skill academy

# CSS Nav Bar and Responsive web Design

# Navigation Bar

- Navigation bar comes under **Graphical User Interface.**
- It helps the webpage user to access information by **navigating** to other sections of the website using **links**.
- Usually the navigation bar is on the **top** of the web page as a **horizontal list of links**.
- It can be placed **below the header or logo**, but it must always be placed **before** the main content.
- A website with **easy-to-use** navigation bar allows the user to visit any section easily and quickly.
- **Standard HTML** is the base for the Navigation Bar.
- Basically **Navigation Bar** is a **list of links**.

**Navigation bar**

Vertical Navigation bar

Horizontal Navigation bar

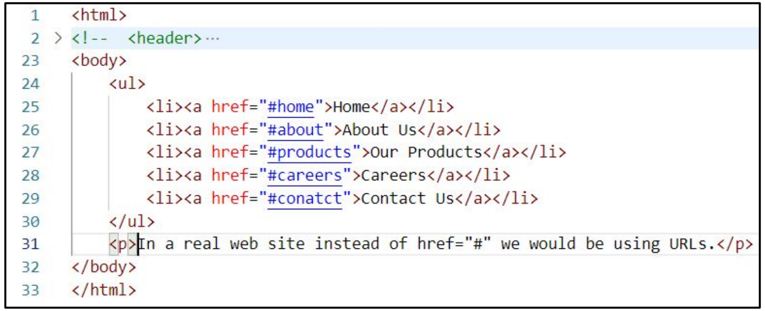
Responsive Web Design

RWD - Examples

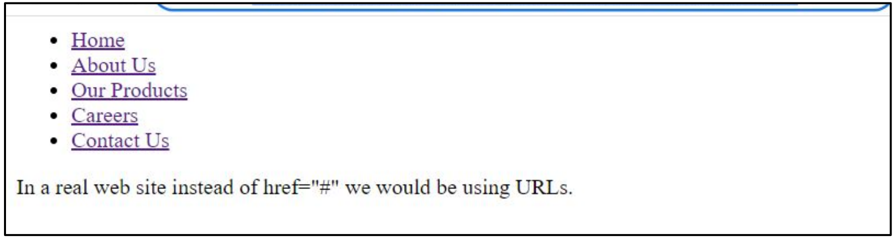
# Navigation Bar

skill academy

## Example: Creating a Nav bar using HTML list element



```
<html>
<!--  <header>···
<body>
    <ul>
        <li><a href="#home">Home</a></li>
        <li><a href="#about">About Us</a></li>
        <li><a href="#products">Our Products</a></li>
        <li><a href="#careers">Careers</a></li>
        <li><a href="#conatct">Contact Us</a></li>
    </ul>
    <p>In a real web site instead of href="#" we would be using URLs.</p>
</body>
</html>
```



- Home
- About Us
- Our Products
- Careers
- Contact Us

In a real web site instead of href="#" we would be using URLs.

**Navigation bar**

Vertical Navigation bar

Horizontal Navigation bar

Responsive Web Design

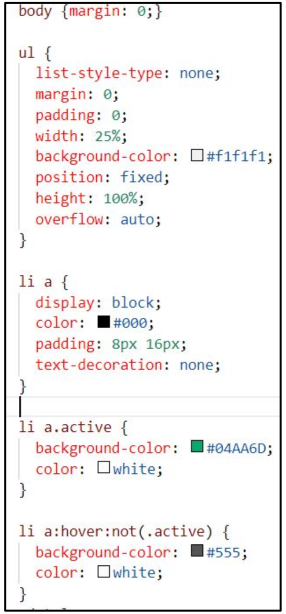
RWD - Examples

# Types of Navigation Bar

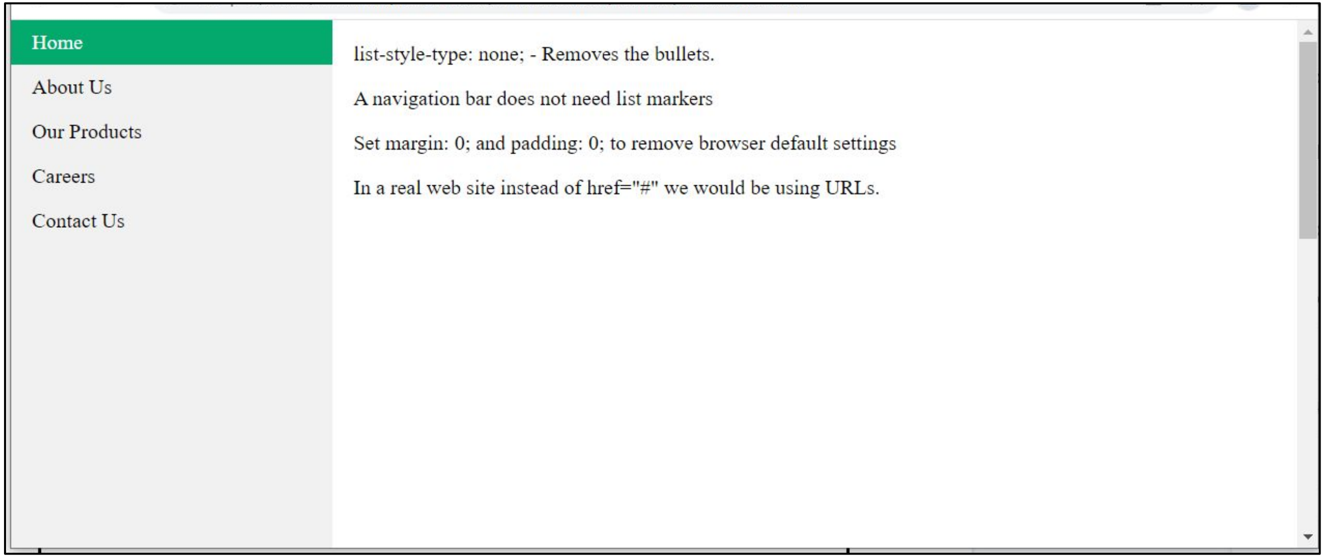
- Vertical Navigation Bar
- Horizontal Navigation Bar

# Vertical Navigation Bar

Styles applied to the same code and the difference in output:



```
body {margin: 0;}

ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  width: 25%;
  background-color: □#f1f1f1;
  position: fixed;
  height: 100%;
  overflow: auto;
}

li a {
  display: block;
  color: ■#000;
  padding: 8px 16px;
  text-decoration: none;
}

li a.active {
  background-color: 🟩#04AA6D;
  color: □white;
}

li a:hover:not(.active) {
  background-color: ■#555;
  color: □white;
}
```



list-style-type: none; - Removes the bullets.

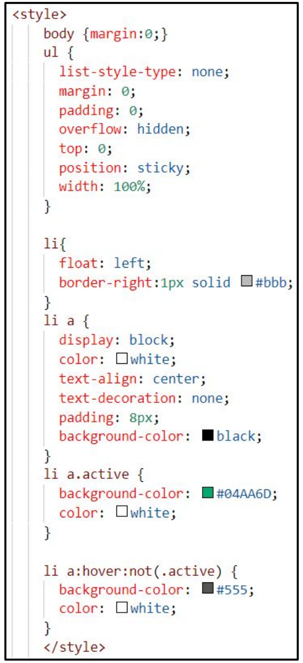
A navigation bar does not need list markers

Set margin: 0; and padding: 0; to remove browser default settings

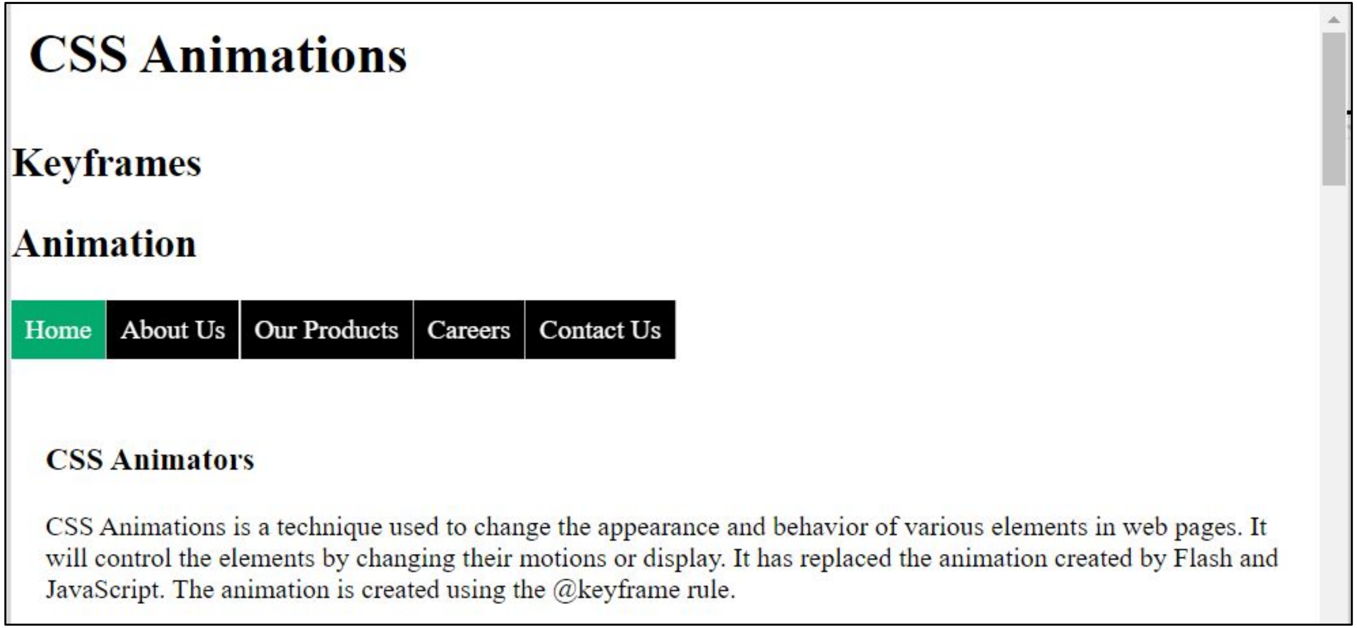
In a real web site instead of href="#" we would be using URLs.

# Horizontal Navigation Bar

Styles applied and output:



```
<style>
    body {margin:0;}
    ul {
        list-style-type: none;
        margin: 0;
        padding: 0;
        overflow: hidden;
        top: 0;
        position: sticky;
        width: 100%;
    }

    li{
        float: left;
        border-right:1px solid ▨#bbb;
    }
    li a {
        display: block;
        color: □white;
        text-align: center;
        text-decoration: none;
        padding: 8px;
        background-color: ■black;
    }
    li a.active {
        background-color: 🟩#04AA6D;
        color: □white;
    }

    li a:hover:not(.active) {
        background-color: ■#555;
        color: □white;
    }
    </style>
```



# CSS Animations

## Keyframes

## Animation

Home | About Us | Our Products | Careers | Contact Us

### CSS Animators

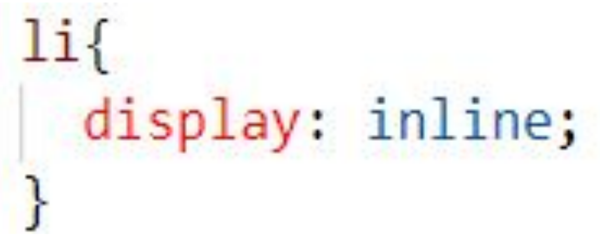
CSS Animations is a technique used to change the appearance and behavior of various elements in web pages. It will control the elements by changing their motions or display. It has replaced the animation created by Flash and JavaScript. The animation is created using the @keyframe rule.

# Horizontal Navigation Bar

- To create a horizontal navigation bar there are two ways.
- They are
  - Inline List Items
  - Float List Items

# Inline List Items

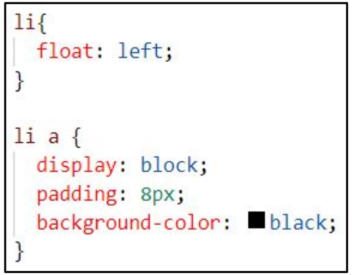
- Horizontal Navigation Bar is created by specifying <li> element as **inline**.
- The line breaks before and after each item in the list is removed to display in one line.
- <li> elements are displayed as **block** by default.



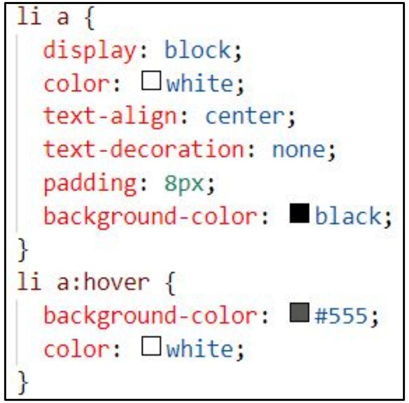
```
li{
  display: inline;
}
```

# Float List Items

- Horizontal Navigation Bar is created by specifying <li> element as ***float: left;.***
- Setting ***float: left;*** will make the block elements float next to each other.
- ***display: block;*** - We display the links as block elements which allows us to specify the width, padding, margin, height, etc.
- Setting the navigation bar **background color** to **black** and when the user moves the cursor over the menu the background color of the link has to change to **gray**.



```
li{
  float: left;
}

li a {
  display: block;
  padding: 8px;
  background-color: ■black;
}
```



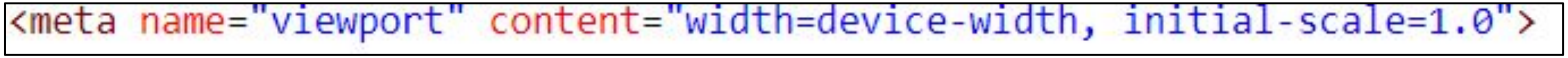
```
li a {
  display: block;
  color: □white;
  text-align: center;
  text-decoration: none;
  padding: 8px;
  background-color: ■black;
}
li a:hover {
  background-color: ■#555;
  color: □white;
}
```

# Responsive Web Design

- Web Pages can be viewed on **any device** like mobile, laptop, tablet, etc. Regardless of the device, our web page should **look good** and **easy to use** for this we go for **Responsive Web Design**.
- Responsive Web Design uses only **HTML** and **CSS** to create a web page that looks good on all devices.
- While viewing on **smaller devices**, the web page should not leave any information. Instead, the content should **fit** any device.
- To **resize, hide, shrink, enlarge or move** the content using **HTML and CSS** to look good on any screen is called **Responsive Web Design**.

# View Port

- The **Visible area** of a web page is called **Viewport**.
- The viewport differs from device to device. **Mobile phones** have **smaller** viewports when compared to computers.
- Initially, Web pages are designed **only** for **computers** by having **static design** and **fixed size**.
- **Fixed-size** web pages are **too large** for the viewport of **mobiles and tablets**.
- **<meta>** tag in **HTML5** helped web designers by taking control over the **viewport**.



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

# View Port

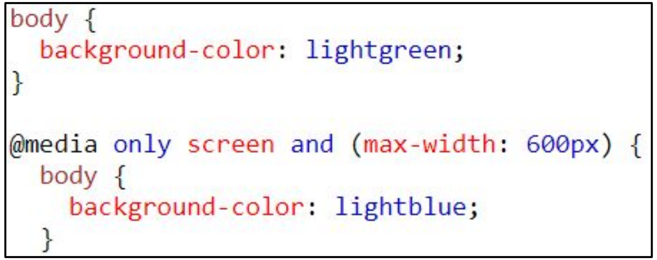
- Elements with **large fixed widths** should not be used.
- The content **should not** rely on particular **viewport width** to render well.
- Page elements **should not** have **large absolute CSS width**. Instead, use **relative** width values such as **width: 100%. Large absolute values** cause the element to fall **outside** the viewport.

# Ingredients of Responsive Web Design

- Media Queries
- Fluid Grids
- Flexible Images

# Media Queries

- Media Query is a technique in **CSS3**
- If a condition is true, the **@media** rule is used to include a block of CSS properties.
- **Media queries** are used to make images smaller on all mobiles but larger on other devices like ipads, desktops, etc.
- If the browser window is within 600px the background color is light blue or else the background color will be light green.



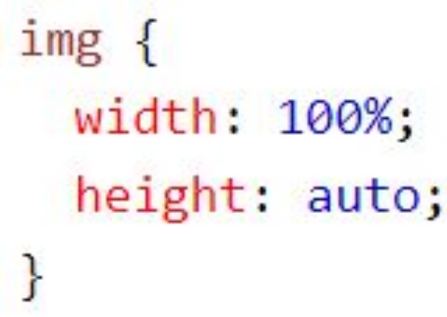
```
body {
  background-color: lightgreen;
}

@media only screen and (max-width: 600px) {
  body {
    background-color: lightblue;
  }
```

# Flexible Grid / Fluid Grid

The **rearrangement of columns** themselves to fit the screen size is possible only by **Flexible Grid**. These Flexible Grids are created using CSS.

# Flexible Visuals

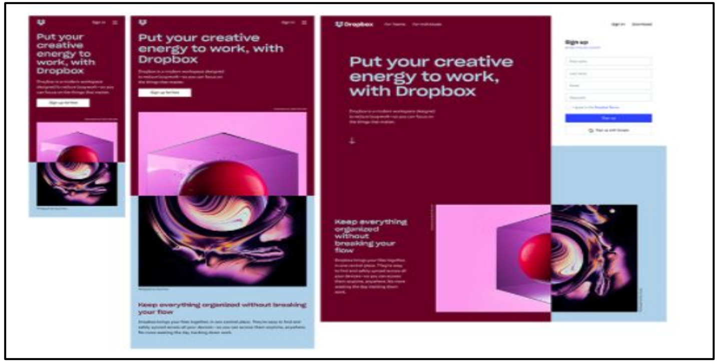
Images will scale up and down if **width: 100%;** and **height: auto;** is set.



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

# Responsive Web Design - Examples

- **Drop Box:** DropBox uses **fluid grid** and **flexible visuals** to achieve this responsive website.
- **Dribbble:** Dribbble website uses a **flexible grid** that condenses from 5 columns on computers to 2 columns on mobiles and tablets.



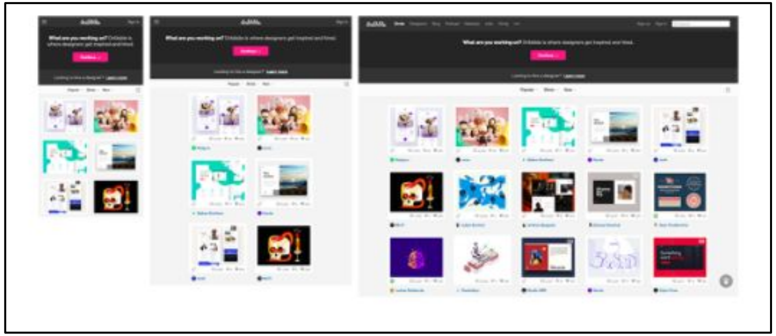
Put your creative energy to work, with Dropbox

Put your creative energy to work, with Dropbox

Put your creative energy to work, with Dropbox

Keep everything organized without breaking your flow

Keep everything organized without breaking your flow



What are you working on? Dribbble is where designers get inspired and hired.

Navigation bar

Vertical Navigation bar

Horizontal Navigation bar

Responsive Web Design

**RWD - Examples**