



NAZARBAYEV  
UNIVERSITY



# Web Programming and Problem Solving

## **Responsive Web Design**

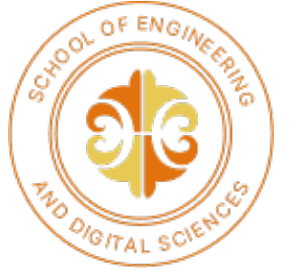
Date: 19.09.2022

Instructor: Zhandos Yessenbayev



NAZARBAYEV  
UNIVERSITY

# Content

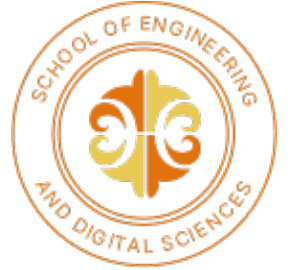


- Responsive Web Design
  - What is RWD?
  - Main Strategies
    - Media Queries
    - Flexible Layouts
    - Grid View
    - Responsive Images
    - Relative Units
    - Viewport



NAZARBAYEV  
UNIVERSITY

# What is Responsive Web Design?



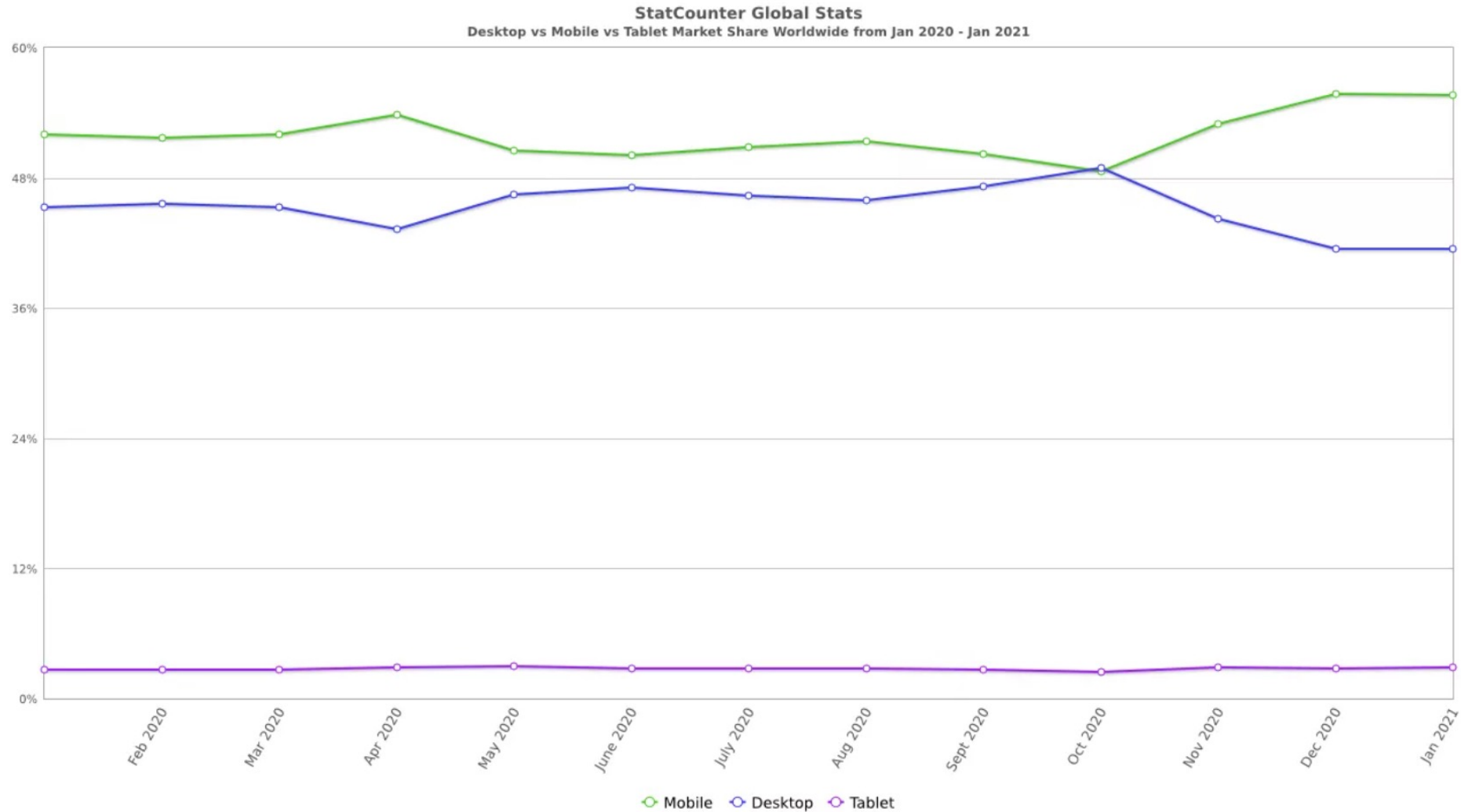
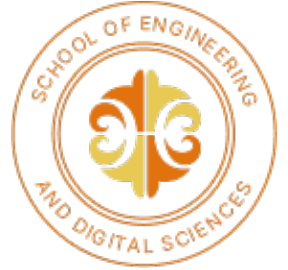
# What is Responsive Design?

- **Responsive web design** refers to a design strategy to make websites render correctly for various devices (mobile, tablet, and desktop).



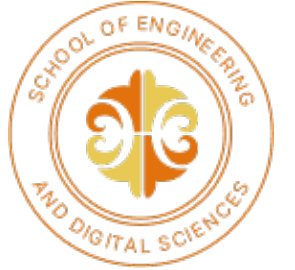


# Device Statistics





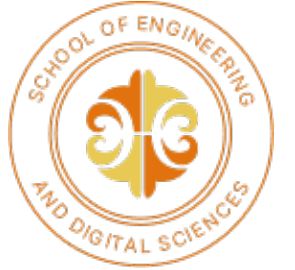
# Main Strategies



Here are some main strategies:

- Media queries
- Flexible layouts
- Grid View
- Responsive images
- Relative units
- Viewport

# Media queries



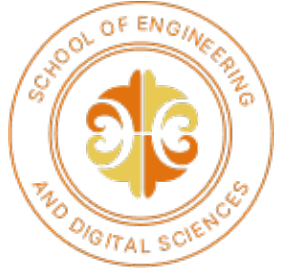
**Media queries** are **@media** rules used to change CSS properties if some conditions are true

```
@media not|only mediatype and  
      (mediafeature and|or|not mediafeature) {  
    CSS-Code;  
}
```

**mediatype**: all, print, **screen**, speech

**mediafeature**: max-width, min-width, orientation, etc.

# Media queries



Media queries use **breakpoints** (screen sizes) to decide how content needs to be rendered

Common breakpoints (in pixels):

- **Mobile:** up to 767
- **Tablet:** 768 – 1023
- **Desktop:** 1024 – 1179
- **Desktop (extra):** 1200 +

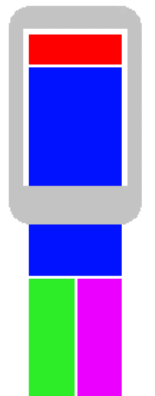
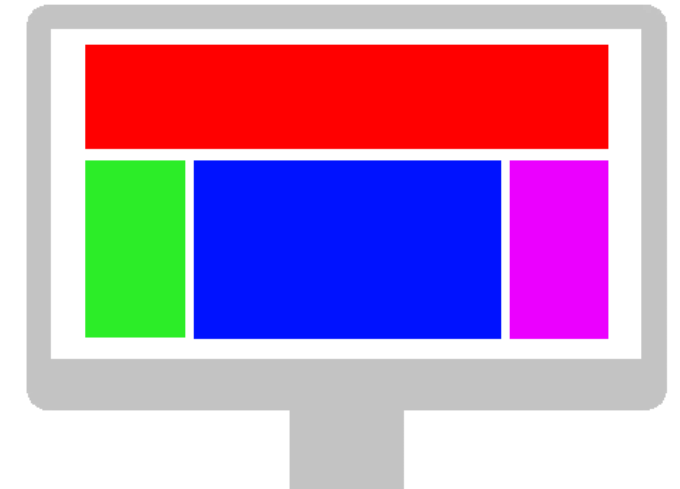
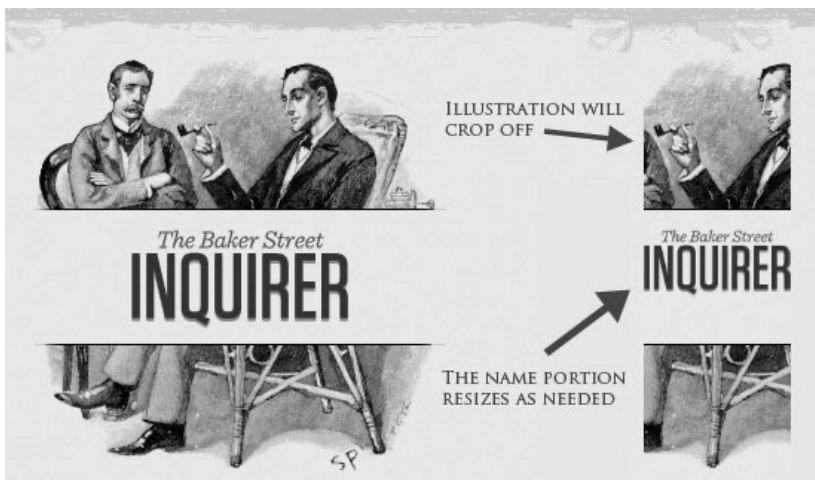
```
@media screen and (min-width: 768px) {  
  p {  
    width: 100%;  
    background-color: lightblue;  
  }  
}
```



# Flexible layouts

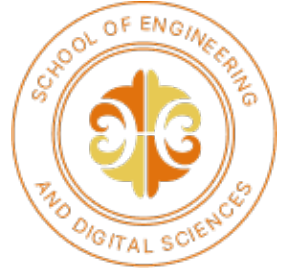
Website layout can be optimized for a device's screen size:

- reposition elements
- resize elements (images)
- hide or partially show elements





# Grid View



Many web pages are based on a **grid-view**, which means that the page is divided into columns:

- a responsive grid-view often has **12 columns**
- the **percentage** for one column:  $100\% / 12 \text{ columns} = 8.33\%$
- the columns should be **floating** to the left and other elements **cleared**
- make **one class** for each of the 12 columns: `.col-1 {width: 8.33%;}`, etc.
- The columns should be wrapped with a **row**, where they **add up** to 12

# Responsive images

Some techniques to make images responsive:

- Set **max-width** and **height**
- Use **picture** and **srcset**
- Vectors (svg) vs Bitmap (jpg)

```
img {  
  max-width: 100%;  
  height: auto  
}
```

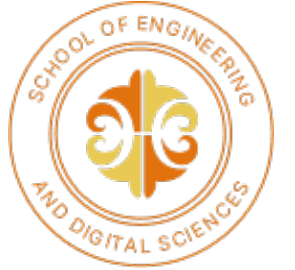
```
<picture>  
  <source srcset="small.jpg" media="(max-width: 768px)">  
  <source srcset="medium.jpg" media="(max-width: 1024px)">  
  <source srcset="large.jpg">  
    
</picture>
```

## Responsive Images Test Page

In supported browsers, the following image will load either small or large version depending on screen resolution, making a single 1kb request before requesting the appropriate size.



# Relative units



Use **relative units** for width/height, font size, etc.

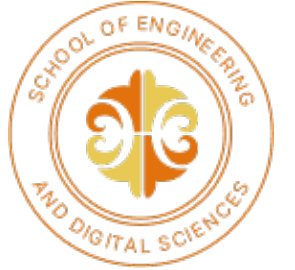
- percentage (width: 100%)
- em (font size: 1 em)
- vw, vh, vmin, vmax (height: 100vh)

One more important addition to RWD strategy is to add the following magic line in HTML code:

```
<meta name="viewport"  
      content="width=device-width, initial-scale=1.0">
```

The **viewport** is the user's visible area of a web page.

**Content** is sized to the **viewport** and the **scaling factor** set to 1



- **Key takeaways:**

- **Content is like water**, so target all types of devices (mobile, tablet, dektop)
- To make your website responsive, use:
  - **Media queries**
  - **Flexible layouts**
  - **Grid view**
  - **Responsive images**
  - **Relative units**
  - **Viewport**

Thanks for Attention!