

Responsive Web Design

Mendel Rosenblum

Web App Challenges: Screen real estate

320x640

640x320

768x1024

768x1024

1920x1028

Cell Phones

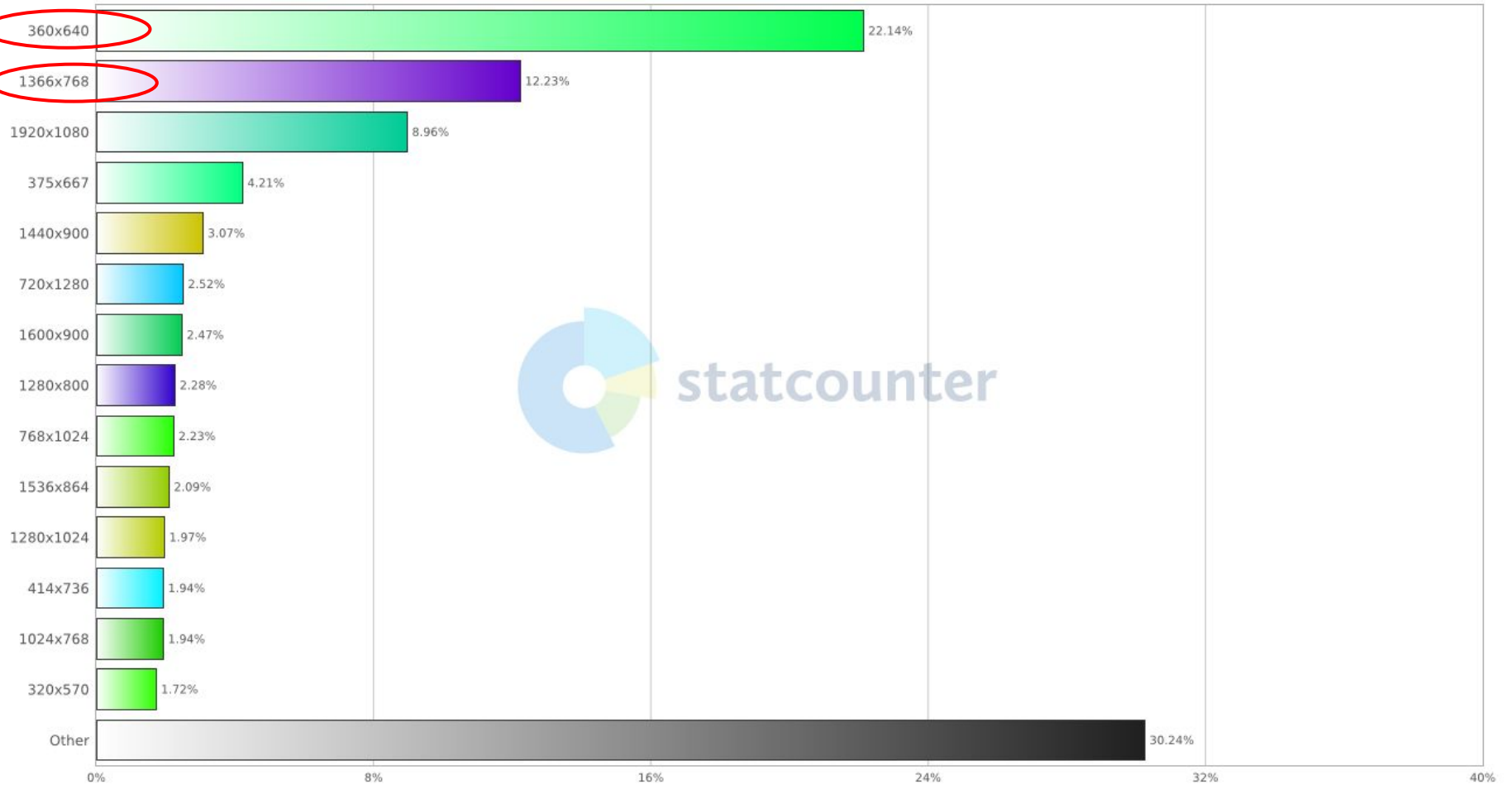
Tablets

Desktops

- Do we need to build N versions of each web application?

StatCounter Global Stats

Screen Resolution Stats Worldwide on 26 Apr 2018



Responsive Web Design

- Context is like water!
 - The web app should flow into and fill whatever device you have.
- Possible with recent CSS extensions:
 - Add grid layout system with relative (e.g. 50%) rather than absolute (e.g. 50pt) measures
 - Specify element packing into columns and rows
 - Add @media rules based on screen sizes
 - Switch layout based on screen size
 - Made images support relative sizes
 - Autoscale image and videos to fit in screen region

```
img { width: 100%; height: auto; }
video { width: 100%; height: auto; }
```

Example of Responsive Web Layout

Menu #1 - 25%	Menu #2 - 25%	Menu #3 - 25%	Menu #4 - 25%
Nav #1 - 25%	View component - 75%		
Nav #2 - 25%			
Nav #3 - 25%			
Footer - 100%			

Menu #1 - 25%	Menu #2 - 25%	Menu #3 - 25%	Menu #4 - 25%
Nav #1 - 25%	View component - 75%		
Nav #2 - 25%			
Nav #3 - 25%			
Footer - 100%			

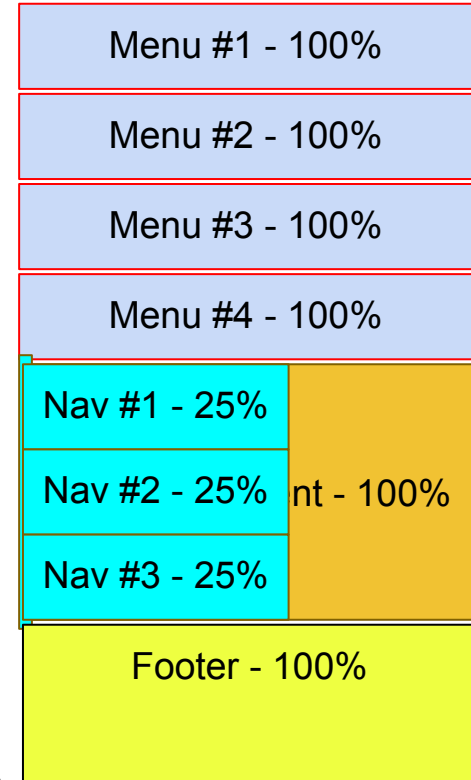
CSS Breakpoints

CSS Rules:

```
@media only screen and (min-width: 768px) {  
  /* tablets and desktop layout */ }
```

```
@media only screen and (max-width: 767px) {  
  /* phones */ }
```

```
@media only screen and (max-width: 767px)  
  and (orientation: portrait) {  
  /* portrait phones */ }
```



Responsive implementation

- Build components to operate at different screen sizes and densities
 - Use relative rather than absolute
 - Specify sizes in device independent units
- Use CSS breakpoints to control layout and functionality
 - Layout alternatives
 - App functionality conditional on available screen real estate
- Mobile first popular
 - Expand a good mobile design to use more real estate