

# Data visualization

COSC 480B

Reyan Ahmed

[rahmed1@colgate.edu](mailto:rahmed1@colgate.edu)

# Lecture 4

Advanced CSS

# Rounded Corners

```
#rcorners1 {  
  border-radius: 25px;  
  background: #73AD21;  
  padding: 20px;  
  width: 200px;  
  height: 150px;  
}
```



# Border Images

```
#borderimg {  
  border: 10px solid transparent;  
  padding: 15px;  
  border-image: url(border.png) 30 round;  
}
```

border-image: url(border.png) 30 round;



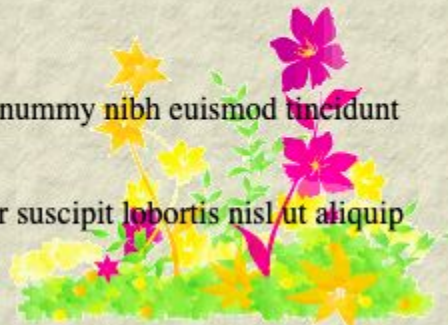
# Backgrounds

```
#example1 {  
  background-image: url(img_flwr.gif), url(paper.gif);  
  background-position: right bottom, left top;  
  background-repeat: no-repeat, repeat;  
  padding: 15px;  
}
```

## Lorem Ipsum Dolor

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

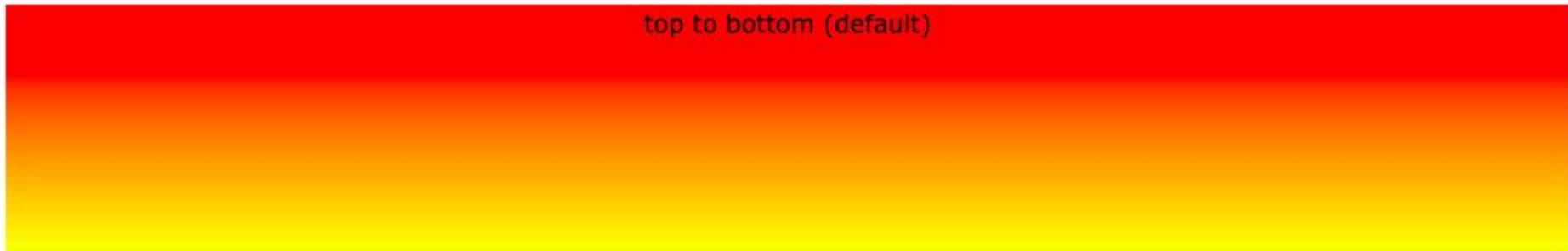
Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.



# Gradients

```
#grad {  
  background-image: linear-gradient(red, yellow);  
}
```

top to bottom (default)



# Shadow Effect

```
h1 {  
  text-shadow: 2px 2px;  
}  
</style>  
</head>  
<body>  
  
<h1>Text-shadow effect!</h1>
```

**Text-shadow effect!**

# Text Effects

```
p.test1 {  
  white-space: nowrap;  
  width: 200px;  
  border: 1px solid #000000;  
  overflow: hidden;  
  text-overflow: clip;  
}
```

```
p.test2 {  
  white-space: nowrap;  
  width: 200px;  
  border: 1px solid #000000;  
  overflow: hidden;  
  text-overflow: ellipsis;  
}
```

**text-overflow: clip:**

This is some long text that will

**text-overflow: ellipsis:**

This is some long text that ...



# Translate

```
div {  
  width: 300px;  
  height: 100px;  
  background-color: yellow;  
  border: 1px solid black;  
  -ms-transform: translate(50px,100px); /* IE 9 */  
  transform: translate(50px,100px); /* Standard syntax */  
}
```

The `translate()` method moves an element from its current position:

This div element is moved 50 pixels to the right, and 100 pixels down from its current position.

# Rotate

```
div {  
  width: 300px;  
  height: 100px;  
  background-color: yellow;  
  border: 1px solid black;  
}
```

```
div#myDiv {  
  -ms-transform: rotate(20deg); /* IE 9 */  
  transform: rotate(20deg); /* Standard syntax */  
}
```

The rotate() method rotates an element clockwise or counter-clockwise.

This a normal div element.

This div element is rotated clockwise 20 degrees.

# Scale

The `scale()` method increases or decreases the size of an element.

```
div {  
  margin: 150px;  
  width: 200px;  
  height: 100px;  
  background-color: yellow;  
  border: 1px solid black;  
  -ms-transform: scale(2,3); /* IE 9 */  
  transform: scale(2,3); /* Standard syntax */  
}
```

This div element is two times of its original width, and three times of its original height.

# Scale along X-axis

The `scaleX()` method increases or decreases the width of an element.

```
div {  
  margin: 150px;  
  width: 200px;  
  height: 100px;  
  background-color: yellow;  
  border: 1px solid black;  
  -ms-transform: scaleX(2); /* IE 9 */  
  transform: scaleX(2); /* Standard syntax */  
}
```

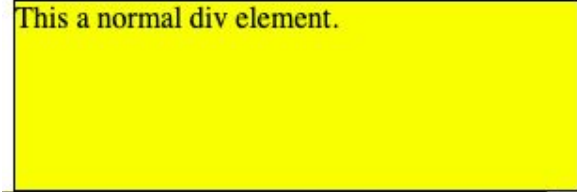
**This div element is two times  
of its original width.**

# Skew

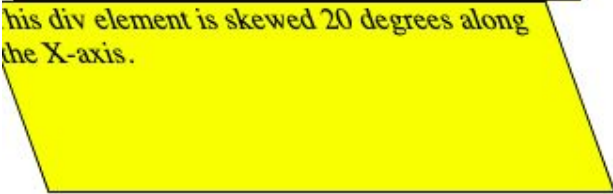
```
div {  
  width: 300px;  
  height: 100px;  
  background-color: yellow;  
  border: 1px solid black;  
}  
  
div#myDiv {  
  -ms-transform: skewX(20deg); /* IE 9 */  
  transform: skewX(20deg); /* Standard syntax */  
}
```

The skewX() method skews an element along the X-axis by the given angle.

This a normal div element.



This div element is skewed 20 degrees along the X-axis.



# Matrix

The parameters are as follow: `matrix(scaleX(),skewY(),skewX(),scaleY(),translateX(),translateY())`

```
div {
  width: 300px;
  height: 100px;
  background-color: yellow;
  border: 1px solid black;
}

div#myDiv1 {
  -ms-transform: matrix(1, -0.3, 0, 1, 0, 0); /* IE 9 */
  transform: matrix(1, -0.3, 0, 1, 0, 0); /* Standard syntax */
}

div#myDiv2 {
  -ms-transform: matrix(1, 0, 0.5, 1, 150, 0); /* IE 9 */
  transform: matrix(1, 0, 0.5, 1, 150, 0); /* Standard syntax */
}
```

# Matrix

The `matrix()` method combines all the 2D transform methods into one.

This a normal div element.

Using the `matrix()` method.

Another use of the `matrix()` method.

# RotateX

```
<style>
div {
  width: 300px;
  height: 100px;
  background-color: yellow;
  border: 1px solid black;
}

#myDiv {
  transform: rotateX(150deg);
}
</style>
</head>
<body>
```

**<h1>The rotateX() Method</h1>**

**<p>The rotateX() method rotates an element around its X-axis at a given degree.</p>**

```
<div>
This a normal div element.
</div>
```

```
<div id="myDiv">
This div element is rotated 150 degrees.
</div>
```



# RotateX

## The rotateX() Method

The rotateX() method rotates an element around its X-axis at a given degree.

This a normal div element.

This div element is rotated 120 degrees.

# 3D Transform Methods

Function	Description
<code>matrix3d</code> <code>(n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n)</code>	Defines a 3D transformation, using a 4x4 matrix of 16 values
<code>translate3d(x,y,z)</code>	Defines a 3D translation
<code>translateX(x)</code>	Defines a 3D translation, using only the value for the X-axis
<code>translateY(y)</code>	Defines a 3D translation, using only the value for the Y-axis
<code>translateZ(z)</code>	Defines a 3D translation, using only the value for the Z-axis
<code>scale3d(x,y,z)</code>	Defines a 3D scale transformation
<code>scaleX(x)</code>	Defines a 3D scale transformation by giving a value for the X-axis
<code>scaleY(y)</code>	Defines a 3D scale transformation by giving a value for the Y-axis
<code>scaleZ(z)</code>	Defines a 3D scale transformation by giving a value for the Z-axis
<code>rotate3d(x,y,z,angle)</code>	Defines a 3D rotation
<code>rotateX(angle)</code>	Defines a 3D rotation along the X-axis
<code>rotateY(angle)</code>	Defines a 3D rotation along the Y-axis
<code>rotateZ(angle)</code>	Defines a 3D rotation along the Z-axis

# Transition

```
<style>
div {
  width: 100px;
  height: 100px;
  background: red;
  transition: width 2s;
}
```

```
div:hover {
  width: 300px;
}
```

```
</style>
</head>
<body>
```

```
<h1>The transition Property</h1>
```

```
<p>Hover over the div element below, to see the transition effect:</p>
```

```
<div></div>
```

# Transition

## The transition Property

Hover over the div element below, to see the transition effect:



## The transition Property

Hover over the div element below, to see the transition effect:



# Tooltip

```
<style>
.tooltip {
  position: relative;
  display: inline-block;
  border-bottom: 1px dotted black;
}
```

```
.tooltip .tooltiptext {
  visibility: hidden;
  width: 120px;
  background-color: black;
  color: #fff;
  text-align: center;
  border-radius: 6px;
  padding: 5px 0;
```

```
/* Position the tooltip */
position: absolute;
z-index: 1;
}
```

```
.tooltip:hover .tooltiptext {
  visibility: visible;
}
```

```
</style>
<body style="text-align:center;">
```

```
<p>Move the mouse over the text below:</p>
```

```
<div class="tooltip">Hover over me
  <span class="tooltiptext">Tooltip text</span>
</div>
```

Move the mouse over the text below:

Hover over me

Move the mouse over the text below:

Hover over me

Tooltip text

# Images

```
<style>
img {
  border-radius: 8px;
}
</style>
</head>
<body>
```

```
<h2>Rounded Images</h2>
```

```
<p>Use the border-radius property to create rounded images:</p>
```

```

```

## Rounded Images

Use the border-radius property to create rounded images:



# Image reflection

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
  -webkit-box-reflect: right;
}
</style>
</head>
<body>

<h1>CSS Image Reflection</h1>
<p>Show the reflection to the right of the image:</p>


</body>
</html>
```

## CSS Image Reflection

Show the reflection to the right of the image:



# Buttons

```
<style>
.button {
  background-color: #4CAF50;
  border: none;
  color: white;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  margin: 4px 2px;
  cursor: pointer;
}
```

```
</style>
</head>
<body>
```

```
<h2>CSS Buttons</h2>
```

```
<button>Default Button</button>
<a href="#" class="button">Link Button</a>
<button class="button">Button</button>
<input type="button" class="button" value="Input Button">
```

## CSS Buttons

Default Button

Link Button

Button

Input Button



# Pagination

```
<style>
.pagination {
  display: inline-block;
}

.pagination a {
  color: black;
  padding: 8px 16px;
  text-decoration: none;
}

.pagination a.active {
  background-color: #4CAF50;
  color: white;
}

.pagination a:hover:not(.active) {background-color: #ddd;}
</style>
</head>
<body>
```

```
<h2>Active and Hoverable Pagination</h2>
<p>Move the mouse over the numbers.</p>
```

```
<div class="pagination">
  <a href="#">&laquo;</a>
  <a href="#">1</a>
  <a class="active" href="#">2</a>
  <a href="#">3</a>
  <a href="#">4</a>
  <a href="#">5</a>
  <a href="#">6</a>
  <a href="#">&raquo;</a>
</div>
```

## Active and Hoverable Pagination

Move the mouse over the numbers.

« 1 2 3 4 5 6 »

## Active and Hoverable Pagination

Move the mouse over the numbers.

« 1 2 3 4 5 6 »

# Multi-column

```
<style>
.newspaper {
  column-count: 3;
}
</style>
</head>
<body>

<div class="newspaper">
  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod
  tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis
  nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.
  Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel
  illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui
  blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber
  tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat
  facer possim assum.
</div>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip

ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit

augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum.

# Box sizing

```
<style>
.div1 {
  width: 300px;
  height: 100px;
  border: 1px solid blue;
}

.div2 {
  width: 300px;
  height: 100px;
  padding: 50px;
  border: 1px solid red;
}
</style>
</head>
<body>

<div class="div1">This div is smaller (width is 300px and height is 100px).</div>
<br>
<div class="div2">This div is bigger (width is also 300px and height is 100px).</div>
```

This div is smaller (width is 300px and height is 100px).

This div is bigger (width is also 300px and height is 100px).

# Responsive CSS



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum.

# Responsive CSS



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

```
.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%;}
```

# Responsive CSS

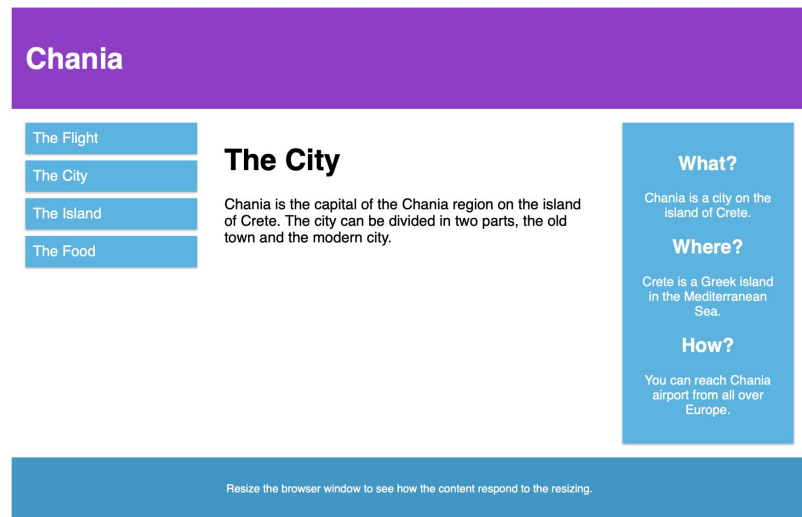
```
<div class="header">
  <h1>Chania</h1>
</div>

<div class="row">
  <div class="col-3 menu">
    <ul>
      <li>The Flight</li>
      <li>The City</li>
      <li>The Island</li>
      <li>The Food</li>
    </ul>
  </div>

  <div class="col-6">
    <h1>The City</h1>
    <p>Chania is the capital of the Chania region on the island of Crete. The city can be
divided in two parts, the old town and the modern city.</p>
  </div>

  <div class="col-3 right">
    <div class="aside">
      <h2>What?</h2>
      <p>Chania is a city on the island of Crete.</p>
      <h2>Where?</h2>
      <p>Crete is a Greek island in the Mediterranean Sea.</p>
      <h2>How?</h2>
      <p>You can reach Chania airport from all over Europe.</p>
    </div>
  </div>
</div>

<div class="footer">
  <p>Resize the browser window to see how the content respond to the resizing.</p>
</div>
```



# Responsive CSS





# Responsive CSS

```
@media only screen and (max-width: 768px) {  
  /* For mobile phones: */  
  [class*="col-"] {  
    width: 100%;  
  }  
}
```

## Chania

The Flight

The City

The Island

The Food

## The City

Chania is the capital of the Chania region on the island of Crete. The city can be divided in two parts, the old town and the modern city.

### What?

Chania is a city on the island of Crete.

### Where?

Crete is a Greek island in the Mediterranean Sea.

### How?

You can reach Chania airport from all over Europe.

Resize the browser window to see how the content respond to the resizing.

# Grid

```
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto;
  background-color: #2196F3;
  padding: 10px;
}
.grid-item {
  background-color: rgba(255, 255, 255, 0.8);
  border: 1px solid rgba(0, 0, 0, 0.8);
  padding: 20px;
  font-size: 30px;
  text-align: center;
}
</style>
</head>
<body>
```

```
<h1>Grid Elements</h1>
```

<p>A Grid Layout must have a parent element with the `<em>display</em>` property set to `<em>grid</em>` or `<em>inline-grid</em>`.</p>

<p>Direct child element(s) of the grid container automatically becomes grid items.</p>

```
<div class="grid-container">
  <div class="grid-item">1</div>
  <div class="grid-item">2</div>
  <div class="grid-item">3</div>
  <div class="grid-item">4</div>
  <div class="grid-item">5</div>
  <div class="grid-item">6</div>
  <div class="grid-item">7</div>
  <div class="grid-item">8</div>
  <div class="grid-item">9</div>
</div>
```

## Grid Elements

A Grid Layout must have a parent element with the *display* property set to *grid* or *inline-grid*.

Direct child element(s) of the grid container automatically becomes grid items.

1	2	3
4	5	6
7	8	9

# Grid

```
<style>
.grid-container {
  display: grid;
  grid-template-columns: 80px 200px auto 30px;
  grid-gap: 10px;
  background-color: #2196F3;
  padding: 10px;
}
```

```
.grid-container > div {
  background-color: rgba(255, 255, 255, 0.8);
  text-align: center;
  padding: 20px 0;
  font-size: 30px;
}
</style>
</head>
<body>
```

```
<h1>grid-template-columns</h1>
```

<p>Use the `grid-template-columns` property to specify the size of each column.</p>

```
<div class="grid-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
  <div>5</div>
  <div>6</div>
  <div>7</div>
  <div>8</div>
</div>
```

## grid-template-columns

Use the `grid-template-columns` property to specify the size of each column.

1	2	3	4
5	6	7	8

# Grid

```
.item1 {  
  grid-column: 1 / 5;  
}
```

1				2	3
4	5	6	7	8	9
10	11	12	13	14	15