

# 21. CSS Display, Max-width, Position & Overflow

In this lesson you will learn about the display property in CSS and using max-width layout in a CSS. The display property is the most important CSS property for controlling layout. The display property specifies if/how an element is displayed. Every HTML element has a default display value depending on what type of element it is. The default display value for most elements is block or inline. The max-width property defines the maximum width of an element. If the content is larger than the maximum width, it will automatically change the height of the element. If the content is smaller than the maximum width, the max-width property has no effect.

The following activities are shown in **ELSIE**.

**ACTIVITY 21A01a.** A block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can). Here are the examples of block-level elements: <div>, <h1> - <h6>, <p>, <form>, <header>, <footer>, and <section>.

This is a block <div> element

inside a paragraph.

Examples of block-level elements: <div>, <h1> - <h6>, <p>, <form>, <header>, <footer>, <section>

An inline element does not start on a new line and only takes up as much width as necessary.

This is an inline <span> element inside a paragraph.

Examples of inline elements: <span>, <a>, <img>

display: none; is commonly used with JavaScript to hide and show elements without deleting and recreating them. Take a look at our last example on this page if you want to know how this can be achieved. The <script> element uses display: none; as default.

As mentioned, every element has a default display value. However, you can override this. Changing an inline element to a block element, or vice versa, can be useful for making the page look a specific way, and still follow the web standards.

A common example is making inline <li> elements for horizontal menus:

```
<html>
<head>
  <title>21A01a</title>
  <style>
    li {
      display: inline;
    }
    body {
      background-color: pink;
      color: red;
      font-size: 30px;
    }
  </style>
</head>
<body>
  <p>Display a list of links as a horizontal menu:</p>
  <ul>
    <li><a href="https://www.w3schools.com/html/default.asp" target="_blank">HTML</a></li>
    <li><a href="https://www.w3schools.com/css/default.asp" target="_blank">CSS</a></li>
    <li><a href="https://www.w3schools.com/js/default.asp" target="_blank">JavaScript</a></li>
  </ul>
</body>
</html>
```

Should look like this:  
Display a list of links as a horizontal menu:

[HTML](https://www.w3schools.com/html/default.asp) [CSS](https://www.w3schools.com/css/default.asp) [JavaScript](https://www.w3schools.com/js/default.asp)

**ACTIVITY 21A01b.** Setting the display property of an element only changes how the element is displayed, NOT what kind of element it is. So, an inline element with display: block; is not allowed to have other block elements inside it. The following example displays <span> elements as block elements:

```
<html>
<head>
  <title>21A01b</title>
  <style>
    span {
      display: block;
    }
    body {
      background-color: pink;
      color: red;
      font-size: 50px;
    }
  </style>
</head>
<body>
  <span>no rain</span>
  <span>no flowers</span>
</body>
</html>
```

```
</head>
<body>
  <span>no rain</span>
  <span>no flowers</span>
</body>
</html>
```

Should look like this:

no rain  
no flowers

**ACTIVITY 21A01c.** The element generates a block element box, generating line breaks both before and after the element when in the normal flow. The following example displays <a> elements as block elements:

```
<html>
<head>
  <title>21A01c</title>
  <style>
    a {
      display: block;
    }
    body {
      background-color: pink;
      color: red;
      font-size: 30px;
    }
  </style>
</head>
<body>
  <p>Display links as block elements:</p>
  <a href="https://www.w3schools.com/html/default.asp" target="_blank">HTML</a>
  <a href="https://www.w3schools.com/css/default.asp" target="_blank">CSS</a>
  <a href="https://www.w3schools.com/js/default.asp" target="_blank">JavaScript</a>
</body>
</html>
```

Should look like this:

Display links as block elements:

[HTML](https://www.w3schools.com/html/default.asp)  
[CSS](https://www.w3schools.com/css/default.asp)  
[JavaScript](https://www.w3schools.com/js/default.asp)

**ACTIVITY 21A02a.** Hiding an element can be done by setting the display property to none. The element will be hidden, and the page will be displayed as if the element is not there:

```
<html>
<head>
  <title>21A02a</title>
  <style>
    h1.hidden {
      display: none;
    }
    body {
      background-color: pink;
      color: red;
      font-size: 30px;
    }
  </style>
</head>
<body>
  <h1>This is a visible heading</h1>
  <h1 class="hidden">This a hidden heading</h1>
  <p>no rain, no flowers</p>
</body>
</html>
```

Should look like this. Notice that the h1 element with display: none; does not take up any space.

This is a visible heading

no rain, no flowers

**ACTIVITY 21A02b.** The visibility: hidden; also hides an element. However, the element will still take up the same space as before. The element will be hidden, but still affect the layout:

```
<html>
<head>
  <title>21A02b</title>
  <style>
    h1.hidden {
      visibility: hidden;
    }
    body {
      background-color: pink;
      color: red;
      font-size: 30px;
    }
  </style>
</head>
<body>
  <h1>This is a visible heading</h1>
  <h1 class="hidden">This a hidden heading</h1>
  <p>no rain, no flowers</p>
</body>
</html>
```

Looks like this:

This is a visible heading

no rain, no flowers

Notice that the hidden heading still takes up space.

**Using width, max-width and margin: auto;**

**ACTIVITY 21A03.** As mentioned above; a block-level element always takes up the full width available (stretches out to the left and right as far as it can). Setting the width of a block-level element will prevent it from stretching out to the edges of its container. Then, you can set the margins to auto, to horizontally center the element within its container. The element will take up the specified width, and the remaining space will be split equally between the two margins:

This <div> element has a width of 500px, and margin set to auto

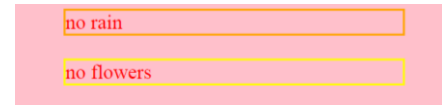
The problem with the <div> above occurs when the browser window is smaller than the width of the element. The browser then adds a horizontal scrollbar to the page. Using max-width instead, in this situation, will improve the browser's handling of small windows. This is important when making a site usable on small devices:

Resize the browser window to less than 500px wide, to see the difference between the two divs! Here is an example of the two divs above:

```
<html>
  <head>
    <title>21A03</title>
    <style>
      div.ex1 {
        width: 500px;
        margin: auto;
        border: 3px solid orange;
      }
      div.ex2 {
        max-width: 500px;
        margin: auto;
        border: 3px solid yellow;
      }
    </style>
  </head>
  <body>
    <div class="ex1">no rain</div>
    <br />
    <div class="ex2">no flowers</div>
  </body>
</html>
```

```
body {
  background-color: pink;
  color: red;
  font-size: 30px;
}
</style>
</head>
<body>
  <div class="ex1">no rain</div>
  <br />
  <div class="ex2">no flowers</div>
</body>
</html>
```

Looks like this:



Drag the browser window to smaller than 500px wide, to see the difference between the two divs

## Vocabulary

**DISPLAY** - Specifies how an element should be displayed.

**VISIBILITY** - Specifies whether or not an element should be visible.

**BLOCK LEVEL ELEMENTS** - It always starts on a new line and takes up the full width available.

**INLINE ELEMENTS** - It doesn't start on a new line and only takes up as much width as necessary.

**DISPLAY: NONE;** - It is commonly used with JavaScript to hide and show elements without deleting and recreating them.

**VISIBILITY: HIDDEN;** - It hides an elements and can be done by setting the display property to none.

Next, you will learn about CSS position and overflow property. The position property specifies the type of positioning method used for an element (static, relative, fixed, absolute or sticky). The top, right, bottom, and left properties determine the final location of positioned elements.

The CSS overflow property controls what happens to content that is too big to fit into an area. The options include clipping, showing scrollbars, or displaying the content flowing out of its container into the surrounding area. In order for overflow to have an effect, the block-level container must have either a set height (height or max-height) or white-space set to nowrap.

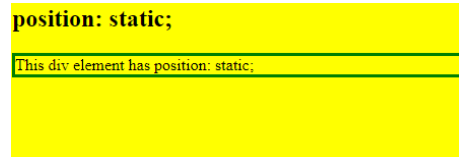
**ACTIVITY 21A04.** There are five different position values: static, relative, fixed, absolute, sticky. Elements are then positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first. They also work differently depending on the position value.

HTML elements are positioned static by default. Static positioned elements are not affected by the top, bottom, left, and right properties. An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page:

Here is the CSS that is used:

```
<html>
  <head>
    <title>21A04</title>
    <style>
      div.static {
        position: static;
        border: 3px solid green;
      }
      body {
        background-color: yellow;
      }
    </style>
  </head>
  <body>
    <h2>position: static;</h2>
    <div class="static">
      This div element has position: static;
    </div>
  </body>
</html>
```

Looks like this:



**ACTIVITY 21A05a.** An element with position: relative; is positioned relative to its normal position. Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap left by the element.

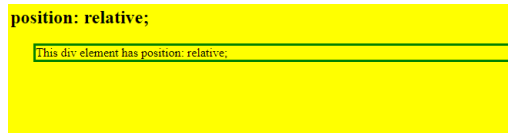
The content of this page is the intellectual and sole property of Mr. W. Liem (Author). Any addition, modification, deletion, copying and/or printing of this page or portion of this page without the written permission from the Author is strictly prohibited. Viewing of this page for educational purposes requires the Author's consent. This page

```

<html>
<head>
  <title>21A05a</title>
  <style>
    div.relative {
      position: relative;
      left: 30px;
      border: 3px solid green;
    }
    body {
      background-color: yellow;
    }
  </style>
</head>
<body>
  <h2>position: relative;</h2>
  <div class="relative">
    This div element has position: relative;
  </div>
</body>
</html>

```

Looks like this:



Try to scroll it too.

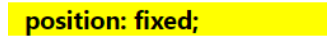
**ACTIVITY 21A05b.** An element with `position: fixed;` is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The `top`, `right`, `bottom`, and `left` properties are used to position the element. A fixed element does not leave a gap in the page where it would normally have been located.

```

<html>
<head>
  <title>21A05b</title>
  <style>
    div.fixed {
      position: fixed;
      bottom: 0;
      right: 0;
      width: 300px;
      border: 3px solid green;
    }
    body {
      background-color: yellow;
    }
  </style>
</head>
<body>
  <h2>position: fixed;</h2>
  <div class="fixed">
    This div element has position: fixed;
  </div>
</body>
</html>

```

Looks like this:



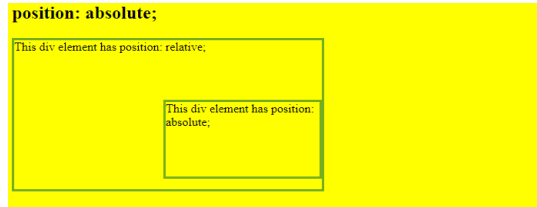
**ACTIVITY 21A05c.** An element with `position: absolute;` is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed). However, if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling. A "positioned" element is one whose position is anything except static.

```

<html>
<head>
  <title>21A05c</title>
  <style>
    div.relative {
      position: relative;
      width: 400px;
      height: 200px;
      border: 3px solid #73AD21;
    }
    div.absolute {
      position: absolute;
      top: 80px;
      right: 0;
      width: 200px;
      height: 100px;
      border: 3px solid #73AD21;
    }
    body {
      background-color: yellow;
    }
  </style>
</head>
<body>
  <h2>position: absolute;</h2>
  <div class="relative">This div element has position: relative;
  <div class="absolute">This div element his position: absolute;</div>
</body>
</html>

```

Looks like this:

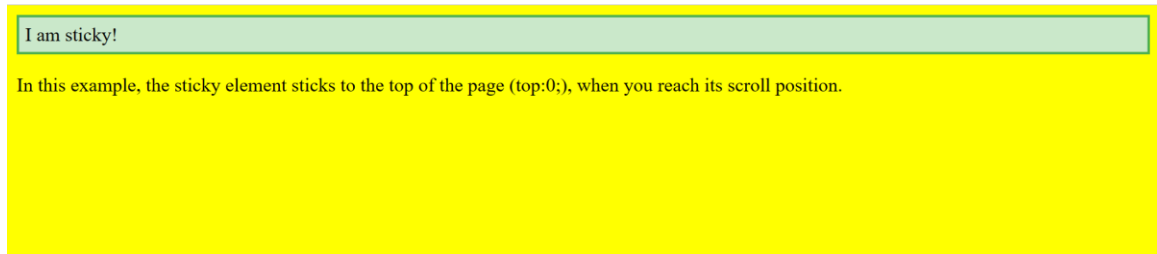


**ACTIVITY 21A05d.** An element with `position: sticky;` is positioned based on the user's scroll position. A sticky element toggles between relative and fixed, depending on the scroll position. It is positioned relative until a given offset position is met in the viewport - then it "sticks" in place (like `position: fixed;`).

In this example, the sticky element sticks to the top of the page (`top: 0;`), when you reach its scroll position.

```
<html>
<head>
  <title>21A05d</title>
  <style>
    div.sticky {
      position: -webkit-sticky;
      position: sticky;
      top: 0;
      padding: 5px;
      background-color: #cae8ca;
      border: 2px solid #4caf50;
    }
    body {
      background-color: yellow;
    }
  </style>
</head>
<body>
  <div class="sticky">I am sticky!</div>
  <p>In this example, the sticky element sticks to the top of the page (top:0;), when you
  reach its scroll position.</p>
  <div style="padding-bottom:2000px;"></div>
</body>
</html>
```

Looks like this:

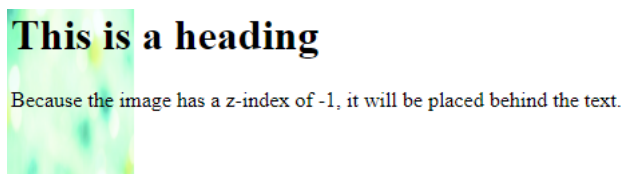


Try to scroll inside this frame to understand how sticky positioning works.

**ACTIVITY 21A06.** When elements are positioned, they can overlap other elements. The `z-index` property specifies the stack order of an element (which element should be placed in front of, or behind, the others). An element can have a positive or negative stack order.

```
<html>
<head>
  <title>21A06</title>
  <style>
    img {
      position: absolute;
      left: 0px;
      top: 0px;
      z-index: -1;
    }
  </style>
</head>
<body>
  <h1>This is a heading</h1>
  
  <p>Because the image has a z-index of -1, it will be place behind the text.</p>
</body>
</html>
```

Looks like this:



An element with greater stack order is always in front of an element with a lower stack order. If two positioned elements overlap without a z-index specified, the element positioned last in the HTML code will be shown on top.

**ACTIVITY 21A07a.** The overflow property specifies whether to clip the content or to add scrollbars when the content of an element is too big to fit in the specified area. The overflow property has the following values: visible, hidden, scroll, and auto. The overflow property only works for block elements with a specified height.

By default, the overflow is visible, meaning that it is not clipped and it renders outside the element's box. You can use the overflow property when you want to have better control of the layout. The overflow property specifies what happens if content overflows an element's box.

```
<html>
<head>
  <title>21A07a</title>
  <style>
    div {
      background-color: pink;
      width: 200px;
      height: 50px;
      border: 1px dotted black;
      overflow: visible;
    }
  </style>
</head>
<body>
  <h2>CSS Overflow</h2>
  <div>You can use the overflow property
  when you want to have better control of
  the layout. The overflow property
  specifies what happens if content
  overflows an element's box.</div>
</body>
</html>
```

Looks like this:

## CSS Overflow

You can use the overflow property when you want to have better control of the layout. The overflow property specifies what happens if content overflows an element's box.

**ACTIVITY 21A07b.** Setting the value to scroll, the overflow is clipped and a scrollbar is added to scroll inside the box. Note that this will add a scrollbar both horizontally and vertically (even if you do not need it):

```
<style>
  div {
    background-color: pink;
    width: 200px;
    height: 100px;
    border: 1px dotted black;
    overflow: scroll;
  }
</style>
```

## CSS Overflow

You can use the overflow property when you want to have better control of the layout. The overflow property specifies what

**ACTIVITY 21A07c.** The auto value is similar to scroll, but it adds scrollbars only when necessary:

```
<style>
  div {
    background-color: pink;
    width: 200px;
    height: 50px;
    border: 1px dotted black;
    overflow: auto;
  }
</style>
```

## CSS Overflow

You can use the overflow property when you want to have better control of the

**ACTIVITY 21A08.** The overflow and overflow-y properties specify whether to change the overflow of content just horizontally or vertically (or both): overflow-x specifies what to do with the left/right edges of the content and overflow-y specifies what to do with the top/bottom edges of the content.

```

<style>
  div {
    background-color: pink;
    width: 200px;
    height: 50px;
    border: 1px dotted black;
    overflow-x: hidden;
    overflow-y: scroll;
  }
</style>

```

## CSS Overflow

You can use the overflow property when you want to have better control of the

### Vocabulary

**POSITION PROPERTY** - Specifies the type of positioning method used for an element.

**POSITION: STATIC** - It is positioned according to the normal flow of the page.

**POSITION: RELATIVE** - It is positioned relative to its normal position.

**POSITION: FIXED** - It is positioned relative to the viewport.

**POSITION: ABSOLUTE** - It is positioned relative to the nearest positioned ancestor.

**POSITION: STICKY** - It is positioned based on the user's scroll position.

**STICKY ELEMENT** - Toggles between relative and fixed, depending on the scroll position.

**Z- INDEX** - Specifies the stack order of an element.

**OVERFLOW PROPERTY** - Controls what happens to content that is too big to fit into an area.

**VISIBLE VALUE** - The overflow is not clipped and the content renders outside the element's box.

**HIDDEN VALUE** - The overflow is clipped and the rest of the content will be invisible.

**SCROLL VALUE** - The overflow is clipped and a scrollbar is added to see the rest of the content.

**AUTO VALUE** - Similar to scroll, but it adds scrollbars only when necessary.

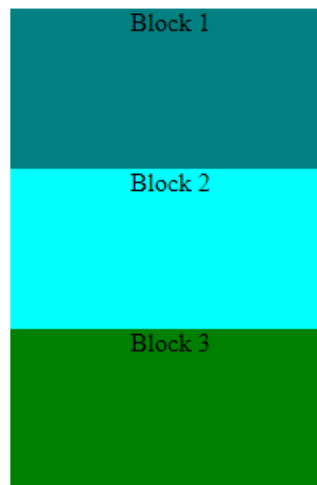
**OVERFLOW-X** - Specifies what to do with left/right edges of the content.

**OVERFLOW-Y** - Specifies what to do with top/bottom edges of the content.

### Exercises:

#### 21E01 BLOCKS

Create a web page with the title **21E01**. In the body of the webpage, create 3 blocks. First block should have a **teal** background the second one should have **cyan** background and the third one should have **green** background. Every block should have **100px** height, **200px** width and use **block** as your display. Everything inside **<body>** should have **50px** margin and align it to the **center**. Save it as **21E01**



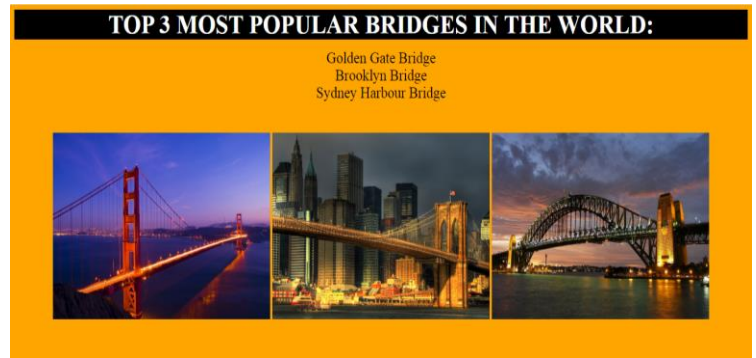
#### 21E02 MAN IN THE RAIN

Make up a webpage that looks like the picture below. Title it **21E02**. Use **<p>** for the text below. It should have **20%** max-width, a **white** text color, **silver** background and **30px** font-size. Use **rain.jpg** as your background image. Margin should be **200px**. Save it as **21E02**



### 21E03 BRIDGES PART 1

Create a web page with the title **21E03**. In the body of the webpage add the pictures of the 3 most popular bridges in the world. Use **golden.jpg**, **brooklyn.jpg**, and **sydney.jpg** as your visual representation of your bridges. All images should have **300px** height, **400px** width and **inline-block** display each. Your heading should have **42px** size **white** text color and **black** background. Add **bold** as your font-weight too. Your texts below the heading should have **25px** size. Add **50px** margin before the images. Align everything in **center** and add **orange** background. Save it as **21E03**



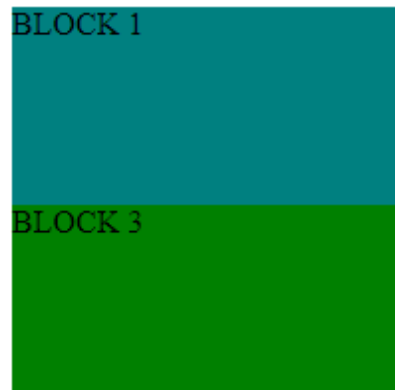
### 21E04 BRIDGES PART 2

Create a web page with the title **21E04**. Open **21E03** as your guide for this exercise. Remove all the text and change the size of the images. Every image should have **200px** height and **300px** width. Change the inline display to **block** display. Save it as **21E04**



### 21E05 NO DISPLAY

Create a web page with the title **21E05**. Open **21E01** as your reference. For the second block **remove** the display but keep all the style that you put in there. Save it as **21E05**



### 21E06 BILLIE EILISH

Create a web page with the title **21E06**. In the body of the webpage add **award.jpg** and change its size to **200px** height and **300px** width. Use **<div>** for the text below the image. This should have **250px** max-width, **#8ebf42** background and **50px** size. Add **yellow** background to the body. Save it as **21E06**



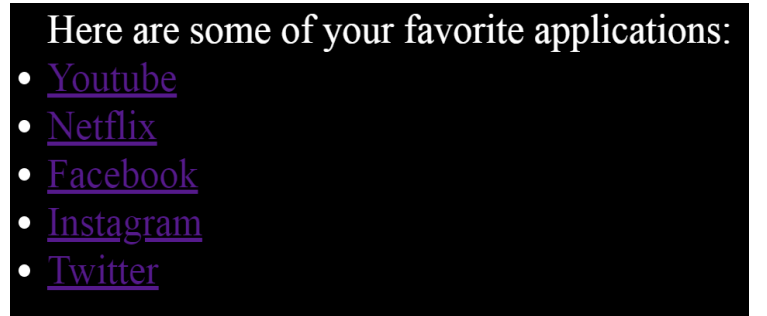
### 21E07 BOOK

Create a web page with the title 21E07. Create two <div> classes. First class should have a purple background and 300px width. Second class should have a pink background, 100% width and 150px max-width. Apply these classes to your webpage separately. Then add book.jpg that should have 100px height and 200px width. Background color should be blue. Save it as 21E07



### 21E08 FAVORITE APPLICATIONS

Create a web page with the title 21E08. In the body of the webpage, use <ul> to create an unordered list about your most favourite applications. Make sure to link it to its real website as reference. Use block as your display and background color should be black. Text color should be white and has 50px font size. Save it as 21E08



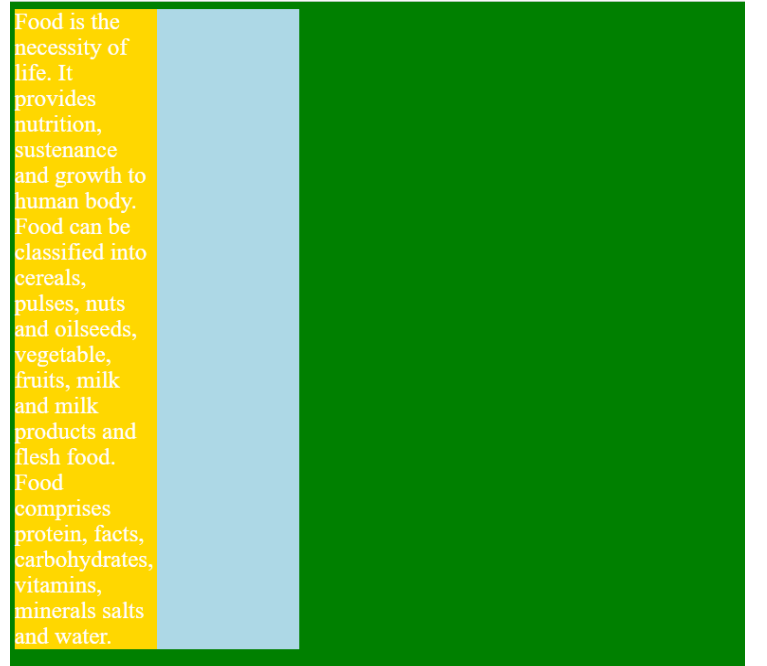
### 21E09 DON'T WORRY BE HAPPY

Make up a webpage that looks like the picture below. Title it 21E09. In the body of the webpage create a class that has 200px height and 200px width. In the same class, change the background color to black and use inline as your display. It should also have 50px margin. Use bg.jpg as your background image. Text color should be white and font size should be 50px. Save it as 21E09a



### 21E10 FOODS FOR LIFE

Create a web page with the title 21E10. In the body of the webpage create a class that has a lightblue background color and 300px width. Create another class that has gold background and 100% width and 150px max-width. Apply the two classes in the paragraph about foods. Background color should be green, text color should be white and it should have 25px text size. Save it as 21E10

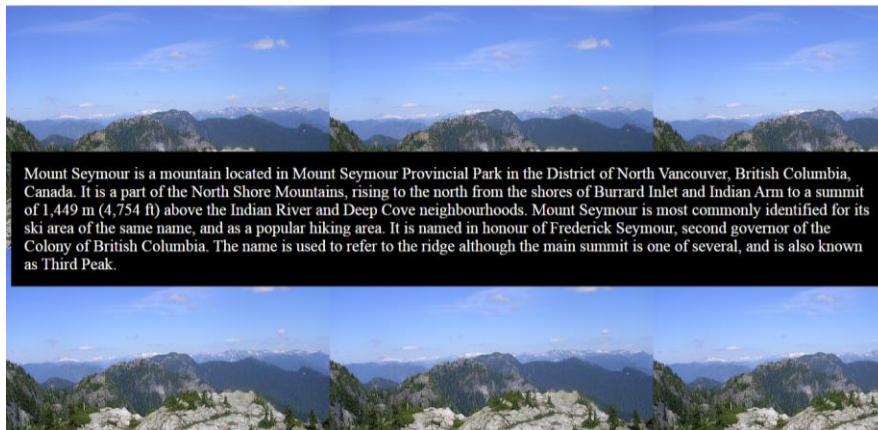


21E11 WRECK IT RALPH Create a web page with the title 21E11. In the body of the webpage, create a class that has 1px dotted green border, orange background color, 20px padding, pink text color, 20px margin and use relative as your position. Create another class for the entire body. Use flex as your display, and all items should be align center. Font size should be 30px and use poster.jpg as your background image. Then copy the paragraph that is stated below. Make the html and body have 100% height. Save it as 21E11 and it should look like this:



Wreck-It Ralph is a 2012 American 3D computer-animated comedy film produced by Walt Disney Animation Studios and released by Walt Disney Pictures. It is the 52nd Disney animated feature film. The film was directed by Rich Moore in his directorial debut, who also directed episodes of The Simpsons and Futurama, and the screenplay was written by Phil Johnston and Jennifer Lee from a story by Moore, Johnston, and Jim Reardon. John Lasseter served as the executive producer. The film features the voices of John C. Reilly, Sarah Silverman, Jack McBrayer and Jane Lynch and tells the story of the eponymous arcade game villain who rebels against his "bad-guy" role and dreams of becoming a hero.

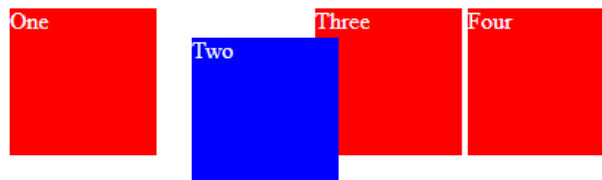
21E12 MT SEYMOUR Make up a webpage that looks like the picture below. Title it 21E12. In the body of the webpage create a class that has black background color, 20px padding, white text color, and use absolute as your position. Make it 31% from the top. Create another class for the entire body. Use flex as your display, and all items should be align center. Font size should be 25px and use seymour.jpg as your background image. Then copy the paragraph that is stated below. Save it as 21E12



Mount Seymour is a mountain located in Mount Seymour Provincial Park in the District of North Vancouver, British Columbia, Canada. It is a part of the North Shore Mountains, rising to the north from the shores of Burrard Inlet and Indian Arm to a summit of 1,449 m (4,754 ft) above the Indian River and Deep Cove neighbourhoods. Mount Seymour is most commonly identified for its ski area of the same name, and as a popular hiking area. It is named in honour of Frederick Seymour, second governor of the Colony of British Columbia. The name is used to refer to the ridge although the main summit is one of several, and is also known as Third Peak.

21E13 LETTERS, NUMBERS, SYMBOLS Create a unique and imaginative web page with the title 21E13. In the body of the webpage, create a class that has letters, numbers and symbols. Save it as 21E13

21E14 BOXES Make up a webpage that looks like the picture below. Title it 21E14. In the body of the webpage you will create a bunch of boxes! One, three and four boxes should have 100px width and height, a red background, a white text color and inline-block display. The two box should have relative position. Top and left as 20px and a blue background. Save it as 21E14



21E15 SPANISH INFLUENZA Make up a webpage that looks like the picture below. Title it 21E15. In the body of the webpage use <h1> for your heading, and make the margin 0. You will create three sets of <p> that should have aqua background, 3px solid blue border, 10px padding and 10px margin. In the last box, all forms of time should be style using 1px solid black border and red background. Everything inside <body> should have 500px width and 0 auto margin.

The first paragraph that is covered:  
*"I am a basic block level element. My adjacent block level elements sit on new lines below me."*

The paragraph over the first paragraph should have 30px top and left absolute position and should have a yellow background color. Save it as 21E15

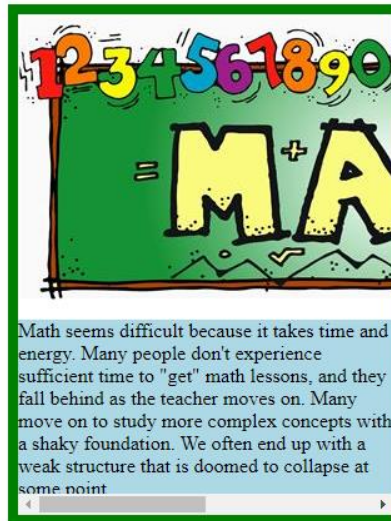
# SPANISH FLU

The Spanish flu, also known as the 1918 flu pandemic or La Pesadilla (Spanish for "The Nightmare"), was an unusually deadly influenza pandemic. Lasting from January 1918 to December 1920, it infected 500 million people

The death toll is estimated to have been anywhere from 17 million to 50 million, and possibly as high as 100 million, making it one of the deadliest pandemics in human history, behind the Black Death.

"The 1918 has gone: a year momentous as the termination of the most cruel war in the annals of the human race; a year which marked, the end at least for a time, of man's destruction of man; unfortunately a year in which developed a most fatal infectious disease causing the death of hundreds of thousands of human beings. Medical science for four and one-half years devoted itself to putting men on the firing line and keeping them there. Now it must turn with its whole might to combating the greatest enemy of all--infectious disease,"  
10/28/1918

21E16 MATH Make up a webpage that looks like the picture below. Title it 21E16. In the body of the webpage use `math.jpg` and create a `<p>` class to style the paragraph. Add some `lightblue` background to it. Next step is to add a `<div>` class to style and combine the picture and the paragraph. Add `8px solid green` border, `300px` width, and `400px` height. Your `overflow-y` should be `hidden` and your `overflow-x` should be `scroll`. Save it as `21E16`



21E17 INSTAGRAM Make up a webpage that looks like the picture below. Title it 21E17. In the body of the webpage insert `ig.png` and style it by adding `10px` margin-right, `100px` height and width. Also, float the image to the `left`. All text should have `30px` size, a pink background, and overflow should be `hidden`. Don't forget to set up your margin-right to `0`. Overall, the body should have an `orange` background color. Save it as `21E17`



Instagram is an American photo and video-sharing social networking service owned by Facebook, Inc. It was created by Kevin Systrom and Mike Krieger, and launched in October 2010 on iOS.

A version for Android devices was released in April 2012, followed by a feature-limited website interface in November 2012, a Fire OS app on June 15, 2014 and an app for Windows 10 tablets and computers in October 2016.

The app allows users to upload photos and videos, which can be edited with filters and organized with tags and location information. Posts can be shared publicly or with pre-approved followers.

Users can browse other users' content by tags and locations, and view trending content. Users can like photos and follow other users to add their content to a feed.

21E18 CATS Make up a webpage that looks like the picture below. Title it 21E18. In the body of the webpage use `<h1>` for your heading. Insert your image `cat.jpg` and change its width to `120px`. All your text should have `rgb(230, 230, 230)` background color, `70%` width, `180px` height, `overflow-y` should be `auto` and `overflow-x` should be `hidden`. Save it as `21E18`

Text:

The cat (*Felis catus*), also known as the domestic cat or housecat to distinguish it from other felines and felids, is a small domesticated carnivorous mammal that is valued by humans for its companionship and ability to hunt vermin and household pests. Cats have been associated with humans for at least 9,500 years, and are currently the most popular pet in the world. Owing to their close association with humans, cats are now found almost everywhere on Earth.

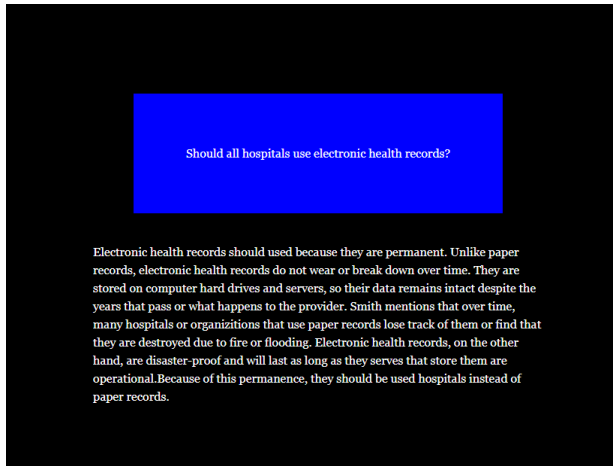
## All About Cats

### Official Info on Cats



The cat (*Felis catus*), also known as the domestic cat or housecat to distinguish it from other felines and felids, is a small domesticated carnivorous mammal that is valued by humans for its companionship and ability to hunt vermin and household pests. Cats have been associated with humans for at least 9,500 years, and are currently the most popular pet in the world. Owing to their close association with

**21E19 ELECTRONIC RECORDS** Make up a webpage that looks like the picture below. Title it **21E19**. Create a `<div>` class to style your heading. It should have a **white** text color, **70px** padding, **blue** background, **50px** margin-top and use **relative** as your position. The rest of text should have **0 auto** margin, **600px** max-width, **40px** margin-top and **1.5** line-height. Everything inside `<body>` should have **Georgia** and **serif** for the font-family. It should have a **black** background, **white** text color. All text should be in **center** and flex-direction should be **column**. Use **flex** for your display. Save it as **21E19**.



**21E20 PRODUCTS** Make up a webpage that looks like the picture below. Title it **21E20**. All text in the webpage should have **white** text color. Create your header "Products" and style it by adding **hotpink** background, **flex** display, **250px** min-height, **90px** font-size and align items to the **center**. Add a justify-content too that should be in **center**. Make the body have 0 margin. Create 2 separate lists. Nav should have **#333** background, **5px** padding and **100%** width. Your second list should have a **sticky** position and top should be **0px**. Your `<ul>`, and `<li>` should have a **20px** margin-left. Lastly, your `<nav>` and `<ul>` should have **flex** display. Save it as **21E20**

