```html
<!DOCTYPE html>
<html lang="en">
  <head>
    …
    <meta name="viewport" content="width=device-width, initial-scale=1">
    …
  </head>
  …
```

Scaling the page to device’s resolution

Alternatively, you could include print styles within your main stylesheet using a media query:

```css
@media print {
  /* print styles go here */
}
```

Able to print styles using CSS

```css
@media (min-width: 360px) {
  body {
    font-size: 1.0em;
  }
}

@media (min-width: 500px) {
  .seven-day-fc .temp-low,
  .seven-day-fc .temp-high {
    display: inline-block;
    width: 45%;
  }

  .seven-day-fc .seven-day-temp {
    margin-left: 5%;
  }

  .seven-day-fc .icon {
    width: 64px;
    height: 64px;
  }
}
```

Optimising the screen based on screen pixel resolution

Let's take a deeper look at the above blog post example. On smaller screens, the Roboto font at `1em` works perfectly giving 10 words per line, but larger screens require a breakpoint. In this case, if the browser width is greater than `575px`, the ideal content width is `550px`.

```css
@media (min-width: 575px) {
  article {
    width: 550px;
    margin-left: auto;
    margin-right: auto;
  }
}
```

Optimising text based on screen pixel resolution.

CSS responsive web design can be done by simply using @media then what the user wants to implement for a responsive web design feature. It queries to apply different web design styles for media types/devices. This can be used to check such as device information and these can be applied to custom settings.