# CSS Grid's Multi-Pass Placement Algorithm

Understanding how CSS Grid places items is crucial for predictable layouts. The browser doesn't randomly assign positions—it follows a systematic, multi-pass approach that processes items in a specific order based on their positioning constraints.

01

## Fully Positioned Items First

Items with explicit row AND column values get placed first. These have the highest priority since their position is completely defined.

02

## Row-Defined Items Second

Next, items with a specific row but automatic column placement. Grid knows which row they belong to and finds available column space.

03

## Column-Defined Items Third

Then items with a specific column but automatic row placement. Grid assigns them to the earliest available row in their designated column.

04

## Auto-Positioned Items Last

Finally, items with both row and column set to auto. These fill remaining gaps using the auto-placement algorithm.

# CSS Animations

CSS Animations provide more control than transitions, allowing you to define multi-step animation sequences. Unlike transitions that only animate between two states, animations use keyframes to specify styles at various points during the animation timeline.

```
@keyframes myName {
  0% { font-size: 20px; }
  50% { font-size: 30px; }
  100% { font-size: 40px; }
}
```

You can use percentage values (0%, 50%, 100%) to define multiple intermediate steps in your animation. The keywords `from` equals 0% and `to` equals 100%.

## Key Animation Properties

### animation-name

References the `@keyframes` rule to use.

### animation-duration

How long the animation takes to complete one cycle.

### animation-timing-function

Controls the acceleration curve (e.g., `ease`, `linear`, `ease-in-out`).

### animation-delay

Time to wait before starting the animation.

### animation-iteration-count

Number of times to repeat the animation (or `infinite`).

### animation-direction

Whether animation plays forward, reverse, or alternates.

## Animation Shorthand

All individual animation properties can be combined into a single, concise declaration using the shorthand property. The order of values is important, especially for `duration` and `delay`. If both are present, the first time value is `duration` and the second is `delay`.

```
animation: name duration timing-function delay iteration-count direction fill-mode play-state;
```

Here's an example using the shorthand:

```
/* Example */
animation: myName 2s linear 3s infinite normal;
```

# Media Queries

Media queries are a CSS feature that enables responsive web design by applying different styles based on device characteristics. They allow you to create layouts that adapt to various screen sizes, orientations, and display capabilities, ensuring your website looks great on any device.

```
@media (condition) {
  /* CSS rules applied when condition is true */
}
```

## Common Media Query Examples

### Screen Width

```
@media (max-width: 768px) { }
```

Targets screens smaller than 768px.

### Min Width

```
@media (min-width: 1024px) { }
```

Targets screens larger than 1024px.

### Orientation

```
@media (orientation: landscape) { }
```

Targets landscape mode.

### Combined Conditions

```
@media (min-width: 768px) and (max-width: 1024px) {
}
```

Targets tablet range.