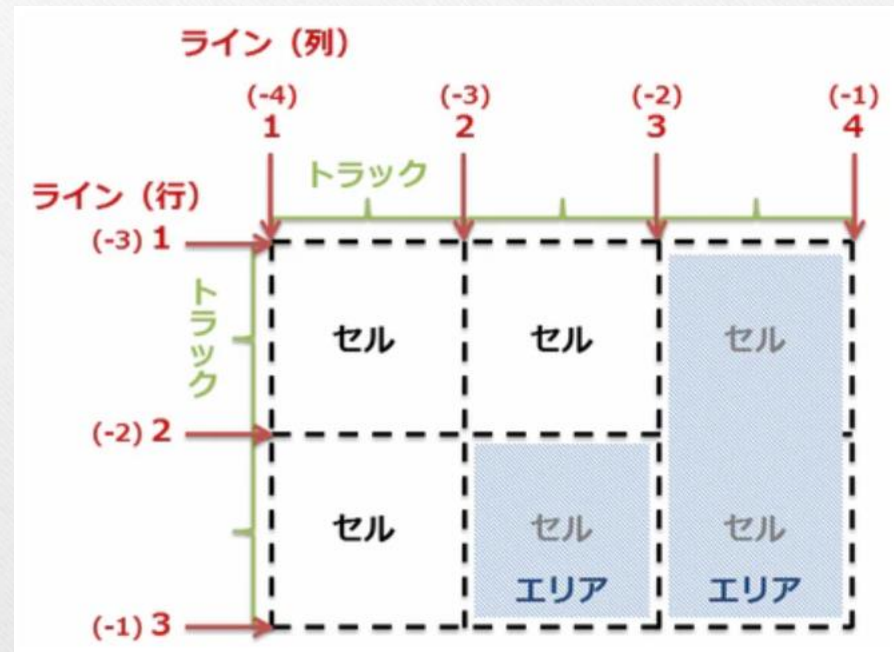


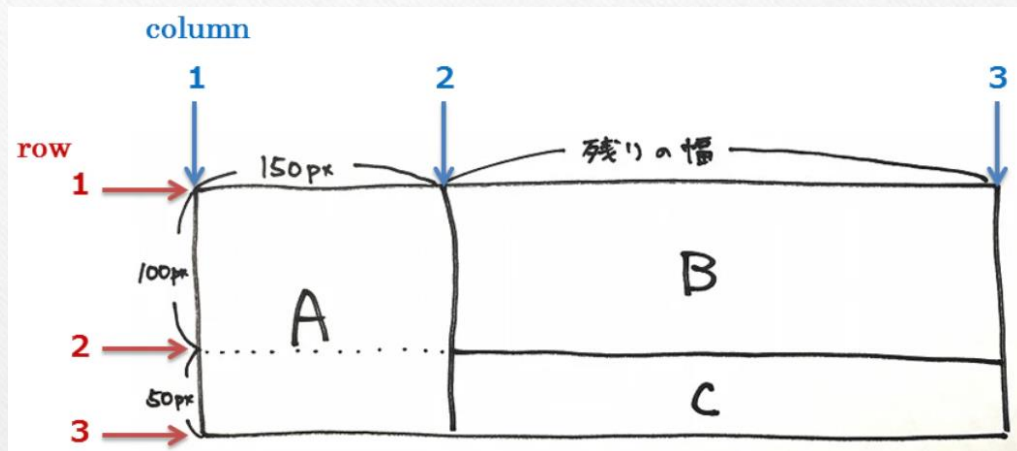
Grid レイアウト

グリッドレイアウトは2次元レイアウトとも呼ばれ、HTML、CSSを使って水平方向、垂直方向の両方に沿って要素を配置できます。



Grid レイアウト(ラインの番号で指定する)

グリッドレイアウトは2次元レイアウトとも呼ばれ、HTML、CSSを使って水平方向、垂直方向の両方に沿って要素を配置できます。



HTML

```
<div id="container"> <!-- コンテナ -->
  <div id="itemA">A</div> <!-- アイテム -->
  <div id="itemB">B</div> <!-- アイテム -->
  <div id="itemC">C</div> <!-- アイテム -->
</div>
```

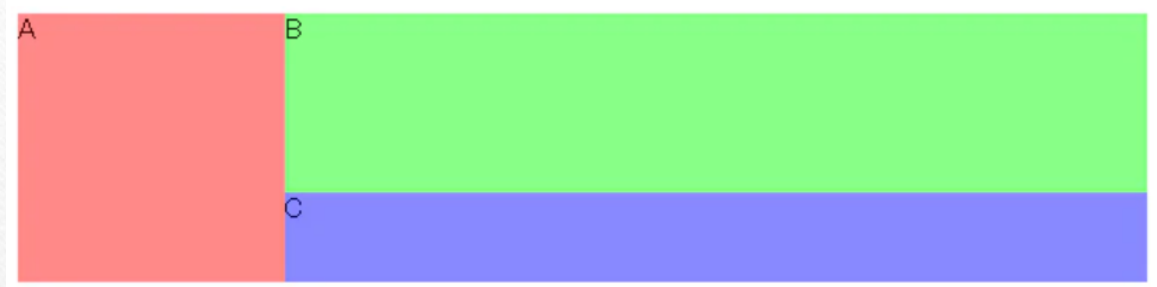
コンテナのCSS

- **grid-template-rows:**
行のトラックの高さを半角スペースで区切って指定
- **grid-template-columns:**
列のトラックの幅を半角スペースで区切って指定

CSS

```
#container {  
  display: grid;  
  grid-template-rows: 100px 50px;  
  grid-template-columns: 150px 1fr;  
}
```

アイテムのCSS



CSS

```
#itemA {  
  grid-row: 1 / 3;  
  grid-column: 1 / 2;  
  background: #8f8;  
}
```

```
#itemB {  
  grid-row: 1 / 2;  
  grid-column: 2 / 3;  
  background: #f88;  
}
```

```
#itemC { grid-row: 2 / 3;  
  grid-column: 2 / 3;  
  background: #88f; }
```


Grid レイアウト(エリアに名前を付けて指定する)



HTML

```
<div id="container"> <!-- コンテナ -->
  <div id="itemA">A</div> <!-- アイテム -->
  <div id="itemB">B</div> <!-- アイテム -->
  <div id="itemC">C</div> <!-- アイテム -->
</div>
```

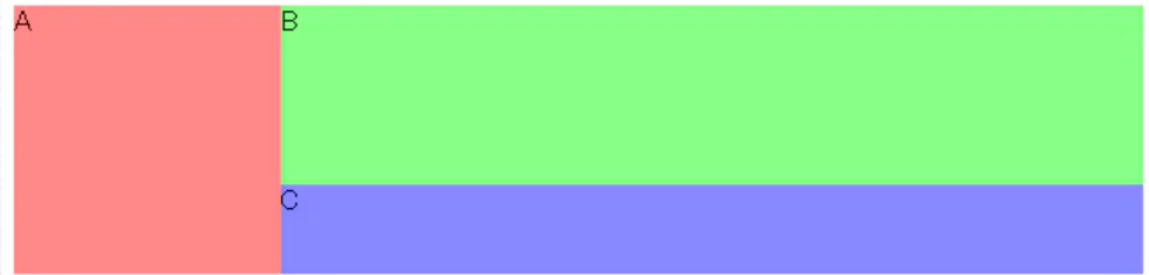
コンテナのCSS

- **grid-template-areas:**
エリアを作ってそれぞれに名前を付ける（コンテナのスタイルとして記述する）
- **grid-area:**
各アイテム が占めるエリアの名前を指定する（アイテムのスタイルとして記述する）

CSS

```
#container {  
  display: grid;  
  grid-template-rows: 100px 50px;  
  grid-template-columns: 150px 1fr;  
  grid-template-areas: "areaA areaB" "areaA areaC";  
}
```

アイテムのCSS



CSS

```
#itemA {  
  grid-area: areaA;  
  background: #8f8;  
}
```

```
#itemB {  
  grid-area: areaB;  
  background: #f88;  
}
```

```
#itemC {  
  grid-area: areaC;  
  background: #88f;  
}
```


Grid レイアウト(アイテムを動的に配置する)

コンテナの大きさに合わせて繰り返す

(repeat(auto-fill | auto-fit, ...))

「100px のトラックを、横に詰めれるだけ詰めたい！」

HTML

```
<div id="container">
  <div>A</div>
  <div>B</div>
  <div>C</div>
  <div>D</div>
  <div>E</div>
  <div>F</div>
  <div>G</div>
</div>
```

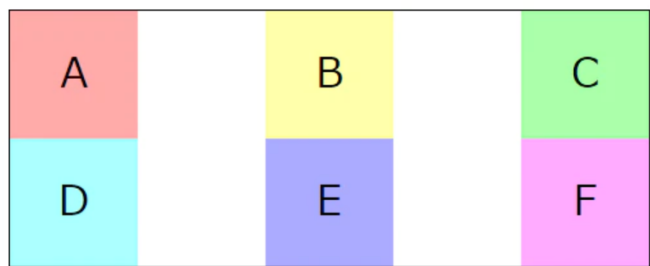
CSS

```
#container {
  display: grid;
  grid-auto-rows: 100px;
  grid-template-columns:
    repeat(auto-fit, 100px);
  justify-content: space-around;
  /* グリッド全体を均等に配置 */
}
```

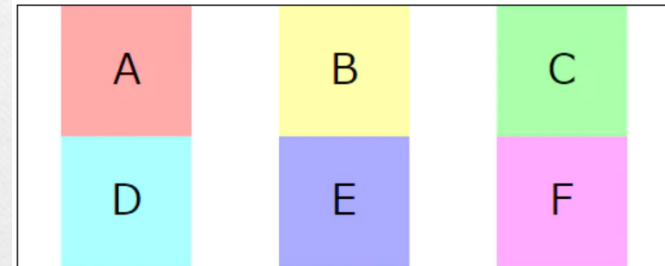

セル同士に隙間をあける(1) (grid-row-gap / grid-column-gap)

セル同士に隙間をあける(2) (justify-content / align-content)

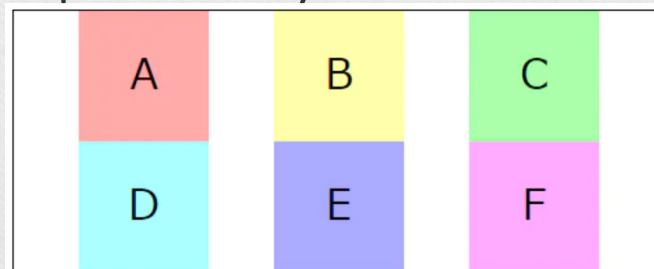
space-between



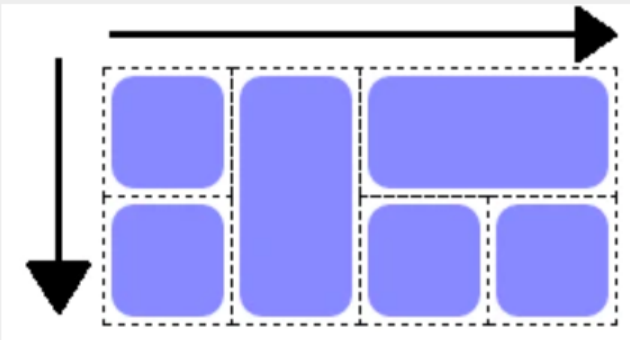
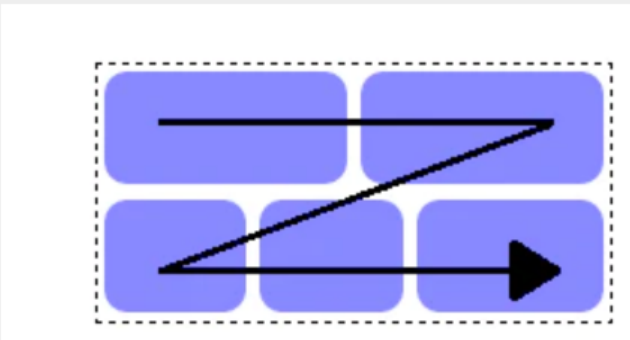
space-around



space-evenly



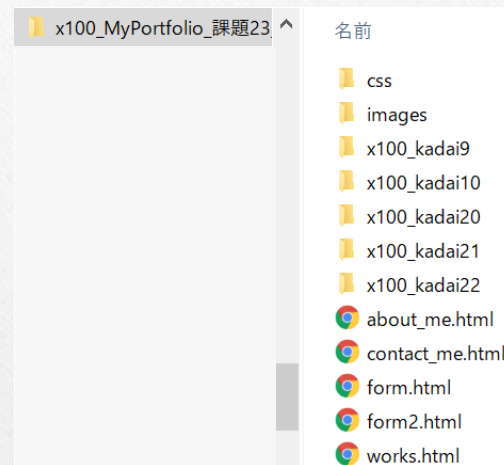
CSS Grid Layout と CSS Flexible Box Layout

	CSS Grid Layout	CSS Flexible Box Layout
CSSの記述	display: grid ; /* or inline-grid */	display: flex ; /* or inline-flex */
配置の次元	2次元	1次元
	 <p>縦横自由に配置</p>	 <p>折り返しはするけど一方向</p>

完成例（x100_MyPortfolio_作成日¥works.html）課題8 x100_MyPortfolioをコピーして作成

素材の準備

- ① 課題9～課題22のフォルダを
コピーしてMyPortfolioに保存する。



- ② フォルダ名をローマ字に変更する。（課題9→kadai9）
- ③ 課題9～課題22までのキャプチャをとる。
- ④ それぞれのキャプチャを画像としてimagesに保存する。



works.htmlの作成 (about_me.htmlをコピー)

X100_works

← → ↺ 🏠 ⓘ ファイル | C:/

Taro Sundai

Works

[課題9](#)

[課題10](#)

[課題20](#)

[課題21](#)

[課題22](#)

Last Modified : 2021年11月15日

©Hanako Sundai, All Rights Reserved.

リンクをクリックすると
それぞれの課題ページ
が表示されます。


今後の課題は全
てviewportを入れ
てください。

猫の一日・・・

- ・ ひたすら寝ています
- ・ 猫は湿度に敏感な動物
- ・ 微かな湿気を感じ取って顔を洗う

ひたすら寝ています

猫は毎日12～16時間は睡眠をとると言われています。ただし、熟睡している時間は意外と少なく、ほとんどが浅い眠りです。覚悟がするとすぐ目を見ますのはそのせいなんです。



[戻る](#)


猫は湿度に敏感な動物

猫は湿度を敏感に感じ取ることができる動物だと言われています。湿度が高くなると、湿気によってヒゲに湿りがなくなったり、ヒゲや被毛が濡ってムズムズし、顔を洗うという行動をします。ヒゲは顔にとって様々なことを感じ取るレーダーのようなもので、湿度によってその感覚が鈍ってしまうのを防ぐために、猫は念入りに顔を洗うようです。

[戻る](#)

微かな湿気を感じ取って顔を洗う

雨が降る前であっても湿度は高くなっています。その湿気を感じ取って顔を洗いに流る行動が「猫が顔を洗うと雨が降る」と言われるようになった理由のひとつだと考えられます。勿論、猫が顔を洗う理由には他にもあるので、100%雨が降るかというわけではありません。しかし、目の後ろの方まで水入りで洗っているような、雨が降る確率はより高くなるとも言われています。



[戻る](#)

works.html(一部抜粋)

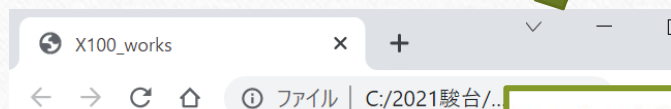
```
<!-- main Section -->
<main>
<!-- works Grid Section -->
<section>
<h2>Works</h2>
<div id="works">
<div>
  <p><a href="X100_kadai9/index.html">課題 9 </a></p>
</div>
```



各自で考えてください。

```
</section><!-- /works -->
</main><!-- /main -->
```

portfolio.cssの作成



Taro Sundai

Works

[課題9](#)

[課題10](#)

[課題20](#)

[課題21](#)

[課題22](#)

Last Modified : 2021年11月15日

©Hanako Sundai, All Rights Reserved.

幅:最大

Taro Sundai

Works



Last Modified : 2021年11月15日

©Hanako Sundai, All Rights Reserved.

幅:750px

Taro Sundai

Works



Last Modified : 2021年11月15日

©Hanako Sundai, All Rights Reserved.

Taro Sundai

Works

幅:500px



Last Modified : 2021年11月15日

©Hanako Sundai, All Rights Reserved.

今回はメディアクエリは使
いません。
gridレイアウトを使います。

portfolio.cssの作成(Gridレイアウト)

Taro Sundai

Works



リンクは画像内のどこでも反応するように以下のように工夫します。

```
#works a {  
  display: inline-block; /*ブロック要素にする*/  
  width: 200px; /*ブロック要素にしたので幅や高さが指定出来る*/  
  height: 200px;  
  font-size: 2em;  
  color: 自由  
  padding-top: 50px;  
}
```