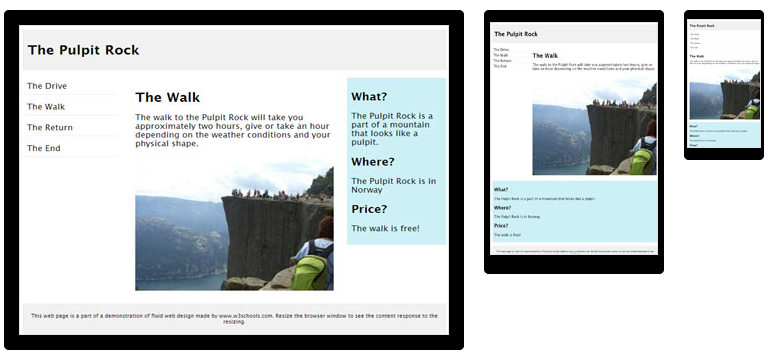
**What is Responsive Web Design?**

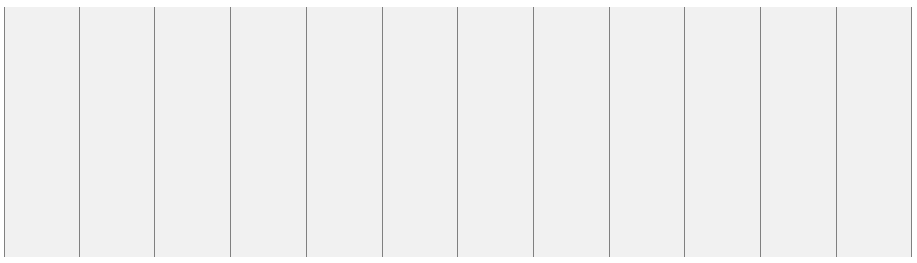
Responsive Web Design makes your web page look good on all devices (desktops, tablets, and phones).

Responsive Web Design is about using CSS and HTML to resize, hide, shrink, enlarge, or move the content to make it look good on any screen:

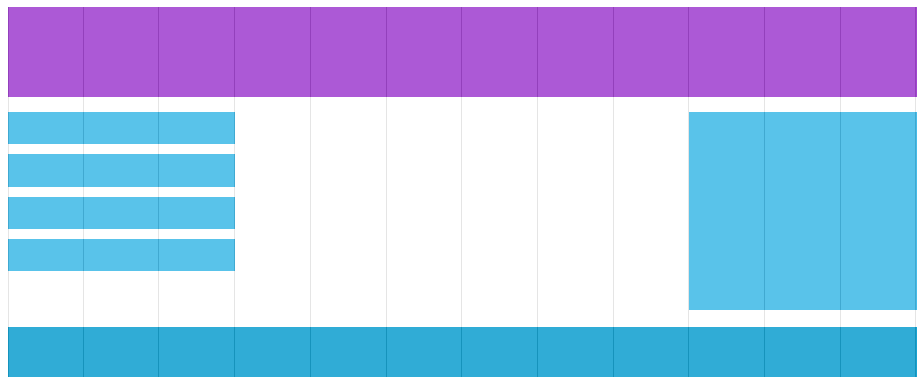


# Responsive Web Design - Grid-View

## What is a Grid-View?

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

Using a grid-view is very helpful when designing web pages. It makes it easier to place elements on the page.

A responsive grid-view often has 12 columns, and has a total width of 100%, and will shrink and expand as you resize the browser window.

**Building a Responsive Grid-View**

Lets start building a responsive grid-view.

1. First ensure that all HTML elements have the box-sizing property set to border-box. This makes sure that the padding and border are included in the total width and height of the elements.

Add the following code in your CSS:

\* {  
    box-sizing: border-box;  
}

2. However, we want to use a responsive grid-view with 12 columns, to have more control over the web page.

First we must calculate the percentage for one column: 100% / 12 columns = 8.33%.

Then we make one class for each of the 12 columns, class="col-" and a number defining how many columns the section should span:

### CSS:

.col-1 {width: 8.33%;}  
.col-2 {width: 16.66%;}  
.col-3 {width: 25%;}  
.col-4 {width: 33.33%;}  
.col-5 {width: 41.66%;}  
.col-6 {width: 50%;}  
.col-7 {width: 58.33%;}  
.col-8 {width: 66.66%;}  
.col-9 {width: 75%;}  
.col-10 {width: 83.33%;}  
.col-11 {width: 91.66%;}  
.col-12 {width: 100%;}

1. All these columns should be floating to the left, and have a padding of 15px:

### CSS:

[class\*="col-"] {  
    float: left;  
    padding: 15px;  
    border: 1px solid red;  
}

1. Each row should be wrapped in a <div>. The number of columns inside a row should always add up to 12:

### HTML:

<div class="row">  
  <div class="col-3">...</div> <!-- 25% -->  
  <div class="col-9">...</div> <!-- 75% -->  
</div>

1. The columns inside a row are all floating to the left, and are therefore taken out of the flow of the page, and other elements will be placed as if the columns do not exist. To prevent this, we will add a style that clears the flow:

### CSS:

.row::after {  
    content: "";  
    clear: both;  
    display: block;  
}

## Add a Breakpoint

Media queries can help with that. We can add a breakpoint where certain parts of the design will behave differently on each side of the breakpoint.

**  
Desktop**

**  
Phone**

Use a media query to add a breakpoint at 768px:

### Example

When the screen (browser window) gets smaller than 768px, each column should have a width of 100%:

/\* For mobile phones: \*/  
[class\*="col-"] {  
    width: 100%;  
}  
@media only screen and (min-width: 600px) {  
    /\* For tablets: \*/  
    .col-m-1 {width: 8.33%;}  
    .col-m-2 {width: 16.66%;}  
    .col-m-3 {width: 25%;}  
    .col-m-4 {width: 33.33%;}  
    .col-m-5 {width: 41.66%;}  
    .col-m-6 {width: 50%;}  
    .col-m-7 {width: 58.33%;}  
    .col-m-8 {width: 66.66%;}  
    .col-m-9 {width: 75%;}  
    .col-m-10 {width: 83.33%;}  
    .col-m-11 {width: 91.66%;}  
    .col-m-12 {width: 100%;}  
}

@media only screen and (min-width: 768px) {  
    /\* For desktop: \*/  
    .col-1 {width: 8.33%;}  
    .col-2 {width: 16.66%;}  
    .col-3 {width: 25%;}  
    .col-4 {width: 33.33%;}  
    .col-5 {width: 41.66%;}  
    .col-6 {width: 50%;}  
    .col-7 {width: 58.33%;}  
    .col-8 {width: 66.66%;}  
    .col-9 {width: 75%;}  
    .col-10 {width: 83.33%;}  
    .col-11 {width: 91.66%;}  
    .col-12 {width: 100%;}  
}