



Testing for selector support

This post assumes you're working on a modern browser that supports `@supports` and the target features you want to use. This is not always a safe assumption to make.

[CSS Conditional Rules Module Level 3](#) defines CSS Feature Queries and the `@supports` rule that allows developers to code defensively and only use a feature if it's supported in browsers and provide fallbacks when it's not.

For example, if we wanted to have different layouts for browsers that support grid and those that don't we can do something like this:

```
@supports (display: grid) {  
  .container {  
    display: grid;  
  }  
}  
  
@supports not (display: grid) {  
  .container {  
    display: block;  
    /*  
      Use Flex or some other layout  
      to replace grid  
    */  
  }  
}
```

[CSS Conditional Rules Module Level 4](#) introduced the `selector` rule for feature queries.

[Miriam Suzanne](#) from the Firefox team does a good job introducing selector feature queries.

The `::marker` CSS pseudo-element matches the marker box of a list item, which typically contains a bullet or number.

The following feature query matches on browsers that support the [::marker](#) pseudo-element.

```
@supports selector(::marker) {  
  ol li::marker {  
    color: rebeccapurple;  
    font-family: cursive;  
    font-size: 2em;  
    font-weight: 900;  
    content: '✅';  
  }  
  
  '✅'li {  
    padding-left: 1em;  
  }  
}
```

This may be tricky as you have to evaluate whether your target browsers support the features that you want both in terms of features and their component rules.

In the `::marker` example above, a browser may support some features of the pseudo element like color, font family, size or weight and not others like content

with an emoji as its value. If this is the case you'll have to decide if this is ok or not.

Support List

The following browsers support selector feature queries

- Firefox 69
- Chromium browsers: Under development; See [crbug 979041](#)