

**Desenvolupament Digital**

CSS. Posicionament

## Posicionament per defecte

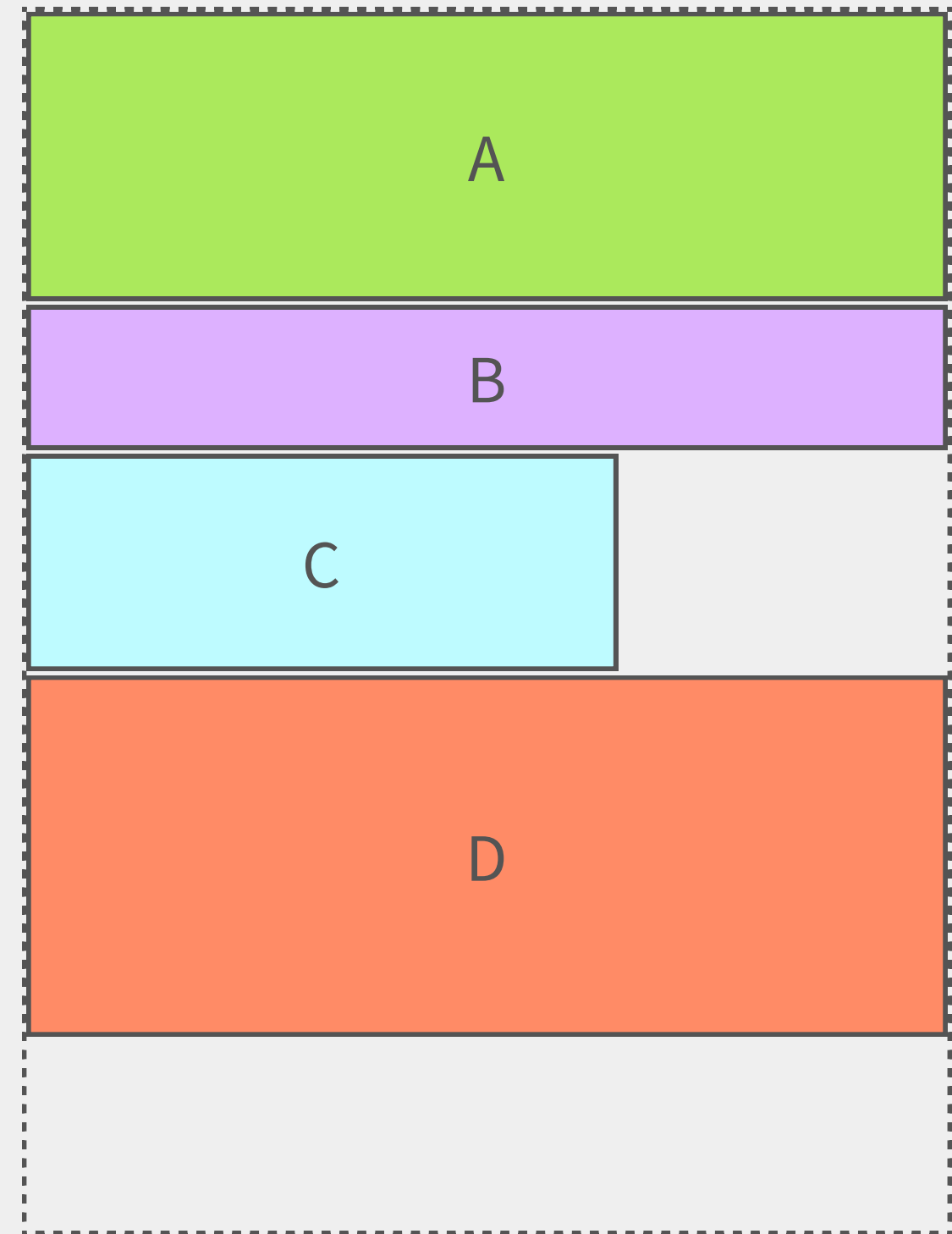
Per defecte, els elements d'un document HTML es posicionen **en relació als elements precedents i al seu contenidor**.

Aquest comportament es pot produir de dues maneres: **en bloc** o **en línia**.

## Elements en bloc

*article, aside, blockquote, div, fieldset, figcaption, figure, footer, form, h1, h2, h3, h4, h5, h6, header, hr, li, main, nav, ol, p, section, ul...*

Se situen **a sota de l'element precedent** i ocupen, per defecte, **tota l'amplada del seu contenidor**. Podem assignar-los una mida específica mitjançant CSS però, tot i així, no permetran que un altre element se situï al seu costat.

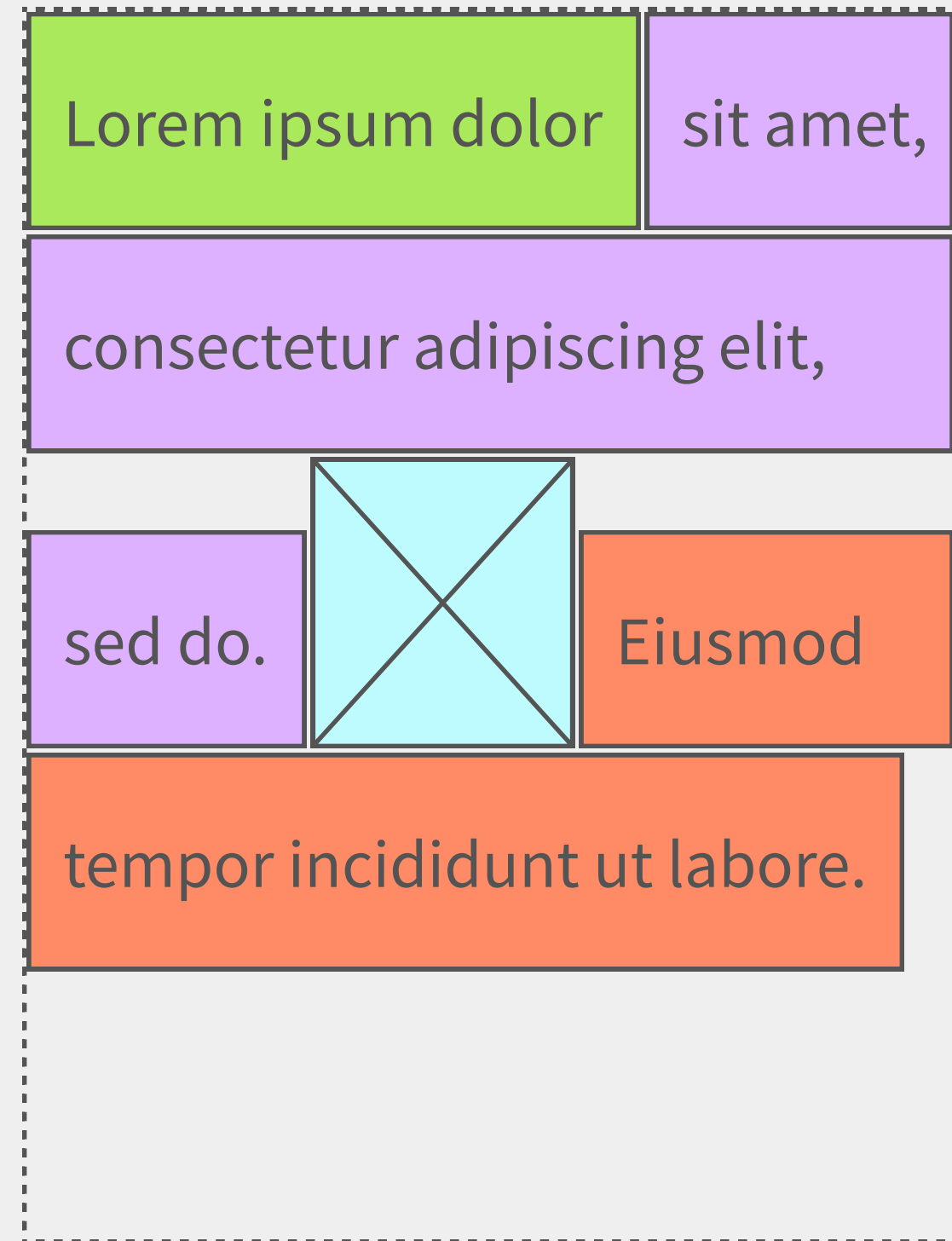


## Elements en línia

*a, b, button, cite, em, i, iframe, img, input, label, picture, q, select, span, strong, svg, textarea, video...*

Se situen **a la dreta de l'element o del text precedent** (en idiomes que es llegeixen d'esquerra a dreta) si aquest també és en línia i sempre que hi hagi espai disponible dins el seu contenidor. Si no, se situen a sota de l'anterior línia d'elements. Si el contingut d'aquests elements és text, pot quedar partit en diverses línies.

Ocupen **només l'espai que necessiten** i generalment no podem assignar-los una mida concreta, però sí que podem definir-hi "**padding**" i "**margin**".



Existeixen dues propietats principals a CSS que ens permeten controlar el posicionament dels elements: **display** i **position**.

## display

La propietat “**display**” ens permet controlar **com es visualitzen els elements en relació als elements precedents i en funció del seu contingut**. D'aquesta manera podem modificar el comportament per defecte dels elements.

Contemplem els següents fragments d'HTML i CSS. A continuació, veurem quin és el resultat d'aplicar diferents valors a la propietat “**display**” de l'element amb la classe “**test-span**”.

```
...  
<p>  
    <span>Fragment 1</span>  
    <span class="test-span">  
        Fragment 2  
    </span>  
    <span>Fragment 3</span>  
</p>  
...
```

```
p {  
    background-color: grey;  
    color: white;  
}  
  
span { background-color: black; }  
  
.test-span {  
    background-color: red;  
    padding: 10px;  
    height: 50px;  
}
```

## **inline**

És el comportament per defecte dels elements en línia. L'element es comporta com un fragment de text i, excepte per als mitjans (img, etc.), l'amplada i l'altura sempre són automàtiques.

```
.test-span {  
  display: inline;  
}
```



Fragment 1

Fragment 2

Fragment 3





## **block**

És el comportament per defecte dels elements en bloc. L'element ocupa per defecte tota l'amplada del contenidor i no permet mai que altres elements se situïn al seu costat.

```
.test-span {  
  display: block;  
}
```



Fragment 1

Fragment 2

Fragment 3

## **inline-block**

L'element se situa en línia, però podem assignar-li una amplada i una altura específiques. No quedarà mai partit en dues línies.

```
.test-span {  
  display: inline-block;  
}
```



<b>none</b>	L'element no es mostra. És com si no hi fos.
-------------	--

```
.test-span {  
  display: none;  
}
```



Fragment 1 | Fragment 3

## Text per a lectors de pantalla

Si volem **amagar un text** destinat exclusivament als **lectors de pantalla**, no ho farem amb “**display: none;**”, ja que els lectors de pantalla ignoren els elements amagats d’aquesta manera.

Podem afegir una **classe** a aquells elements que vulguem **amagar visualment** però que siguin **accessibles per als lectors de pantalla** i aplicar-los les següents propietats:

```
.screen-reader-text {  
    width: 1px;  
    height: 1px;  
    padding: 0;  
    margin: -1px;  
    border: 0;  
    overflow: hidden;  
    position: absolute;  
    clip: rect(1px, 1px, 1px, 1px);  
    clip-path: inset(50%);  
    word-wrap: normal !important;  
}
```

## display: flex i grid

Per a la propietat “**display**” també existeixen els valors “**flex**” (o “inline-flex”) i “**grid**” (o “inline-grid”).

Aquests valors afecten la manera **com es distribueix el contingut de l'element** i ens permeten utilitzar una sèrie de propietats addicionals per controlar de manera molt versàtil aquesta distribució.

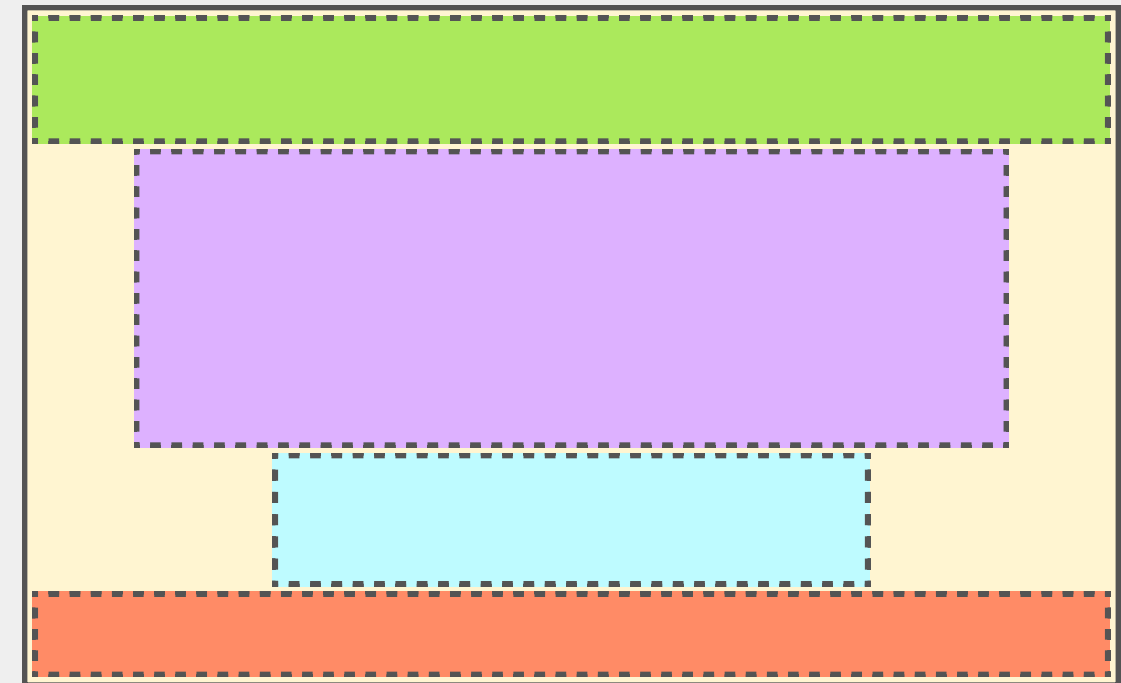
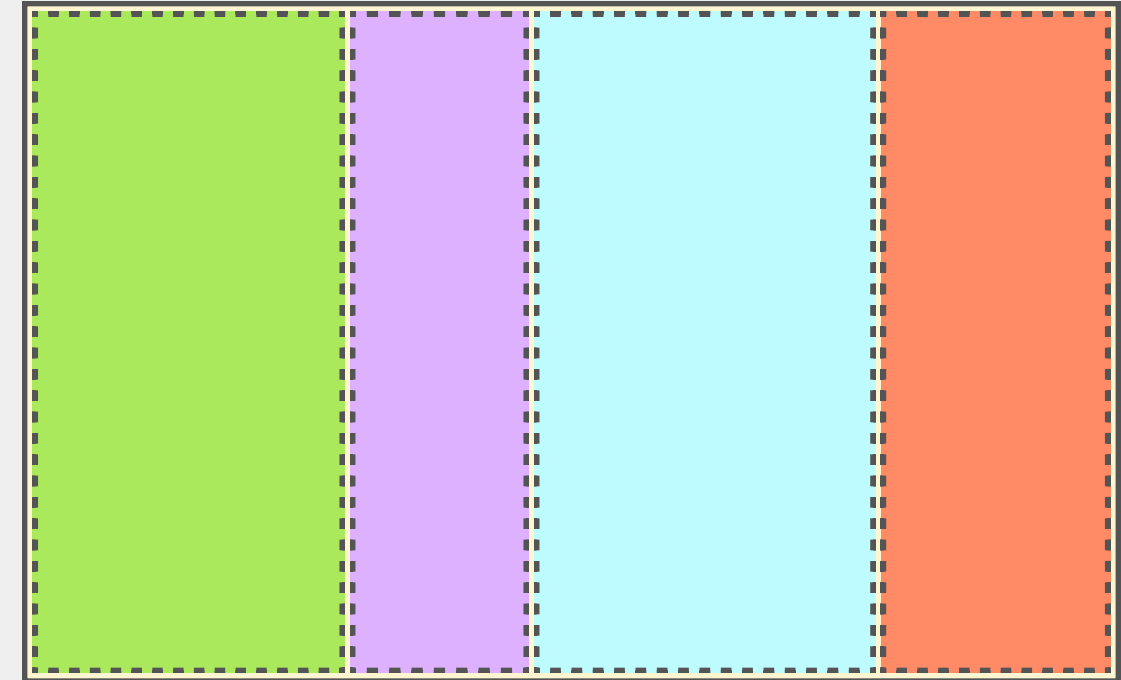
## display: flex (o inline-flex)

Ens permet distribuir elements en una única direcció (fila o columna).

Més informació:

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

**La propietat *display: flex* no s'aplica als elements que volem distribuir, sinó al seu contenidor.**



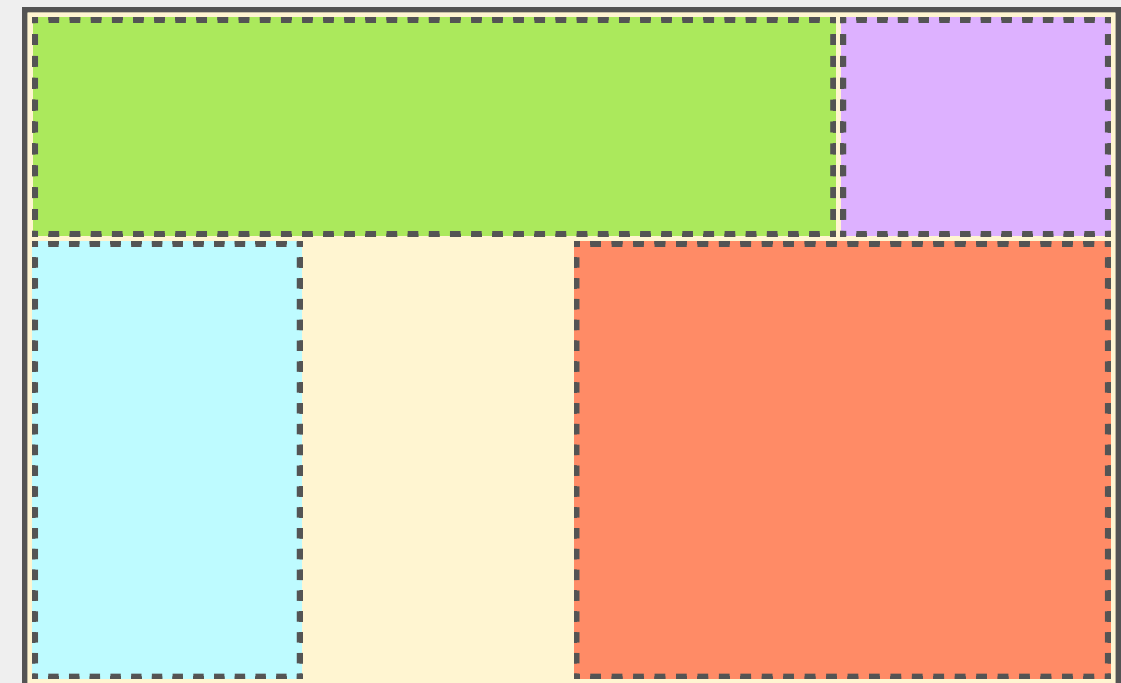
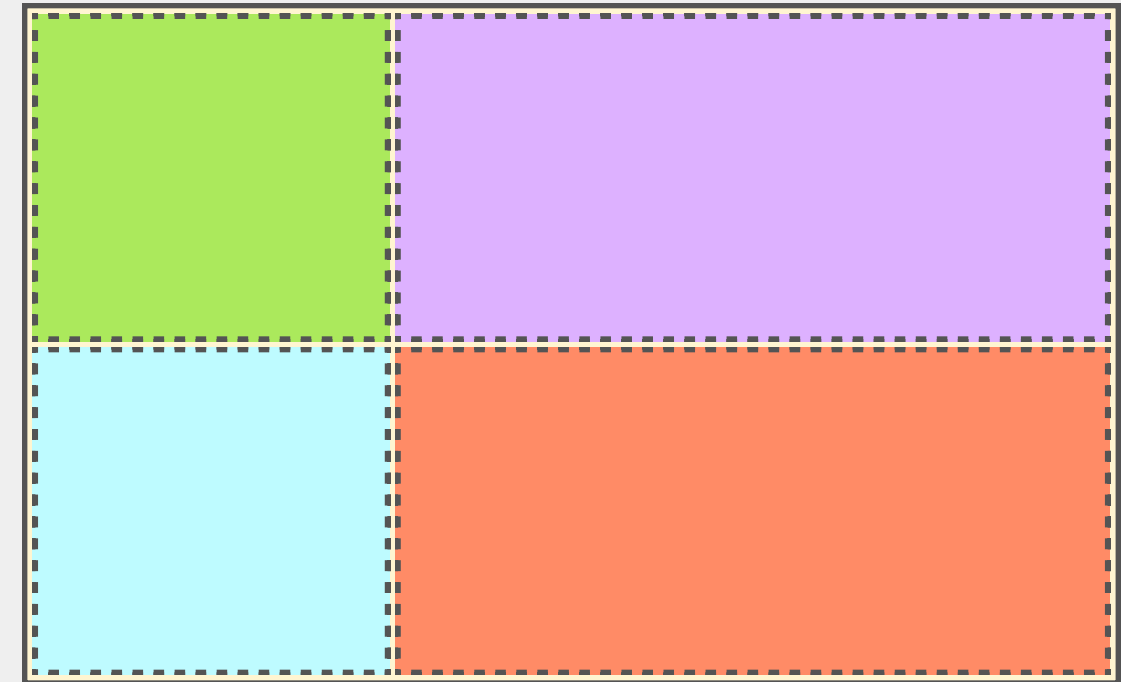
## display: grid (o inline-grid)

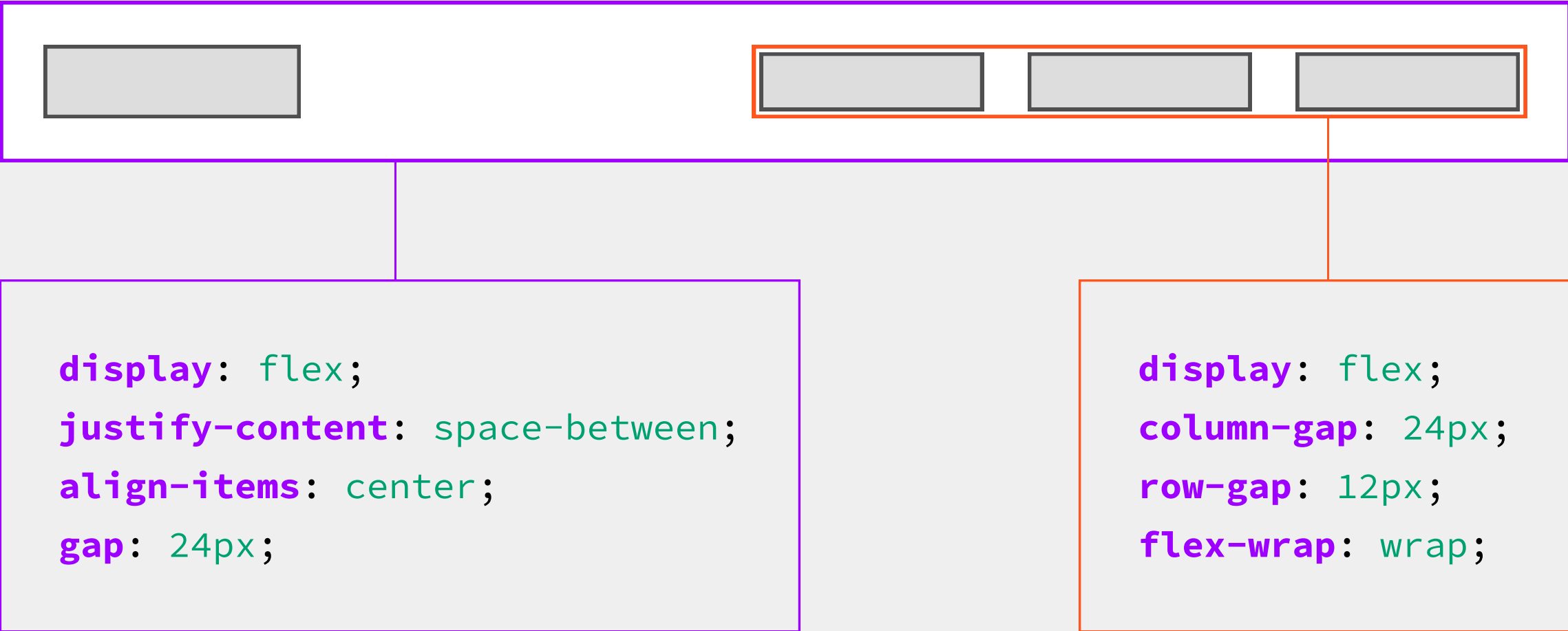
Ens permet distribuir elements en una graella personalitzada.

Més informació:

<https://css-tricks.com/snippets/css/complete-guide-grid/>

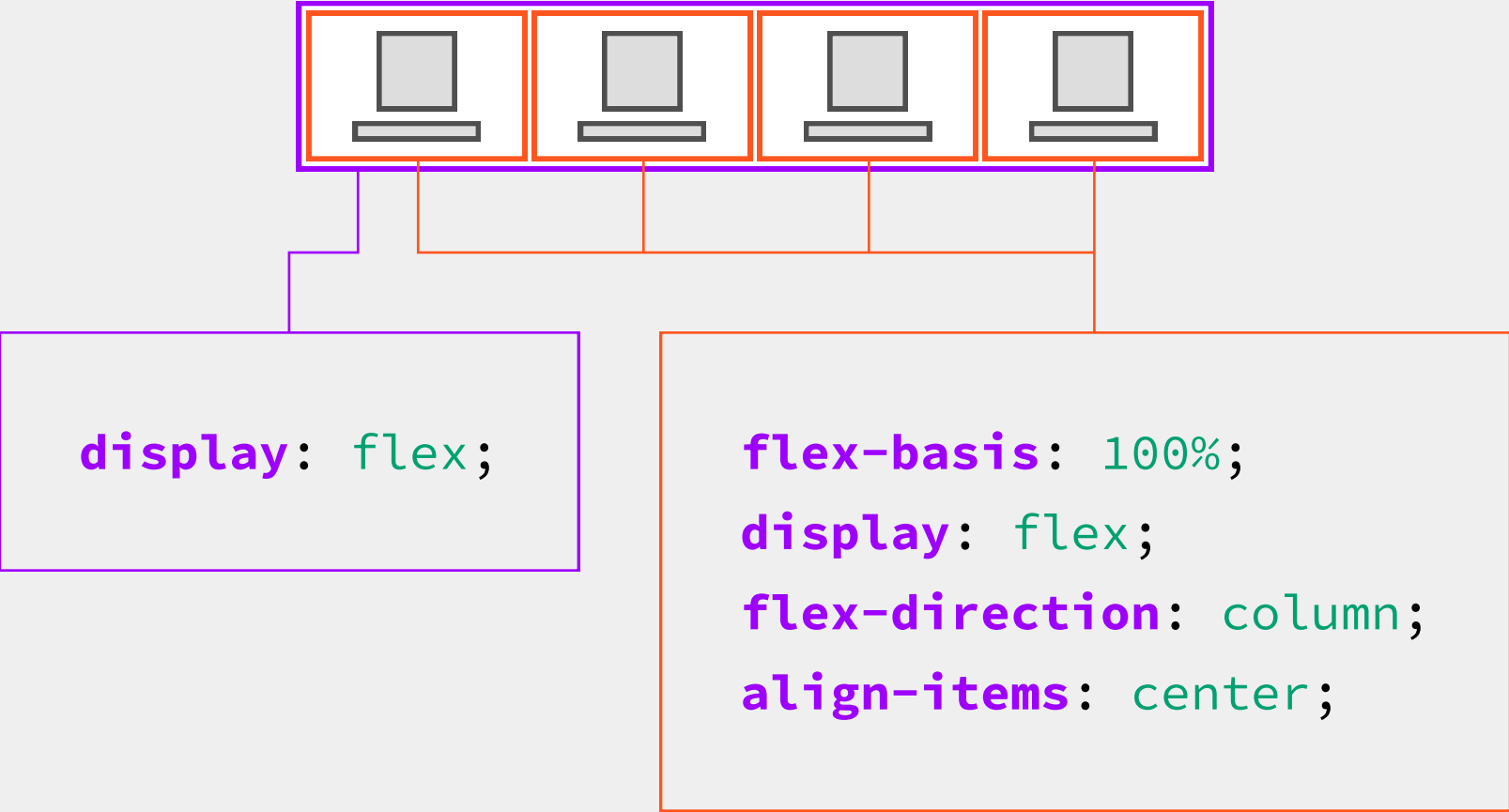
**La propietat *display: grid* no s'aplica als elements que volem distribuir, sinó al seu contenidor.**



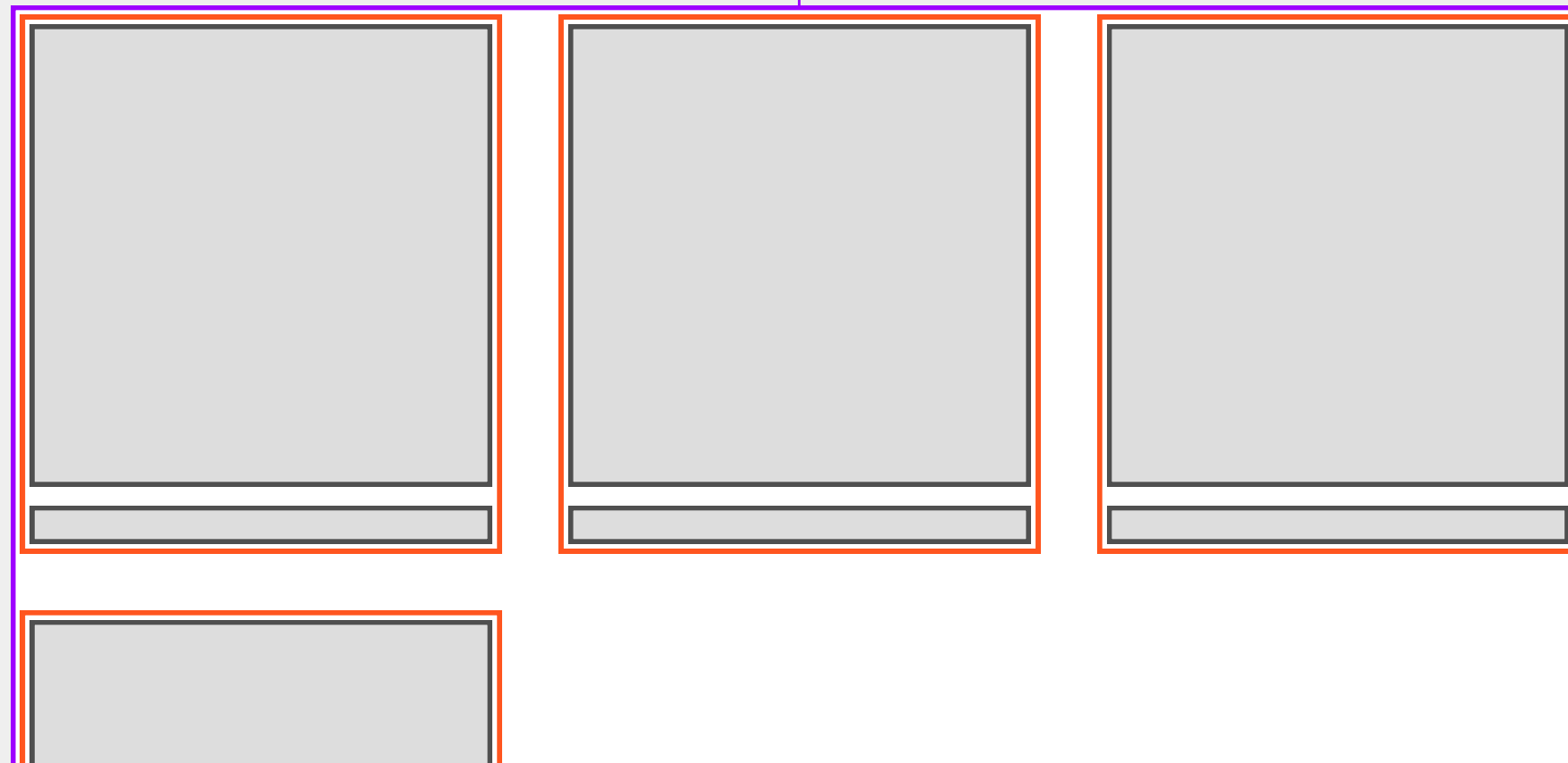




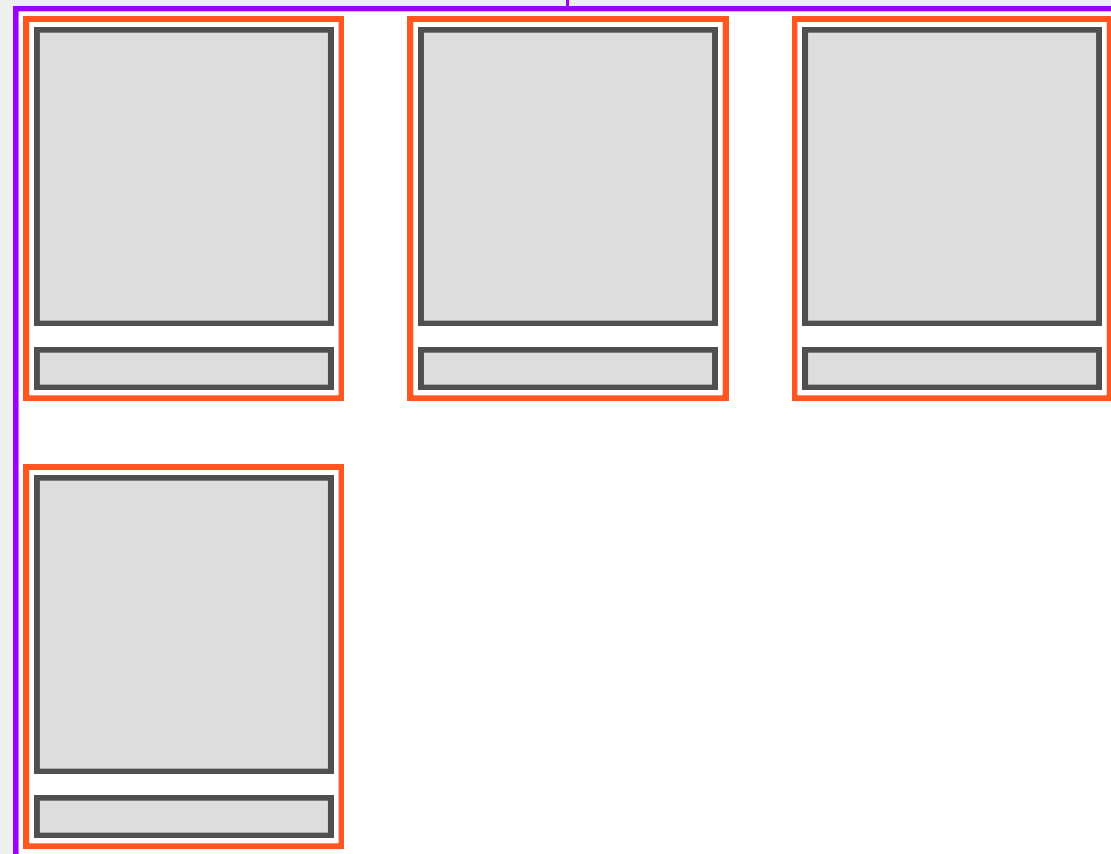




```
display: grid;  
grid-template-columns: repeat(3, 1fr);  
gap: 24px;
```



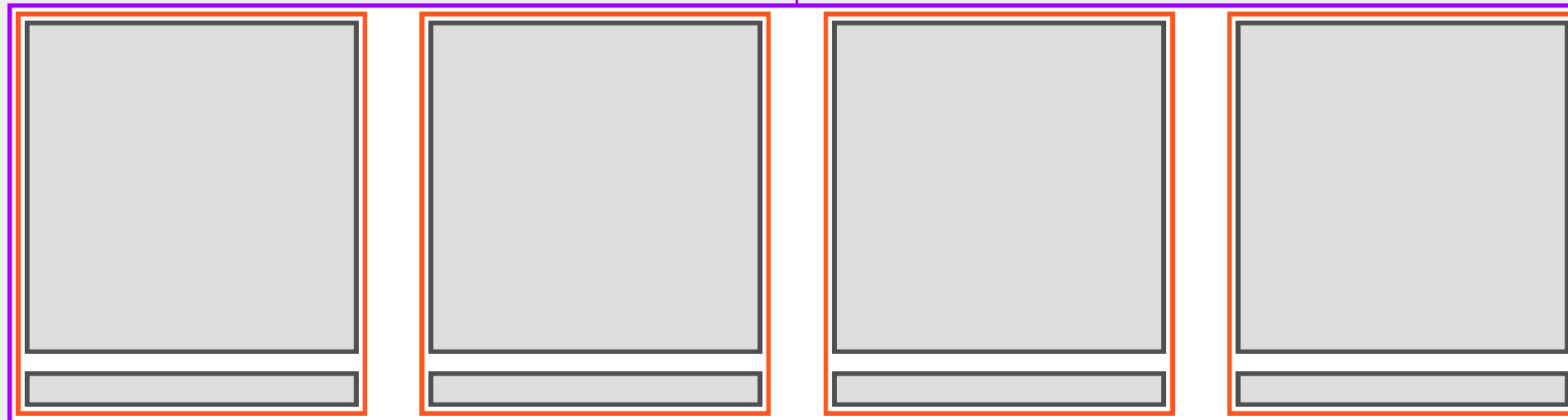
```
display: grid;  
grid-template-columns: repeat(3, 1fr);  
gap: 24px;
```



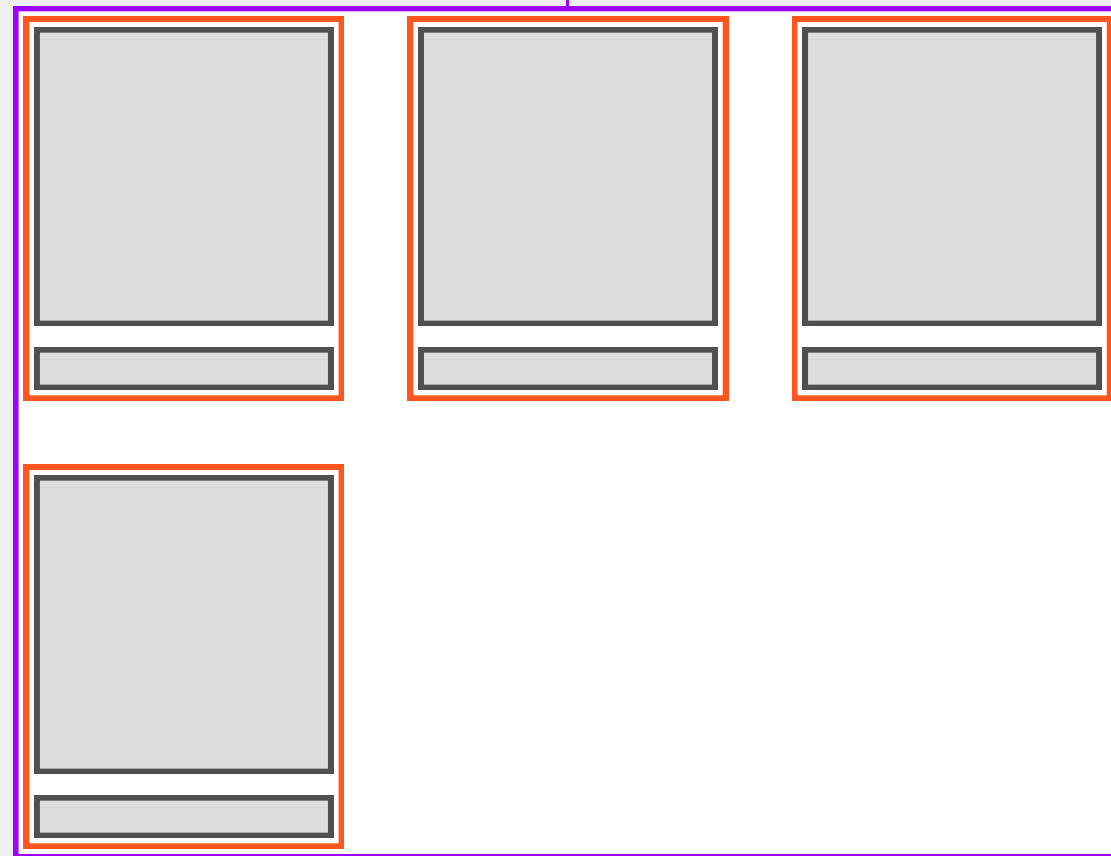
```
display: grid;  
grid-template-columns: repeat(3, 1fr);  
gap: 24px;
```



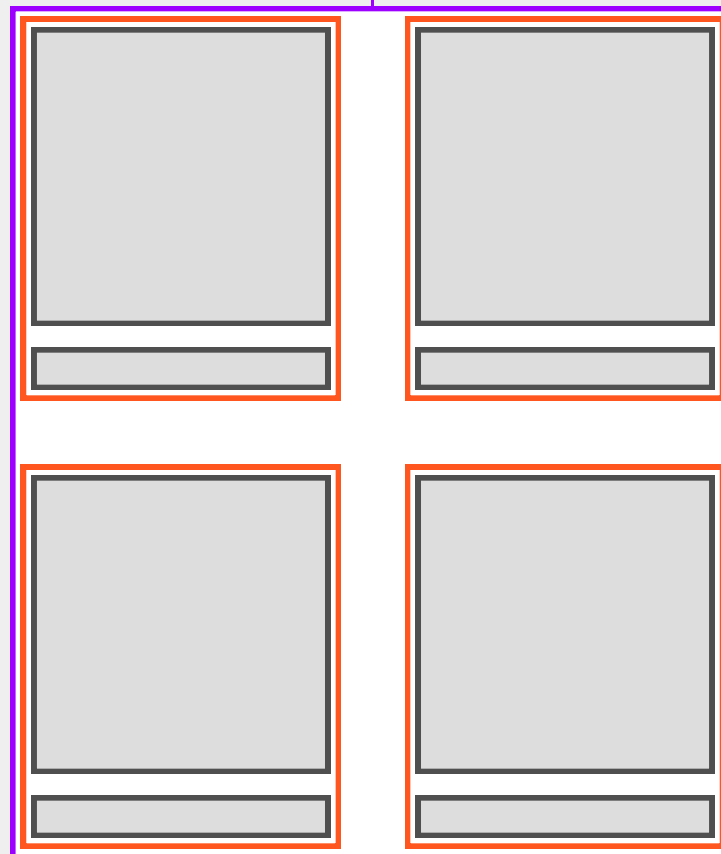
```
display: grid;  
grid-template-columns: repeat(auto-fill, minmax(120px, 1fr));  
gap: 24px;
```



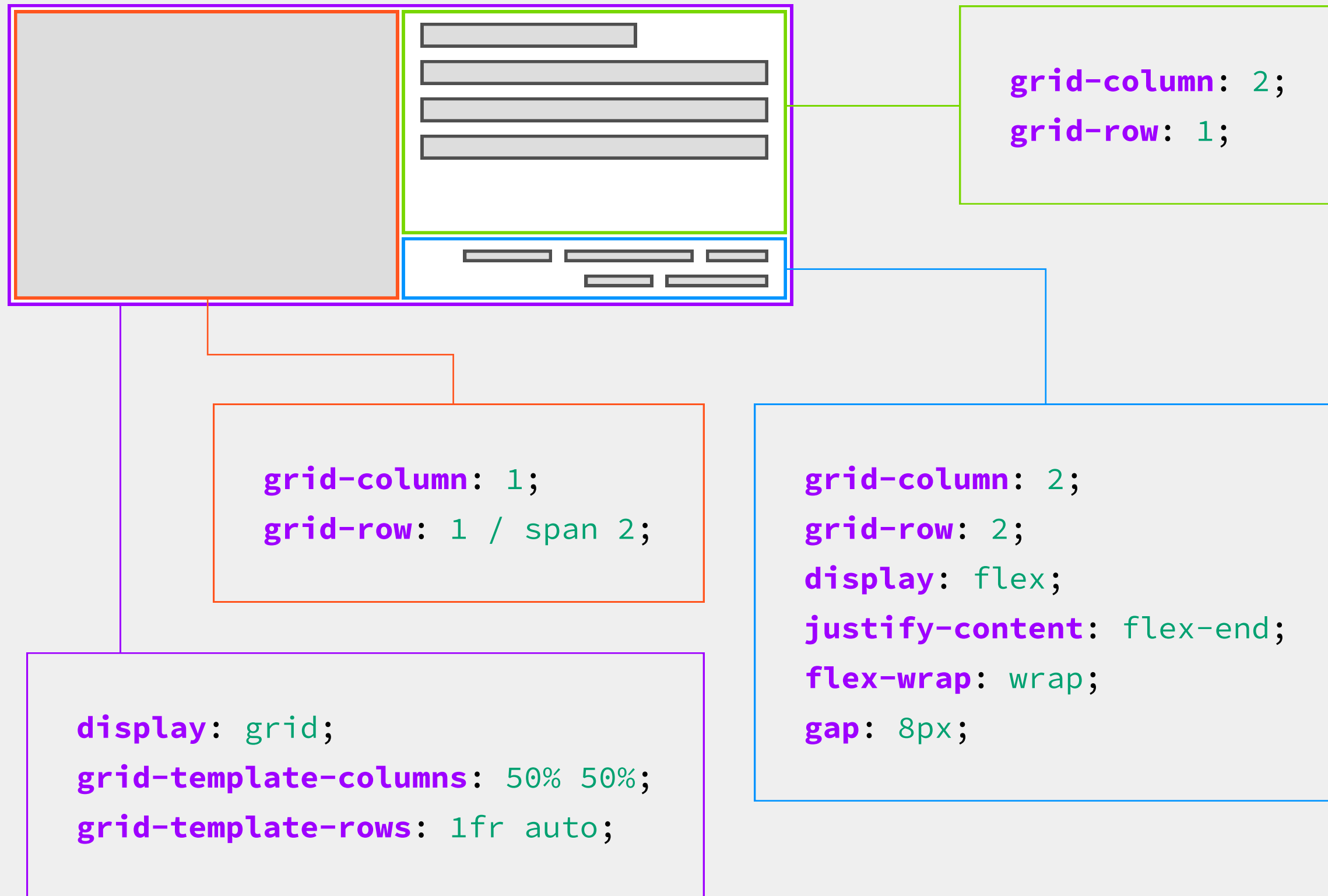
```
display: grid;  
grid-template-columns: repeat(auto-fill, minmax(120px, 1fr));  
gap: 24px;
```



```
display: grid;  
grid-template-columns: repeat(auto-fill, minmax(120px, 1fr));  
gap: 24px;
```







## position

la propietat “**position**” ens permet modificar respecte a quina **àrea de referència** es posicionen els elements.

Juntament amb “position” podem fer servir les propietats “**top**”, “**left**”, “**bottom**” i “**right**” per determinar la **distància entre l'element i les vores de l'àrea de referència**.

Veiem quin és el resultat d'aplicar diferents valors a la propietat “**position**” de l'element amb la classe “**test-p**” del següent exemple.

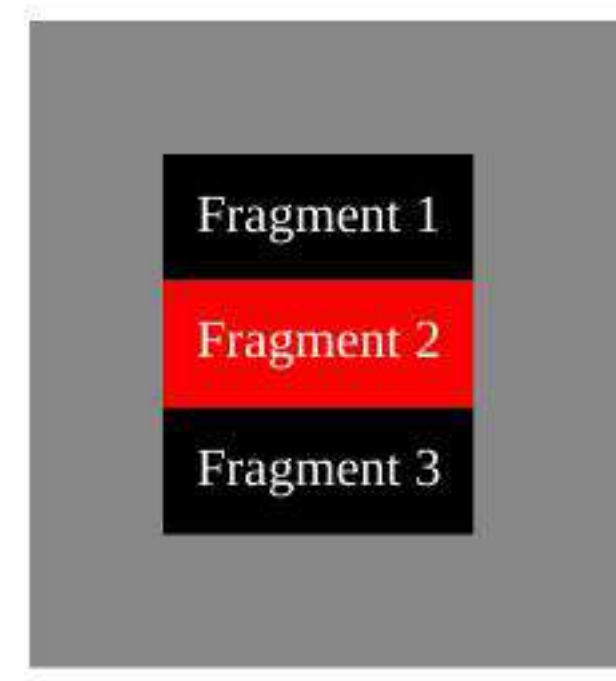
```
...  
<div>  
  <p>Fragment 1</p>  
  <p class="test-p">  
    Fragment 2  
  </p>  
  <p>Fragment 3</p>  
</div>  
...
```

```
div {  
  background-color: grey;  
  padding: 50px;  
}  
  
p {  
  background-color: black;  
  color: white;  
  margin: 0;  
  padding: 10px;  
}  
  
.test-p {  
  background-color: red;  
}
```

## **static**

És el valor per defecte. L'element es posiciona respecte el seu context de la manera habitual. Els elements amb posicionament “estàtic” no poden actuar com a referència de posicionament per als elements que contenen.

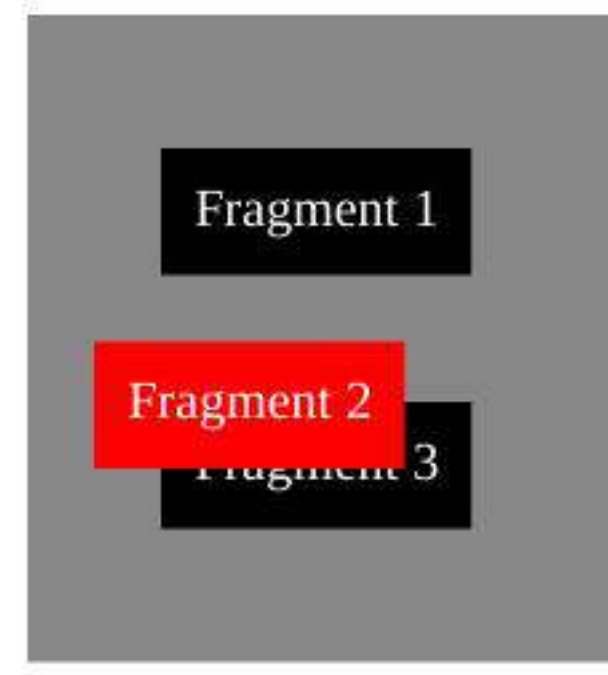
```
.test-p {  
  position: static;  
}
```



## **relative**

Ens permet desplaçar l'element de manera relativa a la seva posició per defecte mitjançant les propietats “top”, “left”, “bottom” i “right”.

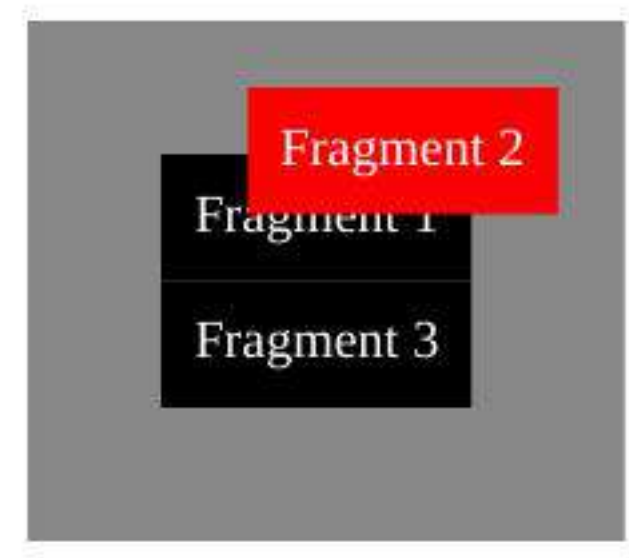
```
.test-p {  
  position: relative;  
  top: 25px;  
  right: 25px;  
}
```



## **absolute**

L'element es posicionarà respecte les vores del contenidor més immediat que no tingui un posicionament “estàtic” i per damunt de la resta de contingut.

```
div {  
    position: relative;  
}  
.test-p {  
    position: absolute;  
    top: 25px;  
    right: 25px;  
}
```



## **fixed**

L'element es posicionarà respecte les vores de la finestra i per damunt de la resta de contingut .

```
.test-p {  
  position: fixed;  
  top: 25px;  
  right: 25px;  
}
```

