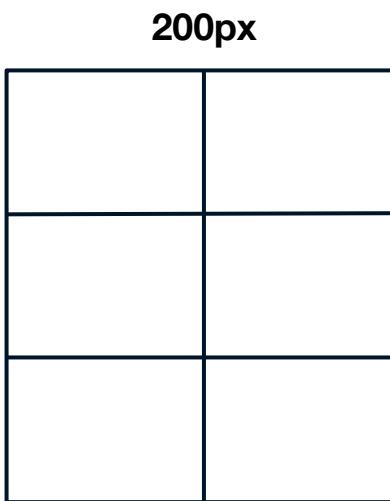
# Grid auto-fit

**Removes empty columns or rows and evenly distributes their space among remaining elements**

**Used with:**

**grid-template-columns  
grid-template-rows**

**grid-template-columns:  
repeat(auto-fit,  
minmax(100px, 1fr));**

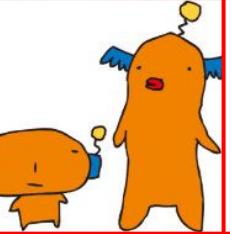
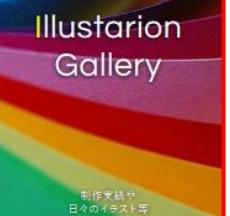


*will create columns with a minimum width of 100px.*

# Adding New Elements to the Grid

Practice



	1	2	3	4	5
1/-5					
2/-4	<p>PATAPATA WORKS(パタパタワークス)は、福岡県福岡市でWEBデザインを始めとしたデザイン業務、イラスト制作、キャラクターデザインなどをしているフリーランスデザイナーです。</p> <p>ホームページ制作やイラスト制作などでお困りでしたらお気軽にお問い合わせください。</p>				
3/-3					
4/-2					
5/-1					

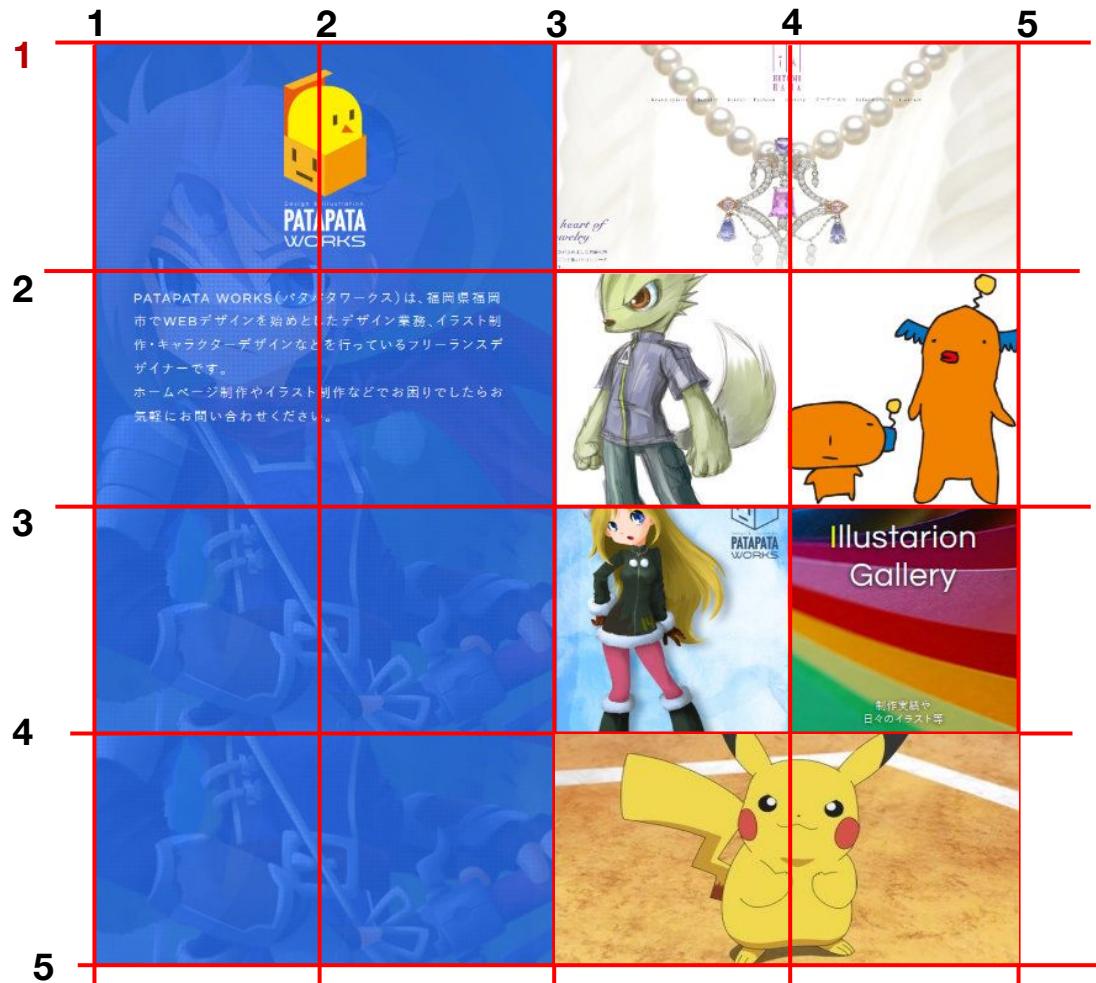
```
<main class="grid-container">
  <article class="grid-cell-description"></article>
  <article class="grid-cell-jewelry"></article>
  <article class="grid-cell-fox"></article>
  <article class="grid-cell-dukes"></article>
  <article class="grid-cell-girl"></article>
  <article class="grid-cell-rainbow"></article>
</main>
```

```
.grid-container {
  grid-auto-flow: row;
}
```

```
.grid-cell-description {
  grid-row: 1 / -1;
}
```

[Demo](#)

\*



Create a grid that repeats the following pattern:

One element takes up two cells.

The next four elements take up one cell each.

Then another element takes up two cells, followed by four elements taking up one cell each.

And so on, infinitely.

# Aligning Elements Inside Cells

# Grid align

The alignment of cells and rows is controlled by (*almost*) the same properties as in flexbox

`align-items + justify-items`  
`align-self + justify-self`  
`align-self + justify-self`  
`align-content + justify-content`

With the same values, but the axes are always fixed.

`align-*` aligns along the column axis  
`justify-*` aligns along the row axis


# Grid align values

**start** – elements are aligned to the start of the container.

**end** – elements are aligned to the end of the container.

**center** – elements are centered.

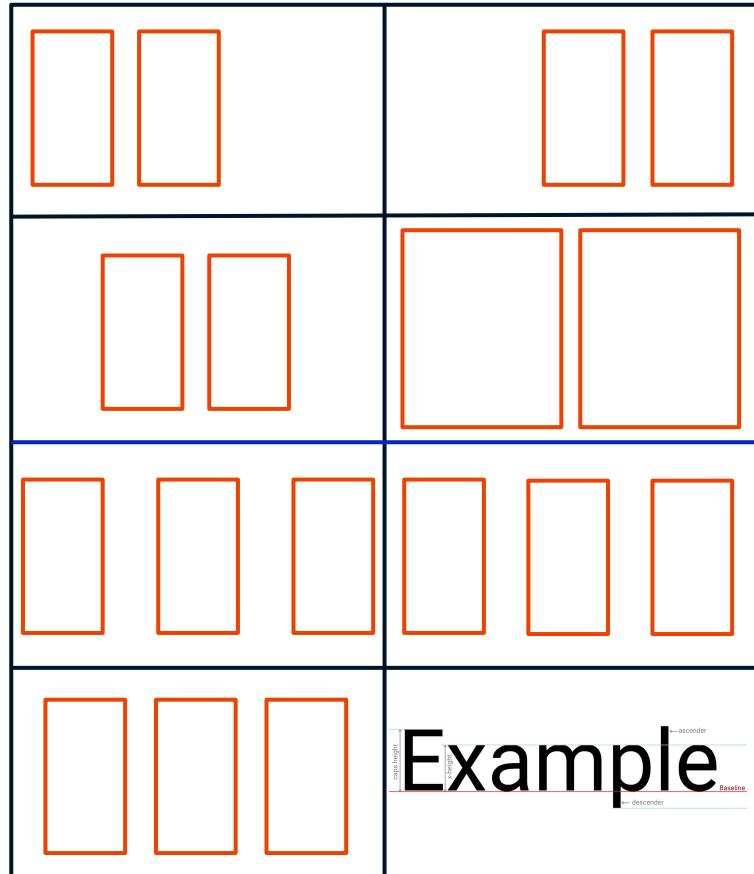
**stretch** – elements stretch to fill the entire available space of the grid item; this is the default value.

**space-between** – elements are distributed with equal spacing between them, and the outer elements are aligned with the edges.

**space-around** – elements are distributed with equal spacing between them, but the spacing at the edges is half of the spacing between the elements.

**space-evenly** – elements are distributed with equal spacing between them, with the same spacing at the edges as between the elements.

**baseline** – aligns grid items along the text's baseline in the entire row. This value is particularly useful for text of different sizes that needs to be aligned on the same line.



# Grid align items

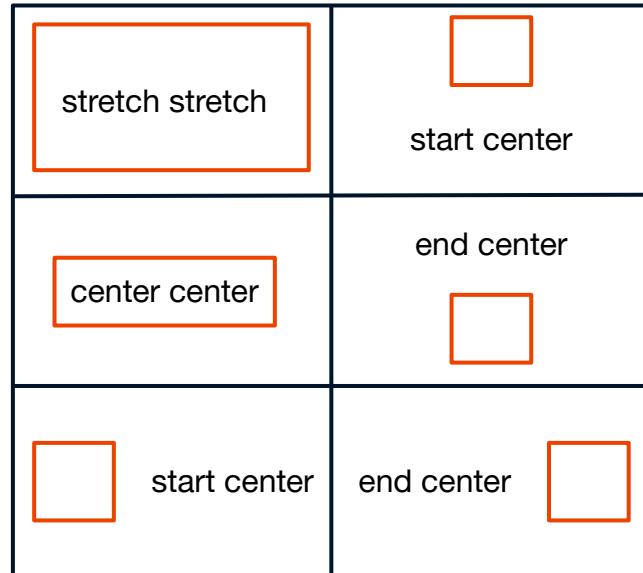
**place-items** – aligns items within the cells

= **align-items** + **justify-items**

Applied to a grid container

**place-items: center center;**

start  
end  
center  
stretch



# Grid align item

**place-self** – aligns a specific item  
within a cell

= **align-self** + **justify-self**

Applied to a **grid cell**

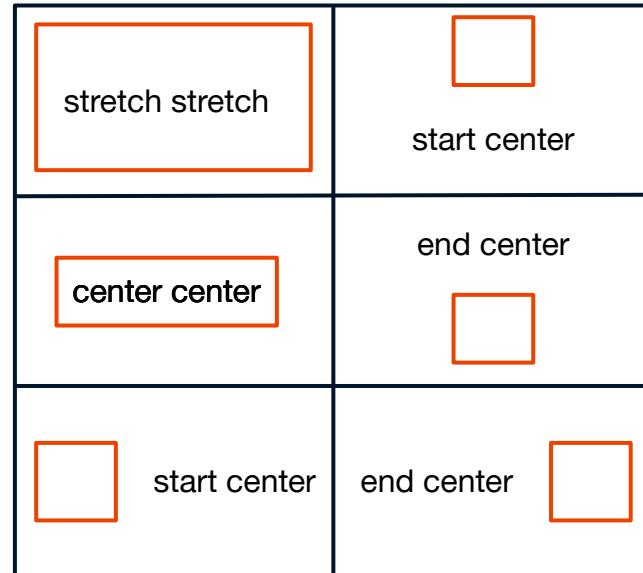
**place-self: center center;**

start

end

center

stretch



# Grid align rows/columns

**place-content** – aligns rows within the container

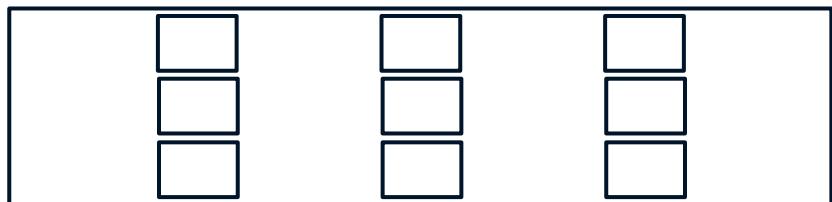
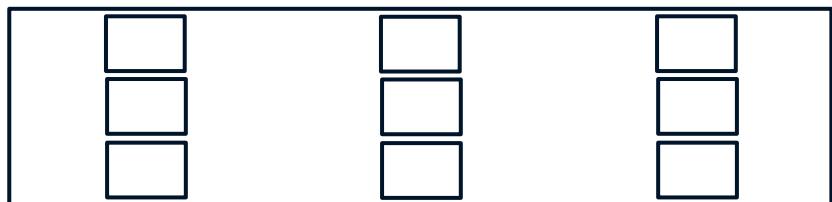
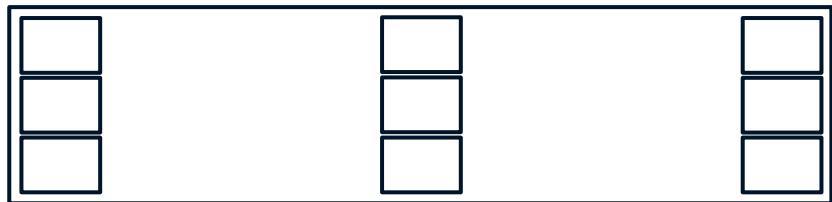
Sometimes the total grid size may be smaller than the size of its container, for example, when grid item sizes are set in pixels.

= **align-content** + **justify-content**

Applied to a **grid container**

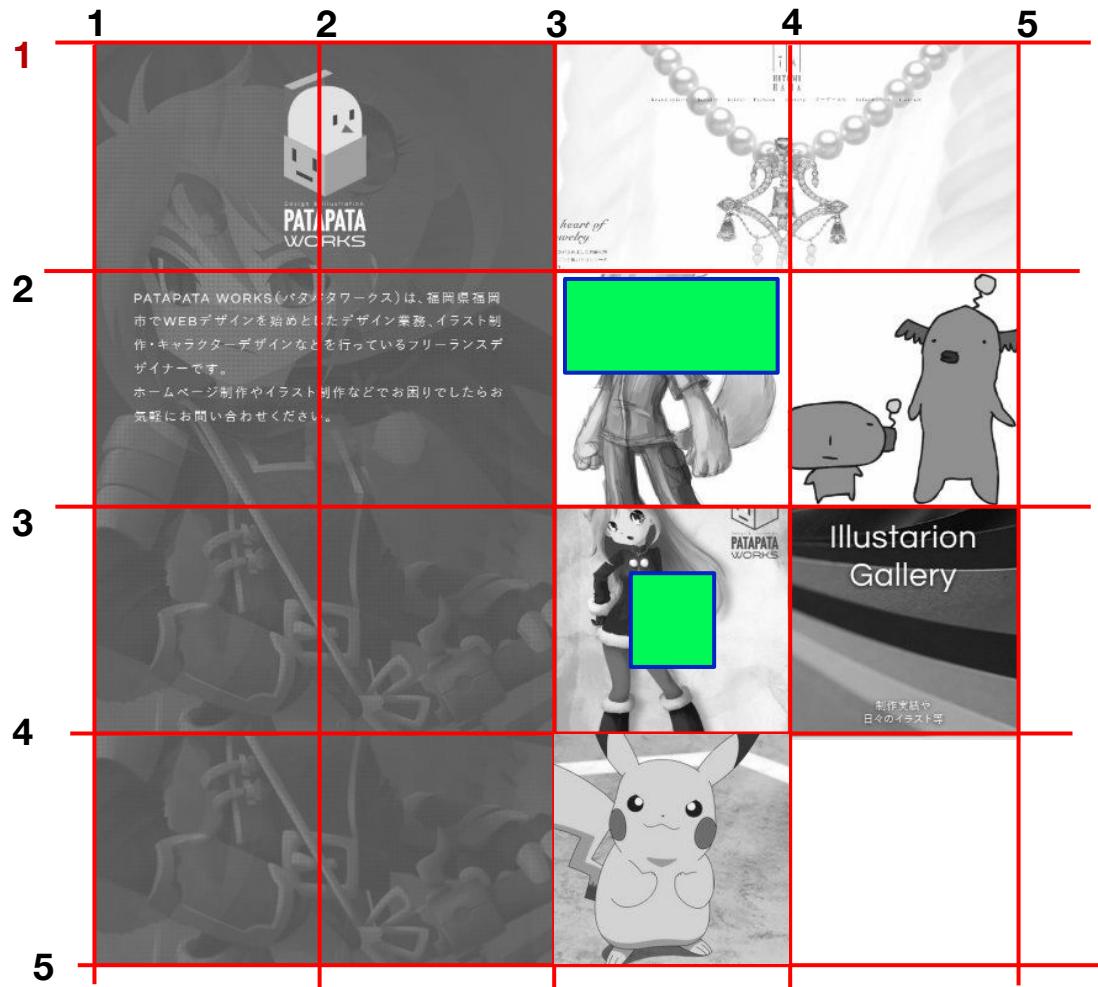
**place-content: space-between center;**

start  
end  
center  
stretch  
space-between  
space-around  
space-evenly



# Aligning Elements Inside Cells

Practice



Demo

Chrome **Dev Tools**

Fruit box

Apple	Kiwi	Mango	
Orange			

```
.fruit-box {
  display: grid;
  grid-gap: 10px;
  grid-template-columns:
    [left] 1fr
    [middle1] 1fr
    [middle2] 1fr
    [right];
}

.orange {
  background: orange;
  grid-column: left / right;
}
```

Elements Console > ⚠ 1 ⚙ ⋮ ×

```
<style>...</style>
<main> grid
<p>Fruit box</p>
...
<div class="fruit-box"> grid $0
  <div class="apple">Apple</div>
  <div class="kiwi">Kiwi</div>
```

html body main div.fruit-box

Styles Computed Layout Event Listeners >

Filter :hov .cls + □

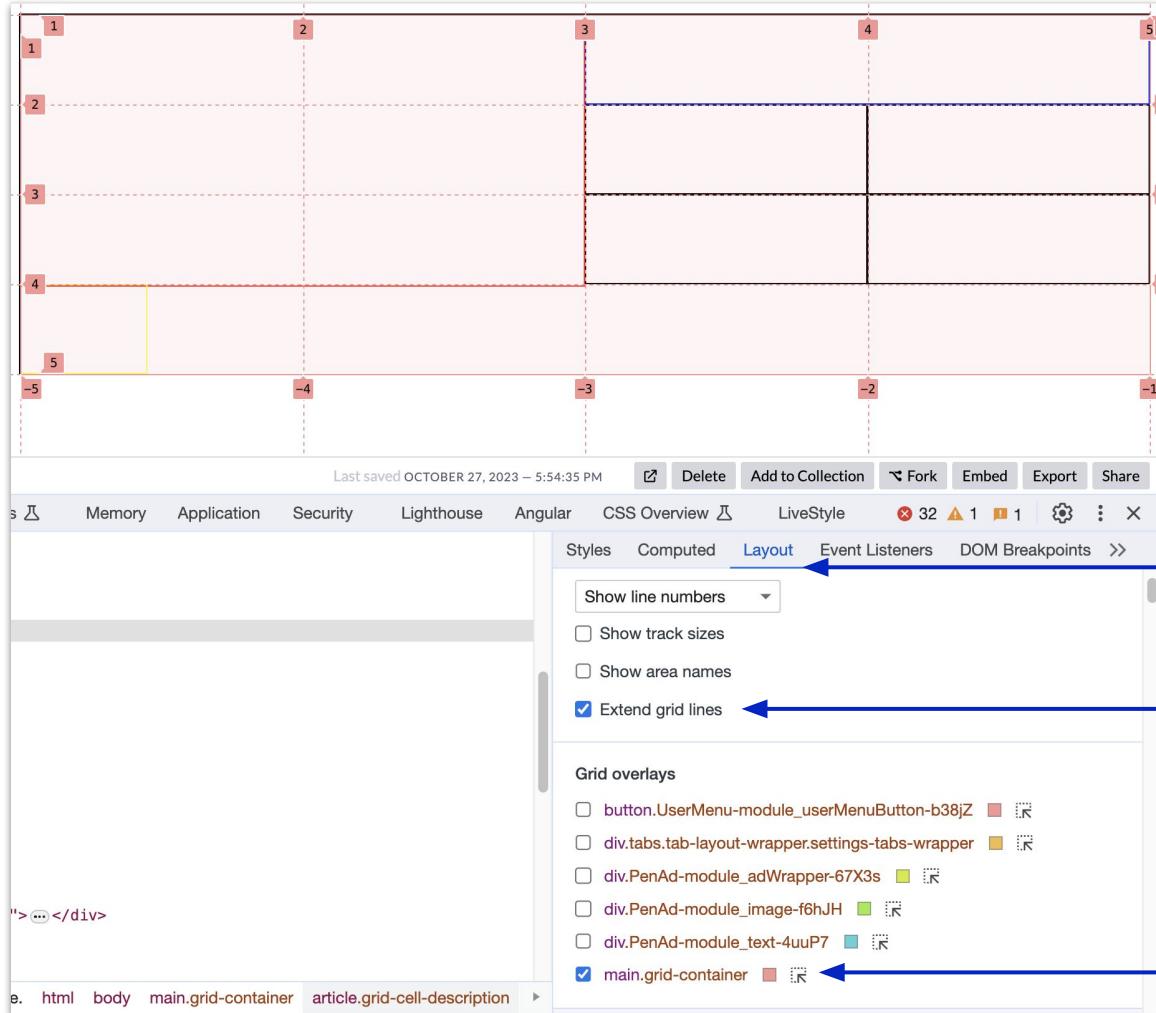
```
element.style {
}

.fruit-box {
  display: grid;
  grid-gap: 10px;
  grid-template-columns: [left] 1fr [middle1] 1fr
    [middle2] 1fr [right];
  border: 2px solid;
  width: 300px;
  padding: 4px;
}

div {
  display: block;
```

css-grid-fruit:1 user agent stylesheet

1



CSS grid- fruit

jec.fyi/demo/css-grid-fruit

Fruit box

Apple Kiwi Mango

Orange

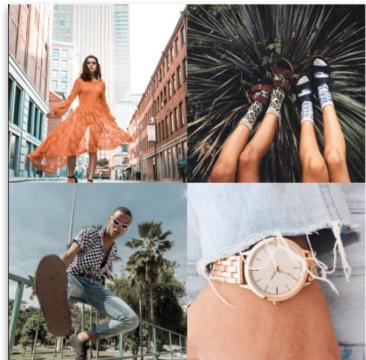
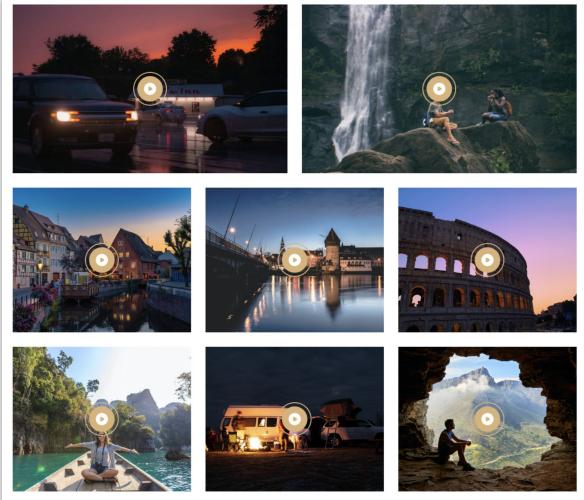
justify-content: start

justify-items: stretch

```
.fruit-box {  
    display: grid;  
    grid-gap: 10px;  
    grid-template-columns: [left] 1fr [middle1] 1fr [middle2] 1fr [right];  
    border: 2px solid;
```

# Grid examples





Instashop

Appear, dry there darkness they're seas, dry  
waters thing fly midst. Beast, above fly  
brought Very green.

[Go to Instagram](#)



**HARVEST**

### Tea Plucking

In the spring and summer, you should notice a fresh flush of young leaves and these are the perfect ones for tea making and the most prized by tea growers! Pick the two youngest leaves and the bud of each branch to give yourself a small pile of fresh soft leaves to work with.



**STEAMING**

### Heat the leaves by steaming them

A steamer works best or you can use a colander over a pan of boiling water; steam for 1-2 minutes. You should never heat the buds (temperature to 45°) and if you can't carefully handle them and dry them for them to go limp, you'll know they are ready for the next step.



**SHAPING**

### Retain tea leaves freshness

Is the process by which the leaf is twisted and the leaf cell walls ruptured to bring the juices to the surface of the leaf. The rolling machines have deep pockets, a pressure cap to apply pressure on the leaf, and the table itself has bathers and a cone at the center to twist the leaf.



**DRYING**

### Keep the tea moisture-free

Drying enhances a tea's flavors and ensures its long shelf-life. To dry the leaves they are fired or roasted at a low temperature for a continued period of time, usually inside an industrial-scale oven. If the leaves are dried too quickly, the tea can turn abrasive and taste harsh.



**2012**

Exercitation photo booth stumptown tote bag Banksy, elit small batch freegan sed. Craft beer elit seitan exercitation, photo booth et 8-bit kale chips prident chillwave deep v laborum. Aliquip veniam delectus. Marfa eiusmod Pinterest in do umami readymade swag.



**2015**

Exercitation photo booth stumptown tote bag Banksy, elit small batch freegan sed. Craft beer elit seitan exercitation, photo booth et 8-bit kale chips prident chillwave deep v laborum. Aliquip veniam delectus. Marfa eiusmod Pinterest in do umami readymade swag.



**2018**

Exercitation photo booth stumptown tote bag Banksy, elit small batch freegan sed. Craft beer elit seitan exercitation, photo booth et 8-bit kale chips prident chillwave deep v laborum. Aliquip veniam delectus. Marfa eiusmod Pinterest in do umami readymade swag.



**2020**

Exercitation photo booth stumptown tote bag Banksy, elit small batch freegan sed. Craft beer elit seitan exercitation, photo booth et 8-bit kale chips prident chillwave deep v laborum. Aliquip veniam delectus. Marfa eiusmod Pinterest in do umami readymade swag.

## Collection

The taste for design, photography and art in general, has led us to focus on this project. Example of fashion store with modern layout.



Cotton sweatshirt  
\$ 12.99



Find out about all our news in our newsletter:

Your email

Subscribe

# Our Products

Enjoy a wide range of fresh and organic products, right from our farms. You will love them!!

[View All Products](#)



Spices



Grains



Vegetables



Fruits



Dairy





NextCrop

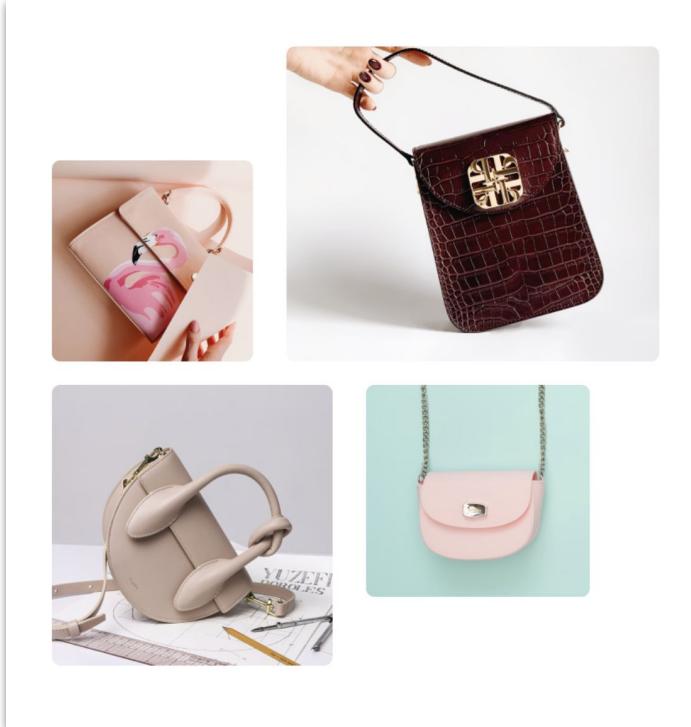
Home About us Shop Services Contact

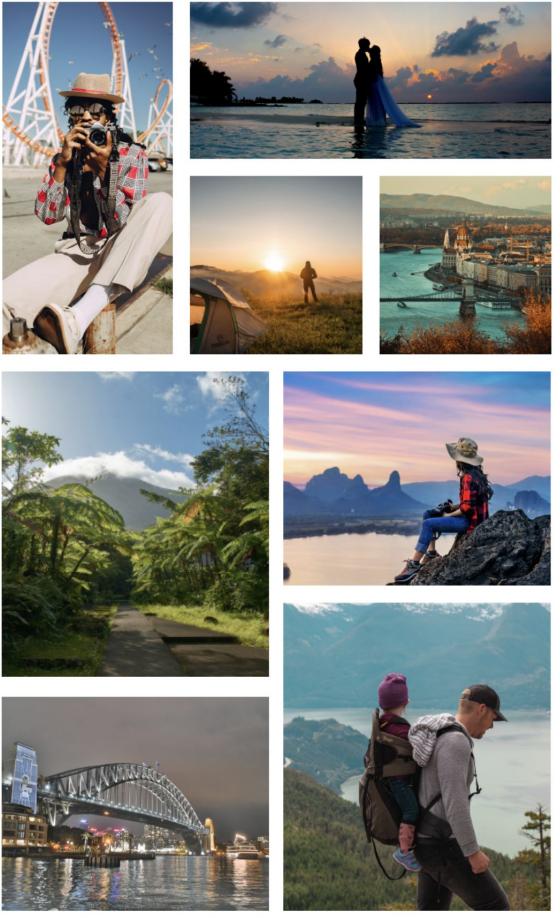
Request a Demo

## The Next Generation of Crops

Improve your crops with reduced pesticides, fast growth yet still full of nutritious elements

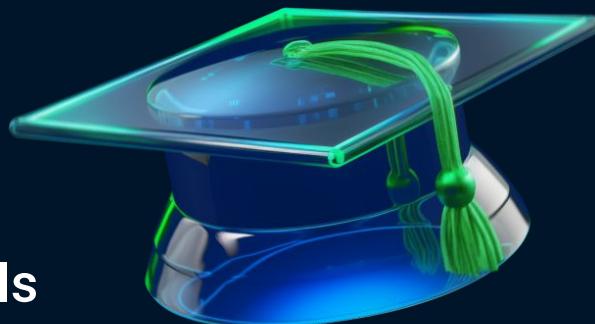
Get Started





# Summary

1. What is Grid, and where to use it
2. How to build a layout using Grid
3. How to position elements in this grid
4. How to name areas in Grid
5. How to align elements inside Grid cells
6. What are implicit grids
7. How to work with Grid in Chrome Dev Tools



# Homework

1. Learn grid with a game.
2. Add theory with a course.
3. Add grids to your project where they are more convenient than flex layouts.





# QUESTIONS?



**Please fill out the feedback form**  
**It's very important for us**



**THANK YOU!**  
**Have a good evening!**