



Škola za umjetnost,  
dizajn, grafiku i  
odjeću Zabok



# Novi pristupi CSS rasporedima

02 / MODERNI WEB



# Prisjetimo se!

---

Pozicioniranje pomoću elementa 'inline-block'

Pozicioniranje pomoću svojstva 'float' – plutajući model (eng. *floating model*)

Pozicioniranje pomoću svojstva 'position'

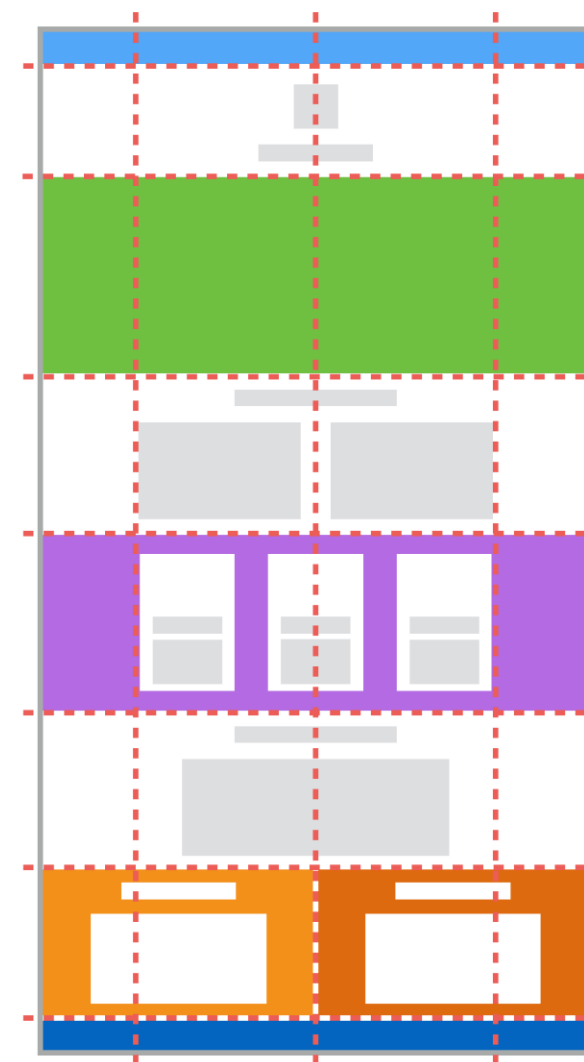
Model fleksibilnih kutija (eng. *flexbox*)

**Grid model**

# CSS grid

## Dosta je trikova!

Od početka weba, web dizajn je 'strgan'. Tijekom vremena samo su se mijenjali načini 'trganja'. CSS nikad nije imao mogućnost pozicioniranja grupa elemenata na stranici horizontalno ili vertikalno, no zajednica je tijekom godina kreirala mnoštvo načina kako to postići – od tablica, preko plutanja i 'inline-block' elemenata pa sve do fleksibilnih kutija. Sve to su zapravo bili trikovi za strukturiranje stranice.



# Što je CSS grid?

---

CSS grid je dvodimenzionalni pristup CSS rasporedima u obliku mreže pomoću kojeg je moguće pozicionirati grupe elemenata bilo gdje na stranici – bilo to horizontalno ili vertikalno.

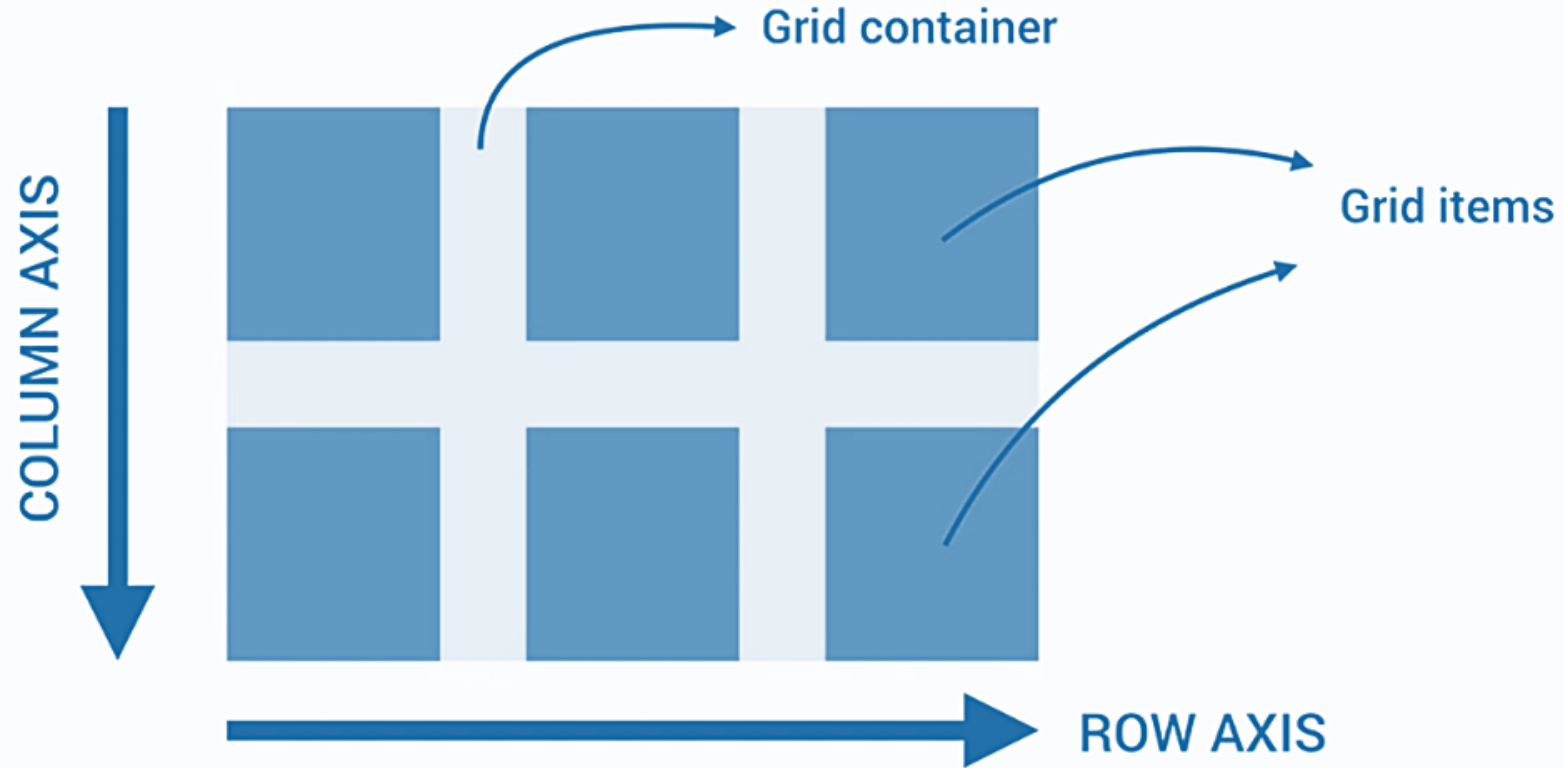
- Elemente grida moguće je proizvoljno premještati bilo gdje po mreži, međusobno ih preklapati ili spajati, u cilju dobivanja nove strukture za sadržaj koji se u njemu nalazi.

Osnovni dijelovi grida su **grid container** i **grid item**.

- Grid container je HTML element koji definira grid te sadrži grid elemente.
- Grid item su HTML elementi koju su neposredna djeca grid containera te strukturalno i semantički definiraju elemente stranice (npr. zaglavlje, sadržaj, podnožje).
- Kada se u container postavi deklaracija `display: grid;` taj container postaje grid container, a elementi prve razine koji se nalaze u njemu postaju grid elementi.



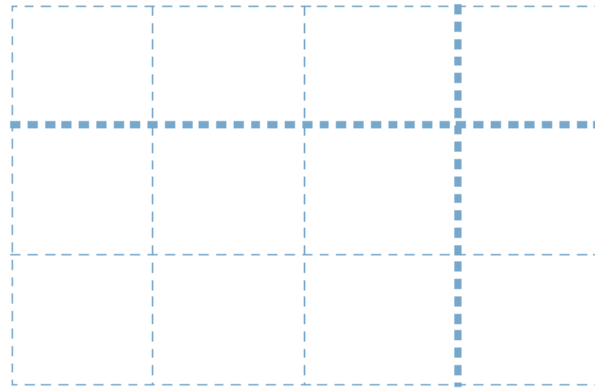
# CSS GRID TERMINOLOGY



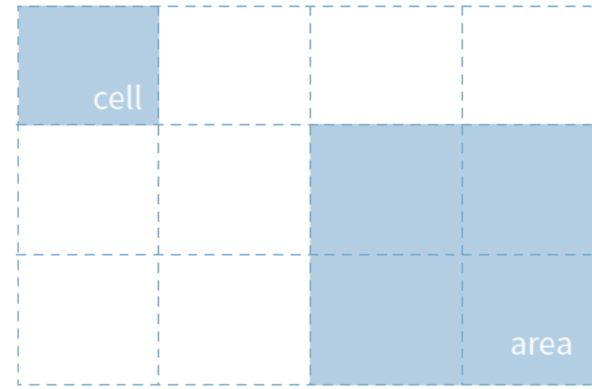
```
display: grid
```

```
(display: grid-inline)
```

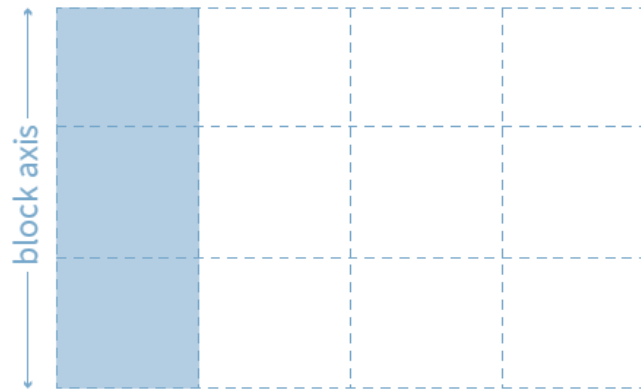
# CSS GRID TERMINOLOGY



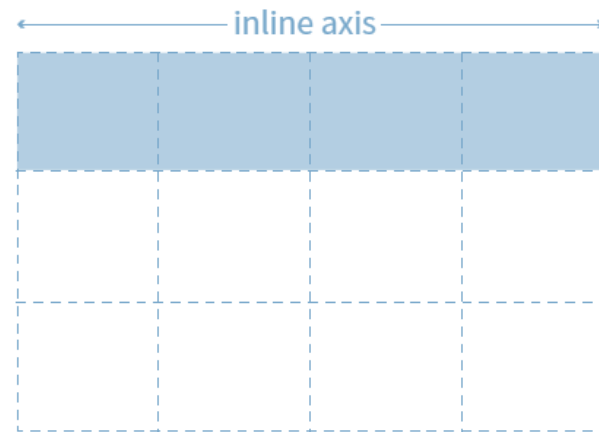
grid lines



grid cell and grid area



grid track (column)



grid track (row)

# CSS GRID PROPERTIES OVERVIEW

## CONTAINER

1 `grid-template-rows`  
`grid-template-columns`  
`grid-template-areas` } `grid-template`

2 `grid-row-gap`  
`grid-column-gap` } `grid-gap`

`justify-items`

3 `align-items`  
`justify-content`  
`align-content`

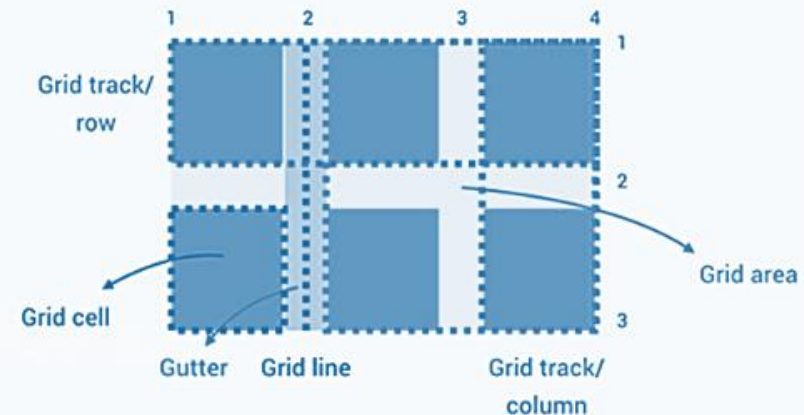
4 `grid-auto-rows`  
`grid-auto-columns`  
`grid-auto-flow`

## ITEM

1 `grid-row-start`  
`grid-row-end`  
`grid-column-start`  
`grid-column-end` } `grid-row`  
} `grid-column` } `grid-area`

2 `justify-self`  
`align-self`















3 `order`



# CSS grid – pregled svojstava

GRID

container

<b>display</b> Establishes a new grid formatting context for children.	<b>grid-template</b> Defines the rows and columns of the grid.	<b>grid-gap</b> Specifies the size of column and row gutters.	<b>justify-items</b> Aligns content in a grid item along the row axis.
 <code>display: grid;</code>	 <code>grid-template-columns: 12px 12px 12px;</code> <code>grid-template-rows: 12px 12px 12px;</code>	 <code>grid-row-gap: 1px;</code> <code>grid-column-gap: 9px;</code>	 <code>justify-items: start;</code>
 <code>display: inline-grid;</code>	 <code>grid-template-columns: repeat(3, 12px);</code> <code>grid-template-rows: repeat(3, auto);</code>	 <code>grid-gap: 1px 9px;</code>	 <code>justify-items: end;</code>
 <code>display: subgrid;</code>	 <code>grid-template-columns: 8px auto 8px;</code> <code>grid-template-rows: 8px auto 12px;</code>	 <code>grid-gap: 6px;</code>	 <code>justify-items: center;</code>
	 <code>grid-template-columns: 22% 22% auto;</code> <code>grid-template-rows: 22% auto 22%;</code>		 <code>justify-items: stretch; (default)</code>
<b>align-items</b> Aligns content in a grid item along the column axis.	<b>justify-content</b> Justifies all grid content on row axis when total grid size is smaller than container.	<b>align-content</b> Justifies all grid content on column axis when total grid size is smaller than container.	<b>grid-auto-flow</b> Algorithm for automatically placing grid items that aren't explicitly placed.





# fr

## Frakcija – nova mjerna jedinica

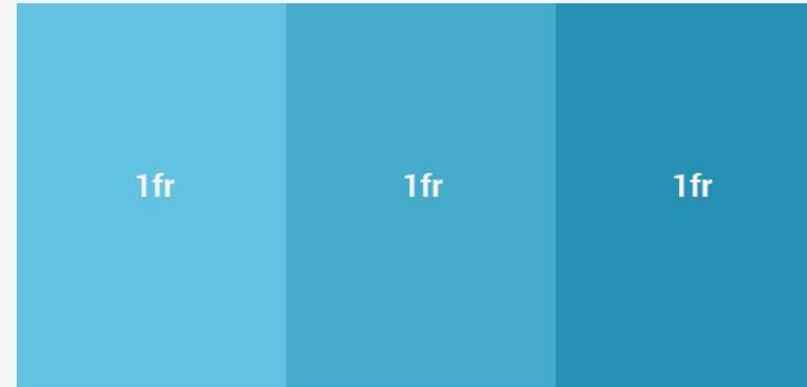
Za potrebe grida stvorena je i nova mjerna jedinica fr (frakcija ili djelić) U slučaju da želimo definirati grid od tri stupca, jednostavno možemo odrediti svojstvo `grid-template-columns: 1fr 1fr 1fr`; ili `grid-template-columns: repeat(3, 1fr)`; gdje svaki stupac zauzima jedan djelić retka (u konkretnom primjeru oko 33% širine retka)

### grid-template-columns: 1fr 1fr 1fr

33% kompletne širine grida

33% kompletne širine grida

33% kompletne širine grida

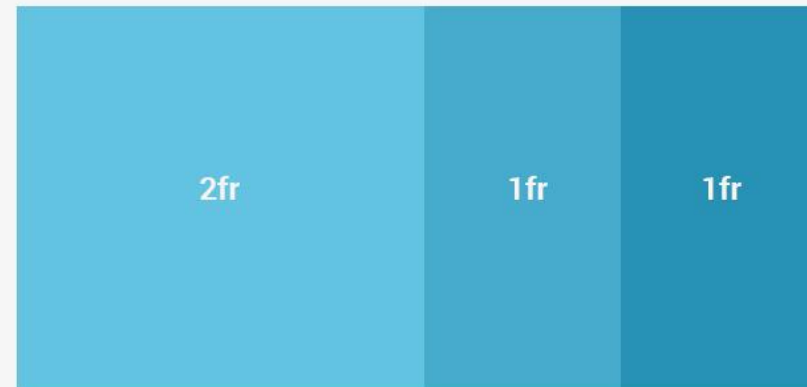


### grid-template-columns: 2fr 1fr 1fr

50% kompletne širine grida

25% kompletne širine grida

25% kompletne širine grida



# Ponovimo!

The screenshot shows a web browser window with the URL `cssgridgarden.com`. The page title is "GRID GARDEN" and it indicates "Level 1 of 28".

Welcome to Grid Garden, where you write CSS code to grow your carrot garden! Water only the areas that have carrots by using the `grid-column-start` property.

For example, `grid-column-start: 3;` will water the area starting at the 3rd vertical grid line, which is another way of saying the 3rd vertical border from the left in the grid.

```
1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #water {
8   |
9 }
10
11
12
13
14
```

Next

Grid Garden is created by [Codepij](#) • [GitHub](#) • [Twitter](#) • [English](#)

Want to learn CSS flexbox? Play [Flexbox Froggy](#).

The grid garden is a 5x5 grid. The top-left cell contains a blue water icon. The second cell in the second row contains two carrot icons.

