Responsive Web Design

1. Define standard breakpoints (320px-480px: Mobile devices etc..)
2. Use the **viewport meta tag** to control layout on mobile browsers inside the index.html **>
<meta name="viewport" content="width=500, initial-scale=1">**
3. Use **scalable units** (ems, rem, instead of px)
4. Consider image optimization using **srcset** (to avoid using high-quality images on low-resolution screens) - browser technology continues to improve *the client* (browser) gets to make the decision what to use

```
<img
 srcset="
 /wp-content/uploads/flamingo4x.jpg 4x,
 /wp-content/uploads/flamingo3x.jpg 3x,
 /wp-content/uploads/flamingo2x.jpg 2x,
 /wp-content/uploads/flamingo1x.jpg 1x
 "
src="/wp-content/uploads/flamingo-fallback.jpg"
>
```

- Largest or original — image-4x.jpg
- Scaled down to 75% — image-3x.jpg
- Scaled down to 50% — image-2x.jpg
- Scaled down to 25% — image-1x.jpg

Specify Image density (2x) or image Width, Height:
<img srcset=" /wp-content/uploads/flamingo4x.jpg 4025w, /wp-content/uploads/flamingo3x.jpg 1006w">

Cognizant

Responsive Web Design

Consider Fluid Grid Systems Fluid grids allow to create responsive designs which suit dynamic screen sizes.

The complexity of developing a fluid grid can be minimized by using an existing CSS framework. – Tailwind, fluidable.com, MaterialUI, bootstrap-react or **even better to make your own**

Adjust the fluid grid when necessary according to your design and try to provide the best browsing experience for the user.

Cognizant