Grid

È un moderno **sistema di layout** basato su griglia, con righe e colonne che permette di disporre gli elementi all’interno di un contenitore.

# Flexbox vs Grid

|  |  |
| --- | --- |
| **Dimensione del layout** | |
| Grid è progettato per layout **bidimensionali**, consentendo di posizionare gli elementi sia in righe che in colonne.  Questo lo rende più adatto per layout complessi che richiedono una disposizione in entrambe le dimensioni. | Flexbox progettato per gestire layout **unidimensionali**, sia in orizzontale che in verticale. È utile quando si desidera allineare e distribuire spazio tra elementi in una singola direzione. |
| **Spazio vs Contenuto** | |
| Grid Si **concentra sul layout stesso**.  Si definisce una griglia con righe e colonne e poi si posizionano gli elementi all’interno di questa griglia, indipendentemente dalle dimensioni del contenuto | Flexbox Si concentra sul **contenuto degli elementi**.  Gli elementi flessibili si adattano in base alle dimensioni del contenitore e al contenuto interno, rendendo più semplice l’allineamento e la distribuzione dello spazio |
| **Allineamento** | |
| Grid Consente di posizionare gli elementi in **celle specifiche** e di controllare il loro allineamento sia all’interno delle celle che rispetto alla griglia stessa | Flexbox L’allineamento può essere facilmente controllato lungo l’asse principale e l’asse trasversale, utilizzando proprietà come justify-content e align-items |

# Grid container and items



# grid-template-rows / grid-template-columns

Queste due proprietà sono utilizzate per definire la struttura delle righe e delle colonne di una griglia:

### grid-template-rows

**definisce l’altezza delle righe** nella griglia.  
L’esempio di sotto mostra due righe:



### grid-template-columns

**Definisce la larghezza delle colonne** nella griglia.  
L’esempio di sotto mostra due colonne:



# Immagine che contiene schermata, Rettangolo, quadrato, Policromia Descrizione generata automaticamenteGrid lines

**Le linee che separano la griglia in righe e colonne** funzionano come quelle della carta millimetrata, facilitando la misurazione e il posizionamento degli elementi

# Grid Tracks

**Gli spazi tra le linee** della griglia rappresentano effettivamente le righe e le colonne stesse.  
Questi spazi definiscono la struttura del layout

# Grid Areas

**Gli spazi racchiusi da quattro linee della griglia**, che si estendono su più celle della griglia, rappresentano regioni rettangolari all’interno del layout della griglia.  
Queste aree possono essere utilizzate per raggruppare contenuti

# Grid Cells

**Le singole aree rettangolari** create dall’intersezione di righe e colonne, **conosciute come celle**, possono contenere contenuti.  
Ogni cella è un’unità fondamentale dove puoi posizionare elementi

# **Auto Placement**

È una funzionalità che consente di **posizionare automaticamente gli elementi** all’interno di una griglia senza dover specificare manualmente le celle in cui devono andare

## Posizionamento automatico

Quando un contenitore grid ha un’altezza specificata ma non ha righe o colonne definite, gli elementi della griglia si posizionano automaticamente nel contenitore.

**Gli elementi vengono disposti nel primo spazio disponibile, seguendo l’ordine in cui sono stati aggiunti**

## Determinazione delle righe

Il browser determina automaticamente il numero di righe necessarie in base al contenuto degli elementi e allo spazio disponibile.

Questo significa che, se gli elementi non si adattano all’altezza specificata, il browser regolerà il numero di righe per garantire che tutti gli elementi rientrino all’interno dell’altezza impostata

## Numero di colonne

**Di default, il numero di colonne è uno.**Questo significa che gli elementi verranno disposti in una singola colonna fino a quando non ci sarà abbastanza spazio verticale per aggiungere una nuova riga

## Dimensione degli elementi

Ogni elemento della griglia si adatta in modo da rimanere all’interno dell’altezza impostata, mantenendo gli spazi definiti dalla proprietà gap

# **Gap**

E’ una proprietà che imposta uno **spazio** uniforme **tre le righe e le colonne** all’interno di un contenitore grid.

## Immagine che contiene schermata, Rettangolo, quadrato, design Descrizione generata automaticamente row-gap

specifica lo spazio **verticale** tra le righe della griglia

## column-gap

regola lo spazio **orizzontale** tra le colonne della griglia

# **Grid sizing**

## Unità frazionarie

Consentono di allocare lo spazio disponibile in modo proporzionale.  
**Utilizzando fr, puoi definire colonne o righe che si adattano automaticamente in base allo spazio disponibile nel contenitore.**

Ad esempio, se hai tre colonne definire come 1fr, 2fr, e 1fr, la seconda colonna occuperà il doppio dello spazio rispetto alle altre due

## Min-content

**Funzione** che dimensiona automaticamente gli elementi della griglia in base alla **larghezza o all’altezza minima necessaria per contenere il contenuto.**

Utilizzando min-content, la griglia si adatta al contenuto più piccolo, garantendo che non venga mai ridotto al di sotto di un certo punto

## Auto

Questa parola chiave **dimensiona automaticamente gli elementi in base al loro contenuto o alle dimensioni del contenitore.**

Se un elemento ha una dimensione **auto**, si espanderà per adattarsi al contenuto senza superare le dimensioni del contenitore

## Funzione Repeat

Consente di creare **più colonne** o **righe** **con un modello ripetuto**, semplificando la scrittura del codice.  
Ad esempio:

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente**repeat(3, 100px)** crea 3 colonne di 100px ciascuna, mentre **repeat(4, 1fr)** ne crea 4 che si dividono lo spazio disponibile in modo frazionale

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

# **Posizionamento degli elementi**

Le proprietà **grid-column** e **grid-row** permettono di specificare in quali celle della griglia si trovano gli elementi.  
Si possono definire le righe e le colonne di partenza e di arrivo, permettendo un posizionamento preciso.

Immagine che contiene testo, Carattere, ricevuta, algebra

Descrizione generata automaticamente

## Valore singolo

Quando si utilizza un valore singolo per **grid-row** e **grid-column**, l’elemento inizia alla linea specificata e termina alla linea successiva, occupando quindi la riga o la colonna corrispondente

Immagine che contiene testo, ricevuta, Carattere, schermata

Descrizione generata automaticamente

## Valori negativi

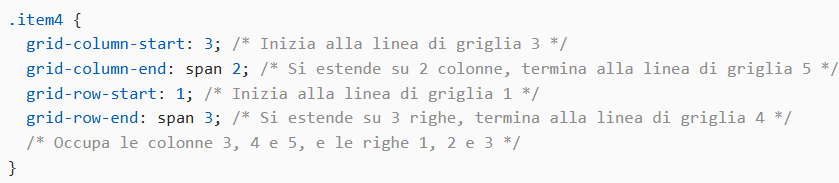
Utilizzare valori negativi **consente di posizionare gli elementi in relazione alla fine della griglia**.

Particolarmente utile quando non si conosce il numero esatto di linee della griglia, permettendo di gestire layout più dinamici

Immagine che contiene testo, ricevuta, Carattere, schermata

Descrizione generata automaticamente

# **Controllo preciso**

Immagine che contiene testo, Carattere, schermata, numero

Descrizione generata automaticamentePer ottenere un controllo più preciso sul posizionamento degli elementi nella griglia, puoi utilizzare le proprietà separate **grid-column-start**, **grid-column-end**, **grid-row-start** e **grid-row-end**. Queste proprietà ti permettono di specificare esattamente dove un elemento deve iniziare e finire.

La proprietà **span** è utilizzata per far espandere un elemento su un numero specifico di righe o colonne, senza dover specificare manualmente il punto finale

# Linee di griglia nominate

Le linee di griglia nominate **consentono di assegnare nomi specifici a determinate linee della griglia**, migliorando la leggibilità del layout e semplificando il posizionamento degli elementi.

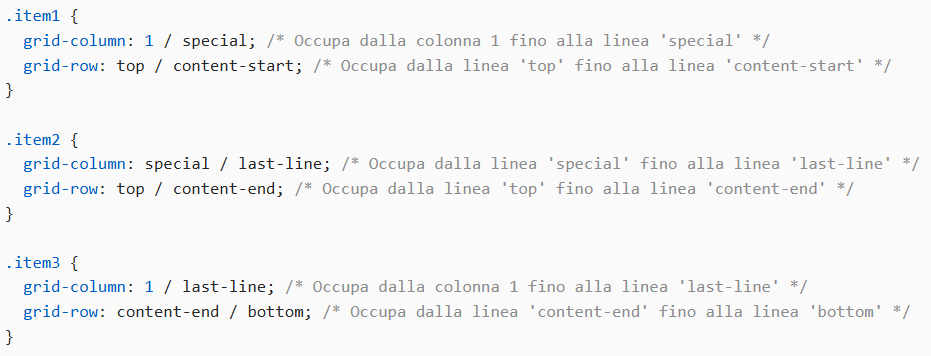
Invece di usare valori numerici per posizionare gli elementi, si possono usare nomi descrittivi, rendendo il codice più intuitivo.

Si possono assegnare nomi alle linee della griglia usando le proprietà **grid-template-columns** e **grid-template row:**

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente

Si possono posizionare gli elementi nella griglia usando i nomi delle linee:



# Keyboard navigation & CSS grid

CSS **grid** è uno strumento per il layout visivo, ma **non modifica la struttura** semantica **del documento HTML**.

Questo significa che, anche se gli elementi possono apparire in un certo ordine visivo sulla pagina, l’ordine in cui sono scritti nel codice rimane invariato.

Quando un utente naviga in una pagina web usando la tastiera, il focus segue l’ordine in cui gli elementi sono presenti nel DOM, non l’ordine in cui appaiono visivamente sulla pagina.

# **Immagine che contiene testo, schermata, design Descrizione generata automaticamenteGrid-template-area**

**È una proprietà che viene utilizzata per definire la disposizione delle aree della griglia.**

Permette di disegnare una mappa o un modello che assegna nomi a specifiche aree della griglia, e quindi puoi associare diversi elementi HTML a quelle aree tramite la proprietà **grid-area**.

# **Grid alignment**

## Align-items

Proprietà che serve per **allineare gli elementi all’interno delle celle della griglia lungo l’asse verticale**.  
Valori:

* **start**: allinea gli elementi all'inizio dell'area della griglia, quindi in alto.
* **end**: allinea gli elementi alla fine dell'area della griglia, quindi in basso.
* **center**: centra verticalmente gli elementi all'interno della griglia.
* **stretch** (valore predefinito): gli elementi vengono estesi per riempire tutta l'altezza della cella.

## Immagine che contiene Rettangolo, quadrato, schermata Descrizione generata automaticamenteJustify-items

Simile ad align-items ma **si applica lungo l’asse orizzontale.**

I valori sono gli stessi

## Place-items

*Place-items: align-items justify-items*

**Shorthand** che combina insieme **align-items** e **justify-items**.  
Se si fornisce un solo valore, verrà applicato a entrambi gli assi

## Align-content

È una proprietà che si applica lungo l’asse verticale e controlla **l’allineamento delle righe** della griglia all’interno del contenitore.

Viene utilizzata solo quando l’altezza totale della griglia è inferiore all’altezza del suo contenitore.  
Se la griglia occupa tutto lo spazio del contenitore, questa proprietà non ha effetto.

Valori:

* **start**: allinea le righe della griglia all'inizio (in alto) del contenitore.
* **end**: allinea le righe alla fine (in basso) del contenitore.
* **center**: centra verticalmente le righe della griglia all'interno del contenitore.
* **stretch** (valore predefinito): allunga le righe per riempire tutto lo spazio verticale disponibile.
* **space-between**: distribuisce le righe della griglia uniformemente, mettendo la prima riga in alto e l'ultima in basso, e distribuendo lo spazio tra le righe.
* **space-around**: distribuisce le righe con spazi uguali intorno a ciascuna riga (spazio sia prima che dopo).
* **space-evenly**: distribuisce le righe della griglia in modo che lo spazio tra le righe, e tra le righe e i bordi del contenitore, sia uguale.

## Justify-content

**Proprietà che si applica lungo l’asse orizzontale e controlla come le colonne della griglia vengono allineate o distribuite all’interno del contenitore**.

Anche questa proprietà funziona solo quando la larghezza totale della griglia è inferiore alla larghezza del contenitore.

I valori sono gli stessi di align-content.

## Immagine che contiene Rettangolo, quadrato, schermata, presa Descrizione generata automaticamentePlace-content

*Place-content: align-content justify-content*

**Shorthand** che combina **align-content** e **justify-content**, permettendo di controllare l’allineamento della griglia sia sull’asse verticale che orizzontale in una sola dichiarazione

# Allineamento dei **singoli elementi** della griglia

## Align-self

**Permette di allineare un singolo elemento della griglia lungo l’asse verticale** all’interno della sua cella.  
Sovrascrive il comportamento di align-items

* **start**: allinea l'elemento all'inizio della cella, cioè in alto.
* **end**: allinea l'elemento alla fine della cella, cioè in basso.
* **center**: centra l'elemento verticalmente nella cella.
* **stretch** (valore predefinito): estende l'elemento per riempire tutta l'altezza della cella.

## Justify-self

**Controlla l’allineamento orizzontale di un singolo elemento** della griglia all’interno della sua cella.  
Sovrascrive il comportamento di justify-items

* **start**: allinea l'elemento all'inizio della cella (a sinistra).
* **end**: allinea l'elemento alla fine della cella (a destra).
* **center**: centra l'elemento orizzontalmente nella cella.
* **stretch** (valore predefinito): estende l'elemento per riempire tutta la larghezza della cella.

## Place-self

*Place-self: align-self justify-self*

**Shorthand** che combina insieme **align-self** e **justify-self**, permettendo di impostare simultaneamente l’allineamento sia verticale che orizzontale per un singolo elemento

Se si fornisce un solo valore, viene applicato a entrambi gli assi

# **Auto rows** and **columns**

Queste due **proprietà** sono **usate per definire la dimensione di righe e colonne implicite**, cioè quelle che vengono create automaticamente dal browser quando il layout richiede più righe o colonne rispetto a quelle definite esplicitamente

## Grid-auto-rows

**Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamenteControlla l’altezza delle righe create implicitamente**.

Quando un elemento della griglia viene posizionato in un’area che richiede una riga non definita, il browser genera nuove righe, e questa proprietà ne controlla l’altezza

Valori principali:

* **<length>:** Imposta un'altezza fissa per le righe implicite (es. 100px).
* **<percentage>:** Altezza relativa rispetto all'altezza del contenitore della griglia (es. 20%).
* **auto**: L'altezza della riga sarà basata sul contenuto, espandendosi quanto necessario.
* **min-content**: L'altezza sarà la minima necessaria per far entrare il contenuto.
* **max-content**: L'altezza sarà la massima necessaria per far entrare tutto il contenuto.
* **minmax(min, max):** Imposta un intervallo, con una dimensione minima e massima per le righe implicite.
* **<flex> (es. 1fr):** Definisce un'altezza flessibile che può crescere o ridursi in base allo spazio disponibile.

## Grid-auto-columns

**Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamenteFunziona come grid-auto-rows ma per le colonne**.

**Controlla la larghezza delle colonne create implicitamente** quando ci sono più elementi della griglia rispetto alle colonne definite implicitamente

Valori principali:

* **<length>:** Imposta una larghezza fissa per le colonne implicite (es. 100px).
* **<percentage>:** Larghezza relativa rispetto alla larghezza del contenitore della griglia (es. 20%).
* **auto**: La larghezza della colonna sarà basata sul contenuto, espandendosi quanto necessario.
* **min-content:** La larghezza sarà la minima necessaria per far entrare il contenuto.
* **max-content**: La larghezza sarà la massima necessaria per far entrare tutto il contenuto.
* **minmax(min, max):** Definisce un intervallo, con una dimensione minima e massima per le colonne implicite.
* **<flex> (es. 1fr):** Definisce una larghezza flessibile che può crescere o ridursi in base allo spazio disponibile.

# Grid-auto-flow

**Proprietà che controlla il modo in cui gli elementi vengono disposti automaticamente nella griglia quando non sono specificate le linee della griglia esplicite** o quando gli elementi non sono collocati manualmente con le proprietà come **grid-row** o **grid-column**

Valori:

* **Row** (default): gli elementi vengono inseriti riempiendo prima le righe, quindi quando una riga è piena passa alla riga successiva
* **Column:** gli elementivengono inseriti riempiendo prima le colonne, quindi quando una colonna è piena, passa alla colonna successiva.  
  Questo comporta che gli elementi siano disposti dall’alto verso il basso, riempiendo prima le colonne
* **Dense:** gli elementi vengono disposti in modo più denso, cercando di riempire eventuali spazi vuoti che si creano nella griglia.  
  Se un elemento non si adatta perfettamente allo spazio disponibile, il browser tornerà indietro e proverà a inserirlo in uno spazio vuoto lasciato da un elemento più grande
* **Row dense:** gli elementi vengono posizionati riempiendo le righe come con row, ma con il comportamento denso che cerca di riempire gli spazi vuoti tra gli elementi in modo efficiente
* **Column dense:** gli elementi vengono posizionati riempiendo le colonne come con column, ma con il comportamento denso per riempire gli spazi vuoti

# **Grid** (shorthand)

*grid: [<grid-template-rows> / <grid-template-columns>] [<grid-template-areas>] [<grid-auto-rows>] [<grid-auto-columns>] [<grid-auto-flow>];*

Permette di definire in modo conciso le impostazioni della griglia, combinando in un’unica dichiarazione diverse proprietà che controllano la struttura del layout della griglia

* **grid-template-rows**: definisce l'altezza delle righe della griglia.
* **grid-template-columns**: definisce la larghezza delle colonne della griglia.
* **grid-template-areas**: definisce le aree della griglia, assegnando nomi agli spazi della griglia.
* **grid-auto-rows**: controlla l'altezza delle righe implicite (create automaticamente dal browser).
* **grid-auto-columns**: controlla la larghezza delle colonne implicite.
* **grid-auto-flow**: controlla l'ordine in cui gli elementi vengono disposti nella griglia quando non ci sono posizionamenti espliciti.