

# CSS

## CSS CheatSheet in Hinglish

Bhai, ye CSS properties ka cheat sheet hai, jisme hum sab properties ko Hinglish mein explain karenge. Har property ko ache se samjhaunga taki tumhe asani se samajh aa jaye aur yaad bhi rahe. Memorize karne ke tricks bhi dunga, jaise acronyms ya simple analogies. Har property ke sath related extra properties bataunga jo usse connect hoti hain. Possible values bhi detail mein bataunga: agar fixed values hain to un sab ko list karunga aur har value pe website pe kya behavior dikhega explain karunga. Agar random/user-entered values allowed hain to value types (jaise length, percentage, color) bataunga aur har type ka meaning samjhaunga. Kuch properties missing hain original mein, jaise font-stretch, text-overflow, etc., to unhe bhi add kar raha hoon relevant sections mein. Har section ko original jaise hi structure karunga, with code examples at the end.

## FONT PROPERTIES IN CSS

Font properties se hum text ka look change karte hain, jaise uska face (design), weight (motai), style (tilt). Ye sab milke text ko attractive banate hain, jaise kapde pehnane se insaan ka look badalta hai.

- **Font-Family:** Ye text ke font ka parivar specify karta hai, matlab kis design ka font use karna hai. Samjho jaise ghar mein alag alag members hote hain, waise fonts ke groups. Memorize trick: "Family First" – hamesha pehle font-family set karo. Related extra properties: Font (shorthand jo isko include karta hai). Possible values: User-entered – font names jaise "Arial", "Times New Roman", ya generic families jaise serif (curvy edges wala, jaise newspaper font), sans-serif (clean straight edges, modern look), monospace (equal spacing, code ke liye), cursive (handwriting jaise), fantasy (decorative). Multiple values comma se separate kar sakte ho fallback ke liye (pehle wala na mile to next). Behavior: Agar "serif" dalo to text curvy dikhega, sans-serif se clean; agar custom font na mile to fallback use hoga.
- **Font:** Ye shorthand hai jo font-style, font-variant, font-weight, font-size/line-height, aur font-family ko ek saath set karta hai. Jaise ek package mein sab kuch. Memorize trick: "Style Variant Weight Size Family" – SVWSF acronym

yaad rakho order ke liye. Related extra properties: Sab individual font-\* properties. Possible values: User-entered combination, jaise "italic small-caps bold 20px/1.5 Arial". Order fixed hai, par sab optional except size aur family. Behavior: Sab milke text ko complete look dete hain; agar shorthand use karo to individual override ho sakte hain.

- **Font-Size:** Ye text ka size set karta hai, jaise chhota ya bada. Samjho jaise zoom in-out. Memorize trick: "Size Matters" – hamesha readability ke liye medium size se start. Related extra properties: Line-Height (size ke sath spacing). Possible values: Fixed – xx-small, x-small, small, medium (default), large, x-large, xx-large (browser pe small se large tak gradually badhta size). User-entered – length (jaise 12px – pixel exact size, 1em – parent ke relative, 1rem – root relative), percentage (jaise 150% – parent ke relative). Behavior: Small value se text chhota dikhega, large se bada; percentage use karo responsive design ke liye.
- **Font-Weight:** Ye font ki motai set karta hai, jaise patla ya mota. Memorize trick: "Weight Lifting" – lighter se bolder tak jaise gym weights. Related extra properties: Font (shorthand). Possible values: Fixed – normal (default, 400), bold (700), bolder (relative heavier), lighter (relative lighter). User-entered – numbers 100-900 (100 thinnest, 900 thickest). Behavior: Bold dalo to text bold dikhega (jaise headings), lighter se faint; numbers se precise control, jaise 400 normal reading ke liye.
- **Font-Style:** Ye font ko tilt ya normal set karta hai. Memorize trick: "Style Statement" – italic jaise fashion pose. Related extra properties: Font (shorthand). Possible values: Fixed – normal (straight, default), italic (slanted), oblique (forced slant, italic se similar par artificial). Behavior: Italic se text tilted dikhega (emphasis ke liye), oblique bhi slant par font support pe depend.
- **Font-Variant:** Ye text ko small-caps mein convert karta hai, jisme small letters capital jaise dikhte hain par chhote. Memorize trick: "Variant Virus" – small-caps jaise mutation. Related extra properties: Text-Transform (capitalization related). Possible values: Fixed – normal (default), small-caps. Behavior: Small-caps se text mein lowercase bhi uppercase jaise dikhega par size small, jaise titles mein use.
- **Font-Stretch** (Missing ko add kiya): Ye font ko stretch ya compress karta hai. Memorize trick: "Stretch Like Rubber". Related extra properties: Font-Width

(similar). Possible values: Fixed – normal (default), ultra-condensed, extra-condensed, condensed, semi-condensed, semi-expanded, expanded, extra-expanded, ultra-expanded. Behavior: Condensed se text narrow dikhega (space save), expanded se wide.

Code Example:

```
p {  
  font-family: Times, serif, Arial, Helvetica, sans-serif;  
  font: italic small-caps bold 15px/1.5 Helvetica, sans-serif;  
  font-size: 15px;  
  font-weight: bold;  
  font-style: italic;  
  font-variant: small-caps;  
  font-stretch: expanded; /* Added */  
}
```

## TEXT PROPERTIES IN CSS

Text properties se hum text ko format karte hain, jaise alignment, decoration, spacing. Ye jaise text ko makeup karna.

- **Text-Align:** Ye text ko horizontally align karta hai, jaise left, right. Memorize trick: "Align Like Army" – soldiers ki line jaise. Related extra properties: Text-Align-Last (last line ke liye). Possible values: Fixed – left (default for LTR), right, center, justify (space adjust karke full line). Behavior: Center se text middle mein, justify se newspaper jaise even spacing.
- **Text-Decoration:** Ye text pe underline, overline jaise add karta hai. Memorize trick: "Decorate Like Diwali". Related extra properties: Text-Decoration-Line, Text-Decoration-Color, Text-Decoration-Style. Possible values: Fixed – none (default), underline, overline, line-through (strikethrough). Behavior: Underline se link jaise dikhega, line-through se cancelled text.
- **Letter-Spacing:** Ye characters ke beech space badhata ya ghatata hai. Memorize trick: "Letter Gap Like Traffic". Related extra properties: Word-Spacing. Possible values: User-entered – length (jaise 2px – positive badhaye,

-1px ghataye), normal (default 0). Behavior: Positive se spread out, negative se squeezed.

- **Text-Transform:** Ye text ko uppercase, lowercase convert karta hai. Memorize trick: "Transform Like Transformer". Related extra properties: Font-Variant. Possible values: Fixed – none (default), capitalize (har word capital), uppercase (all caps), lowercase (all small). Behavior: Uppercase se shouting jaise, capitalize se title case.
- **Word-Spacing:** Ye words ke beech space adjust karta hai. Memorize trick: "Word Distance". Related extra properties: Letter-Spacing. Possible values: User-entered – length (jaise 5px), normal (default). Behavior: Badhaoge to words alag alag, ghataoge to close.
- **Text-Indent:** Ye first line ko indent karta hai, jaise paragraph start. Memorize trick: "Indent Like Invoice". Related extra properties: Padding-Left. Possible values: User-entered – length (jaise 20px), percentage (50% container ke). Behavior: Positive se right shift, negative se left overhang.
- **Line-Height:** Ye lines ke beech height set karta hai. Memorize trick: "Height of Lines Like Stairs". Related extra properties: Font-Size. Possible values: Fixed – normal (default ~1.2). User-entered – number (1.5 – multiplier), length (20px), percentage (150%). Behavior: Badhaoge to spacious reading, ghataoge to cramped.
- **Text-Shadow:** Ye text pe shadow add karta hai. Memorize trick: "Shadow Like Spy". Related extra properties: Box-Shadow. Possible values: User-entered – h-shadow v-shadow blur color (jaise 2px 2px 5px red). Multiple shadows comma se. Behavior: Blur se soft shadow, color se colored glow.
- **Text-Overflow** (Missing add): Ye overflow text ko handle karta hai. Memorize trick: "Overflow Like Flood". Related: Overflow. Possible values: Fixed – clip (cut off), ellipsis (... show). Behavior: Ellipsis se truncated text with dots.

Code Example:

```
p {  
  text-align: center;  
  text-decoration: underline;  
  letter-spacing: 5px;  
}
```

```
text-transform: uppercase;
word-spacing: 8px;
text-indent: 40px;
line-height: 1.5; /* Changed to number for better */
text-shadow: 4px 4px 2px #ff0000;
text-overflow: ellipsis; /* Added */
}
```

## BACKGROUND PROPERTIES IN CSS

Background properties se document ka pichla hissa customize karte hain, jaise color, image, position. Jaise wall pe paint ya wallpaper.

- **Background-Image:** Ye background mein image set karta hai. Memorize trick: "Image on Back". Related: Background (shorthand). Possible values: User-entered – url('image.jpg'), none (default), linear-gradient(to right, red, blue). Behavior: Url se image show, gradient se color fade.
- **Background-Size:** Ye image ka size set karta hai. Memorize trick: "Size the Back". Related: Background-Position. Possible values: Fixed – auto (default original), cover (full cover zoom), contain (fit without crop). User-entered – length/percentage (100px 50%). Behavior: Cover se full fill par crop, contain se visible par gaps.
- **Background-Position:** Ye image ki position set karta hai. Memorize trick: "Position Like Photo Frame". Related: Background-Origin. Possible values: Fixed – top, bottom, left, right, center. User-entered – length/percentage (50% 50% center). Behavior: Center se middle, top left se corner.
- **Background-Repeat:** Ye image ko repeat karta hai. Memorize trick: "Repeat Like Echo". Possible values: Fixed – repeat (default both directions), repeat-x (horizontal), repeat-y (vertical), no-repeat, space (even space), round (scale to fit). Behavior: No-repeat se single image, repeat se tile pattern.
- **Background-Color:** Ye background color set karta hai. Memorize trick: "Color the Back". Possible values: User-entered – color names (red), hex (#ff0000), rgb(255,0,0), hsl(0,100%,50%), transparent (default). Behavior: Red se red bg, transparent se see-through.

- **Background-Attachment:** Ye image ko scroll ke sath fix karta hai. Memorize trick: "Attach Like Glue". Possible values: Fixed – scroll (default moves with content), fixed (stays on viewport), local (scrolls with element). Behavior: Fixed se parallax effect, scroll se normal.
- **Background-Origin:** Ye image ka starting point set karta hai. Memorize trick: "Origin Story". Possible values: Fixed – padding-box (default from padding), border-box (from border), content-box (from content). Behavior: Content-box se image content area mein start.
- **Background:** Shorthand for all background-\* . Memorize trick: "All in One Back". Possible values: Combination like color image position/size repeat attachment origin clip. Behavior: Ek line mein sab set.
- **Background-Clip** (Missing add): Ye bg area clip karta hai. Possible values: Fixed – border-box (default), padding-box, content-box. Behavior: Content-box se bg content tak limited.

Code Example:

```
body {
  background-image: url('codeHelp.png');
  background-size: cover;
  background-position: center;
  background-repeat: no-repeat;
  background-color: #ffffff;
  background-attachment: fixed;
  background-origin: content-box;
  background: url('codeHelp.png') no-repeat center/cover #ffffff;
  background-clip: padding-box; /* Added */
}
```

## BORDERS PROPERTIES IN CSS

Borders properties se elements ke around border change karte hain, jaise style, color, radius. Jaise frame lagana.

- **Border-Width:** Ye border ki width set karta hai. Memorize trick: "Width of Wall". Related: Border. Possible values: Fixed – thin, medium (default), thick.

User-entered – length (4px). Behavior: Thick se mota border.

- **Border-Style:** Ye border ka style set karta hai. Memorize trick: "Style the Edge". Possible values: Fixed – none (default no border), hidden, dotted (dots), dashed (dashes), solid (line), double, groove (3D in), ridge (3D out), inset, outset. Behavior: Dotted se dotted line, solid se full line.
- **Border-Radius:** Shorthand for corners round. Memorize trick: "Round Corners Like Ball". Related: Border-Top-Left-Radius etc. Possible values: User-entered – length/percentage (10px, 50% circle). Behavior: 50% se circle, 10px se soft corners.
- **Border-Color:** Ye border color set karta hai. Memorize trick: "Color the Border". Possible values: User-entered – colors like #000. Behavior: Black se black border.
- **Border:** Shorthand for width style color. Memorize trick: "Border All". Possible values: Combination like 1px solid red.
- **Border-Spacing:** Ye table cells ke borders ke beech space. Memorize trick: "Space Borders". Related: Border-Collapse. Possible values: User-entered – length (20px). Behavior: Badhaoge to cells alag.
- **Border-Image** (Missing add): Image as border. Possible values: Url, repeat etc. Behavior: Image border mein tile.

Code Example:

```
div {  
  border-width: 4px;  
  border-style: solid;  
  border-radius: 4px;  
  border-color: #000000;  
  border: 20px dotted coral;  
  border-spacing: 20px;  
  border-image: url('border.png') 30 round; /* Added */  
}
```

## BOX MODEL PROPERTIES IN CSS

Box model har element ko box jaise treat karta hai with content, padding, border, margin. Layout design ke liye.

- **Padding:** Shorthand for padding-top/right/bottom/left. Memorize trick: "Padding Like Cushion". Related: Box-Sizing. Possible values: User-entered – length/percentage (10px). Behavior: Badhaoge to content se border tak space.
- **Visibility:** Element ko visible ya hide. Memorize trick: "Visible or Vanish". Related: Display. Possible values: Fixed – visible (default), hidden (space rahega par dikhega nahi), collapse (table mein row hide). Behavior: Hidden se gone but layout same.
- **Display:** Element ka display type. Memorize trick: "Display Mode". Related: Visibility. Possible values: Fixed – block (full width new line), inline (same line), inline-block, none (remove completely), flex, grid. Behavior: None se completely gone, flex se flexible layout.
- **Height:** Element ki height. Memorize trick: "Height High". Related: Min-Height, Max-Height. Possible values: Fixed – auto (default content based). User-entered – length/percentage (100px, 50%). Behavior: Fixed height se overflow possible.
- **Width:** Element ki width. Similar to height. Possible values: Same as height.
- **Float:** Element ko left/right float. Memorize trick: "Float Like Boat". Related: Clear. Possible values: Fixed – none (default), left, right. Behavior: Left se text wrap around.
- **Clear:** Float ke baad element ko clear. Memorize trick: "Clear the Path". Possible values: Fixed – none, left, right, both. Behavior: Both se no float interference.
- **Margin:** Shorthand for margins. Memorize trick: "Margin Like Boundary". Possible values: User-entered – length/percentage/auto (10px). Behavior: Auto se center.
- **Overflow:** Content overflow handle. Memorize trick: "Overflow Control". Possible values: Fixed – visible (default show outside), hidden (cut), scroll (scrollbar), auto (if needed scrollbar). Behavior: Scroll se scrollbar add.

Code Example:



```
p {
padding: 10px 20px 10px 20px;
visibility: hidden;
display: inline-block;
height: auto;
width: 100px;
float: right;
clear: left;
margin: 20px 10px 20px 10px;
overflow: scroll;
}
```

## COLORS PROPERTIES IN CSS

Color properties se colors add karte hain text, outline pe.

- **Color:** Text color. Memorize trick: "Color Text". Possible values: User-entered – colors like rgb(0,0,0). Behavior: Black se black text.
- **Outline-Color:** Outline color (border jaise par outside). Possible values: Same as color.
- **Caret-Color:** Cursor color in inputs. Possible values: Same, auto default.
- **Opacity:** Element ki transparency. Memorize trick: "Opacity Like Glass". Possible values: User-entered – number 0-1 (0 invisible, 1 full). Behavior: 0.5 se half transparent.
- **Background-Color** (Already in background, but related here).

Code Example:

```
p {
color: rgb(0, 0, 0);
outline-color: #000000;
caret-color: coral;
opacity: 0.8;
}
```

## LAYOUT PROPERTIES IN CSS

Layout properties content ko arrange karte hain, jaise box direction.

Note: Kuch properties old hain like box-\*, modern mein flex/grid use karo, par explain kar raha.

- **Box-Align:** Perpendicular alignment. Possible values: Fixed – start, end, center, baseline, stretch.
- **Box-Direction:** Content direction. Possible values: Fixed – normal, reverse.
- **Box-Flex:** Grow to fill. Possible values: User-entered – number (0 default).
- **Box-Orient:** Horizontal/vertical layout. Possible values: Fixed – horizontal, vertical, inline-axis, block-axis.
- **Box-Sizing:** Width/height calculation. Memorize trick: "Sizing Box". Possible values: Fixed – content-box (default, padding/border extra), border-box (include all). Behavior: Border-box se total size fixed.
- **Box-Pack:** Pack contents. Possible values: Fixed – start, end, center, justify.
- **Min-Width:** Min width. Possible values: User-entered – length/percentage (0 default).
- **Max-Width:** Max width. Similar.
- **Min-Height, Max-Height:** Similar.

Code Example:

```
div {  
  box-align: start;  
  box-direction: normal;  
  box-flex: 1;  
  box-orient: horizontal;  
  box-sizing: border-box;  
  box-pack: justify;  
  min-width: 200px;  
  max-width: 400px;  
  min-height: 100px;
```

```
max-height: 1000px;  
}
```

## TABLE PROPERTIES IN CSS

Table properties tables ko customize.

- **Border-Spacing:** Cells ke borders space. Already in borders.
- **Border-Collapse:** Borders collapse. Possible values: Fixed – collapse (single border), separate (default).
- **Empty-Cells:** Empty cells show. Possible values: Fixed – show (default), hide.
- **Caption-Side:** Caption position. Possible values: Fixed – top (default), bottom.
- **Table-Layout:** Layout algorithm. Possible values: Fixed – auto (default content based), fixed (column widths fixed).

Code Example:

```
table {  
  border-spacing: 4px;  
  border-collapse: separate;  
  empty-cells: show;  
  caption-side: bottom;  
  table-layout: auto;  
}
```

## COLUMNS PROPERTIES IN CSS

Columns for multi-column layout.

- **Column-Gap:** Columns ke beech gap. Possible values: User-entered – length/normal (3.8em default).
- **Column-Rule-Width:** Rule width. Like border-width.
- **Column-Rule-Color:** Rule color.
- **Column-Rule-Style:** Rule style like border-style.

- **Column-Count:** Number of columns. User-entered – number/auto.
- **Column-Span:** Span across columns. Fixed – 1 (default), all.
- **Column-Width:** Column width. User-entered – length/auto.
- **Columns** (Missing add): Shorthand for width count.

Code Example:

```
div {
  column-gap: 4px;
  column-rule-width: medium;
  column-rule-color: #000000;
  column-rule-style: dashed;
  column-count: 3;
  column-span: all;
  column-width: auto;
  columns: 100px 3; /* Added */
}
```

## LIST & MARKER PROPERTIES IN CSS

Lists customize.

- **List-Style-Image:** Image as marker. Possible values: Url/none.
- **List-Style-Position:** Marker position. Fixed – outside (default), inside.
- **List-Style-Type:** Marker type. Fixed – disc (default ul), circle, square, decimal (ol), lower-roman, none.
- **Marker-Offset** (Old, use padding). Possible values: Length/auto.
- **List-Style** (Missing add): Shorthand.

Behavior: Disc se bullet, none se no marker.

Code Example:

```
ul {
  list-style-image: url('codeHelp.png');
```

```
list-style-position: inside;
list-style-type: square;
marker-offset: 10px;
list-style: square inside url('codeHelp.png'); /* Added */
}
```

## ANIMATION PROPERTIES IN CSS

Animations se transitions create.

- **Animation-Name:** Keyframes name. User-entered – string.
- **Animation-Delay:** Start delay. User-entered – time (2s, 500ms).
- **Animation-Duration:** Length. Time (default 0).
- **Animation-Timing-Function:** Speed curve. Fixed – ease (default slow start/end), linear, ease-in, ease-out, ease-in-out, cubic-bezier.
- **Animation-Iteration-Count:** Plays count. User-entered – number/infinite (1 default).
- **Animation-Fill-Mode:** Non-playing style. Fixed – none (default), forwards, backwards, both.
- **Animation-Play-State:** Running/paused. Fixed – running (default), paused.
- **Animation-Direction:** Direction. Fixed – normal (default), reverse, alternate, alternate-reverse.

Code Example:

```
div {
  animation-name: anime;
  animation-delay: 4ms;
  animation-duration: 10s;
  animation-timing-function: ease;
  animation-iteration-count: 5;
  animation-fill-mode: both;
  animation-play-state: running;
}
```

```
animation-direction: normal;
}
```

## TRANSITION PROPERTIES IN CSS

Transitions for state changes.

- **Transition-Property:** Which property. Fixed – all (default), none, or property names.
- **Transition-Delay:** Delay. Time.
- **Transition-Duration:** Duration. Time.
- **Transition-Timing-Function:** Curve. Same as animation.
- **Transition** (Missing add): Shorthand.

Behavior: Ease-in-out se smooth change.

Code Example:

```
div {
  transition-property: width;
  transition-delay: 4ms;
  transition-duration: 10s;
  transition-timing-function: ease-in-out;
  transition: width 10s ease-in-out 4ms; /* Added */
}
```

## CSS FLEXBOX

Flexbox se items align aur distribute easy, responsive ke liye. Jaise rubber band items.

### PARENT PROPERTIES:

- **Display:** Flex enable. Fixed – flex, inline-flex.
- **Flex-Direction:** Items direction. Fixed – row (default), row-reverse, column, column-reverse. Behavior: Column se vertical stack.

- **Flex-Wrap:** Wrap or not. Fixed – nowrap (default), wrap, wrap-reverse. Behavior: Wrap se multi-line.
- **Flex-Flow:** Shorthand direction wrap.
- **Justify-Content:** Main axis align. Fixed – flex-start (default), flex-end, center, space-between (space between), space-around (around), space-evenly. + safe/unsafe. Behavior: Space-between se even gaps no ends.
- **Align-Items:** Cross axis align. Fixed – stretch (default), flex-start, flex-end, center, baseline. Behavior: Center se middle.
- **Align-Content:** Multi-line align. Similar to justify but for lines.

Code Example:

```
.container {
  display: flex;
  flex-direction: row;
  flex-wrap: wrap;
  flex-flow: row wrap;
  justify-content: space-between;
  align-items: center;
  align-content: space-around;
}
```

## CHILD PROPERTIES:

- **Order:** Order change. User-entered – number (0 default). Behavior: Higher number last.
- **Flex-Grow:** Grow ratio. Number (0 default).
- **Flex-Shrink:** Shrink ratio. Number (1 default).
- **Flex-Basis:** Initial size. Length/percentage/auto (auto default).
- **Align-Self:** Self align. Same as align-items, auto default.
- **Flex:** Shorthand grow shrink basis.

Code Example:

```
.item {  
  order: 2;  
  flex-grow: 5;  
  flex-shrink: 4;  
  flex-basis: auto;  
  align-self: center;  
  flex: 5 4 auto;  
}
```

## CSS GRID

Grid 2D layout for complex designs.

### PARENT PROPERTIES:

- **Display:** Grid enable. Fixed – grid, inline-grid.
- **Grid-Template-Columns:** Columns define. User-entered – lengths/repeat(3, 1fr) (fr fraction). Behavior: 1fr 2fr se ratio.
- **Grid-Template-Rows:** Rows similar.
- **Grid-Template:** Shorthand rows/columns/areas.
- **Column-Gap:** Column gaps. Length.
- **Row-Gap:** Row gaps.
- **Gap:** Shorthand.
- **Grid-Gap:** Old shorthand.
- **Align-Items:** Item align. Fixed – start, end, center, stretch (default).
- **Justify-Content:** Similar for main.
- **Align-Content:** For lines.
- **Grid-Auto-Columns:** Auto column size.
- **Grid-Auto-Rows:** Auto row.
- **Grid-Auto-Flow:** Placement. Fixed – row (default), column, dense.

Code Example:



```

.container {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  grid-template-rows: auto;
  grid-template: auto / repeat(3, 1fr);
  column-gap: 10px;
  row-gap: 10px;
  gap: 10px;
  align-items: center;
  justify-content: space-evenly;
  align-content: center;
  grid-auto-columns: 100px;
  grid-auto-rows: auto;
  grid-auto-flow: dense;
}

```

## CHILD PROPERTIES:

- **Grid-Column-Start:** Column start. User-entered – number/name/span number/auto.
- **Grid-Column-End:** End.
- **Grid-Row-Start/End:** Similar for row.
- **Grid-Column:** Shorthand start/end.
- **Grid-Row:** Similar.
- **Grid-Area:** Name or positions.
- **Align-Self:** Self align. Fixed – start, end, center, stretch.
- **Justify-Self** (Missing add): Similar for justify.

Code Example:

```

.item {
  grid-column-start: 1;
  grid-column-end: span 2;
}

```

```
grid-row-start: auto;  
grid-row-end: 3;  
grid-column: 1 / span 2;  
grid-row: 1 / 3;  
grid-area: header;  
align-self: start;  
justify-self: center; /* Added */  
}
```