



**I have the following css file, and want to convert all of the absolute values to relative ones so that the page looks correct on all device sizes, but am having trouble with vh and px. Could you help with this conversion? The browser screen that this was designed on was 1920×878.**

```
head{
  margin: 0;
}

body{
  font-family: Georgia, serif;
  padding: 0;
  margin: 0;
  background-color: #e4eff9;
  color: black;
}

.portal{
  text-align: center;
  font-family: Georgia, serif;
  margin-bottom: 2%;
}

.square{
  height: 50px;
  width: 50px;
  background-color: navy;
}

.headbanner{
  width: 100%;
  height: 11vh;
  background-color: #4d91cc;
  background-position: center;
}

.contentbanner{
  width: 20%;
```

```
height: 32vh;
background-color: #c1dcf3;
background-position: center;
position: absolute;
border-radius: 15px;
border-color: #4d91cc;
border-width: 2px;
border-style: solid;
top: 20%;
left: 40%;
color: black;
font-weight: bold;
font-size: 24px;
}
```

```
.tailbanner{
width: 100%;
height: 6vh;
bottom: 0;
background-color: #4d91cc;
background-position: center;
position: absolute;
color: #e4eff9;
font-weight: normal;
font-size: 18px;
text-align: right;
}
```

```
.formitem{
background-color: #eef3f8;
margin: 0.3%;
font-family: Georgia, serif;
text-align: left;
color: black;
border-radius: 15px;
text-indent: 5px;
height: 3vh;
}
```

```
.error{
color: red;
/* border-radius: 2px;
border-width: 1px;
border-color: red;
border-style: solid; */
position: absolute;
width: 10%;
```

```
height: 2vh;
top: 52%;
left: 45%;
text-align: center;
font-style: italic;
font-size: 12px;
}
```

```
#logo1{
  position: absolute;
  left: 10px;
  top: 6px;
  width: 237px;
  height: 70px;
}
```

```
#logo2{
  position: absolute;
  right: 10px;
  top: 6px;
  width: 152px;
  height: 70px;
}
```

```
#submit{
  padding-left: 10%;
  padding-right: 10%;
  text-align: center;
  text-indent: 0px;
  margin-top: 1.5%;
}
```

```
#loginmode{
  text-indent: 0px;
  margin-bottom: 1.5%;
}
```

A good approach is to treat your original 1920×878 screen as the “design baseline,” then convert fixed pixel sizes and positions into viewport units using simple formulas, and clean up some absolute positioning so the layout recenters itself on different screens.

Below is a revised version of your CSS with comments and conversions; afterwards there is a brief explanation of what was done so you can adapt it further.

```
/* No need for head here; usually use html, body */
html, body{
  font-family: Georgia, serif;
  padding: 0;
  margin: 0;
```

```

    background-color: #e4eff9;
    color: black;
}

/* Centering helper for the main content */
body {
    display: flex;
    flex-direction: column;
    min-height: 100vh; /* full viewport height */
}

.portal{
    text-align: center;
    font-family: Georgia, serif;
    margin-bottom: 2vh; /* relative to height for vertical spacing */
}

.square{
    /* 50px on a 1920px-wide viewport => 50 / 1920 * 100 ≈ 2.6vw [web:1][web:4] */
    height: 2.6vw;
    width: 2.6vw;
    background-color: navy;
}

/* Top banner stays full width and relative height */
.headbanner{
    width: 100%;
    height: 11vh;
    background-color: #4d91cc;
    background-position: center;
    position: relative;
}

/* Content card centered with flex instead of hard-coded top/left */
.contentbanner{
    width: 20vw;          /* 20% of viewport width */
    min-width: 260px;     /* optional: keep it usable on very small screens */
    height: 32vh;
    background-color: #c1dcf3;
    background-position: center;
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%); /* center in viewport */
    border-radius: 15px;
    border-color: #4d91cc;
    border-width: 2px;
    border-style: solid;
    color: black;
    font-weight: bold;

    /* 24px font-size "fluid" with clamp:
       min 1.2rem (~19px), preferred 1.5vw, max 1.6rem (~26px) [web:3][web:6][web:9] */
    font-size: clamp(1.2rem, 1.5vw, 1.6rem);
}

```

```

/* Footer glued to bottom with relative font size */
.tailbanner{
  width: 100%;
  height: 6vh;
  margin-top: auto;          /* push to bottom in flex layout */
  background-color: #4d91cc;
  background-position: center;
  color: #e4eff9;
  font-weight: normal;
  /* 18px => clamp around that size */
  font-size: clamp(1rem, 1.1vw, 1.3rem);
  text-align: right;
  display: flex;
  align-items: center;
  padding-right: 2vw;
}

/* Form items: relative height and spacing */
.formitem{
  background-color: #eef3f8;
  margin: 0.3vh 0.3vw;
  font-family: Georgia, serif;
  text-align: left;
  color: black;
  border-radius: 15px;
  text-indent: 0.5vw;
  height: 3vh;
}

/* Error message: center under content card instead of fixed top/left */
.error{
  color: red;
  position: absolute;
  left: 50%;
  top: calc(50% + 18vh); /* under .contentbanner (half of 32vh is 16vh, plus a bit) */
  transform: translateX(-50%);
  width: 30vw;          /* scale with viewport */
  max-width: 400px;
  height: auto;
  text-align: center;
  font-style: italic;

  /* 12px => smaller fluid size */
  font-size: clamp(0.75rem, 0.8vw, 0.9rem);
}

/* Logos originally: left/right 10px, top 6px, fixed widths/heights.
   Convert positions to vw using 1920px width: 10px ≈ 0.52vw, 6px ≈ 0.31vh [web:1][web:4]
#logo1{
  position: absolute;
  left: 0.52vw;
  top: 0.7vh;

  /* Scale logos with viewport width, roughly proportional to original:
     237px width => 237/1920*100 ≈ 12.3vw, 70px height => 70/878*100 ≈ 8vh [web:1][web:
  width: 12.3vw;

```

```

    height: 8vh;
    object-fit: contain;
}

#logo2{
  position: absolute;
  right: 0.52vw;
  top: 0.7vh;
  width: 8vw; /* 152/1920*100 ≈ 7.9vw */
  height: 8vh;
  object-fit: contain;
}

#submit{
  padding-left: 10%;
  padding-right: 10%;
  text-align: center;
  text-indent: 0;
  margin-top: 1.5vh;
}

#loginmode{
  text-indent: 0;
  margin-bottom: 1.5vh;
}

```

## How the conversions work

- For widths and horizontal lengths that were px, convert to vw using `px / 1920 * 100` because your baseline design width is 1920px. [\[1\]](#) [\[2\]](#) [\[3\]](#)
- For vertical pixel values (like the original logo top offsets), convert to vh using `px / 878 * 100` because your design height is 878px. [\[4\]](#)
- For positions of main elements (`contentbanner`, `.error`), remove hard-coded percentages tied to one screen and instead center using `top: 50%`, `left: 50%`, and `transform: translate(-50%, -50%)`, or position relative to that center. [\[5\]](#) [\[6\]](#) [\[7\]](#)
- For text, use `clamp(min, preferred, max)` with a mix of `rem` and `vw` so font sizes scale smoothly but stay readable on very small or very large screens. [\[8\]](#) [\[9\]](#) [\[10\]](#)

If you share the corresponding HTML, the layout can be refined further (for example, replacing more absolute positioning with flexbox or CSS grid).

✱

## That works, thank you so much!

Glad it helped; you're very welcome! If you run into any other layout issues or want to refactor more CSS to be responsive, feel free to share a snippet and the target behavior.

1. <https://cssunitconverter.com/px-to-vw>
2. <https://cssunitconverter.vercel.app/px-to-vw>

3. <https://cssunitconvert.com/convert/px-to-vw>
4. <https://uisurgeon.com/tools/vh-to-vw-converter>
5. <https://stackoverflow.com/questions/15194335/responsive-absolute-position-to-center-of-div>
6. [https://www.reddit.com/r/Frontend/comments/yava61/best\\_way\\_to\\_center\\_absolute\\_position\\_or\\_flex/](https://www.reddit.com/r/Frontend/comments/yava61/best_way_to_center_absolute_position_or_flex/)
7. <https://ishadeed.com/article/less-absolute-positioning-modern-css/>
8. <https://onenine.com/css-clamp-for-font-scaling-how-it-works/>
9. <https://blog.logrocket.com/fluid-vs-responsive-typography-css-clamp/>
10. <https://blog.openreplay.com/responsive-typography-with-clamp/>