

Pseudo Classes and Elements

Designing for Consistent Appearance

Pseudo-Classes

- Elements that are dynamically populated or dependent on tree structure
- You have seen this before...

```
a:hover { }
```

Types of Pseudo-Classes

- **Link**
 - **:link, :visited**
- **User Action**
 - **:hover, :active, :focus**
- **Forms (interfaces)**
 - **:enabled, :checked, :disabled**

Types of Pseudo-Classes

- **Structural/Positional**
 - **:first-child, :last-child, :nth-child(), :only-child**
 - **:first-of-type, :last-of-type, :only-of-type**

```
li:first-child{  }  
li:nth-child(4){  }  
p:empty{  }  
img:only-of-type{  }  
p:last-of-type{  }
```

Pseudo-Elements

- **These elements aren't part of the DOM**
- **Can be used to style specific (unique) parts of the page**

Types of Pseudo-Elements

- **Textual**
 - **:first-letter, :first-line**
- **Positional/Generated**
 - **:before, :after**
- **Fragments**
 - **::selection**

Review

- **Pseudo-elements and classes are just one more way to add style to your page**
- **I haven't covered every combination so make sure to do some investigation on your own**



© Colleen van Lent
University of Michigan
School of Information

Unless otherwise noted, this work is licensed under the
CC BY-NC 4.0 license.

Transitions

Transitions

- **When elements transition from one state to another, you can alter their appearance**
 - **If you hover over the link, change the color**
 - **If an image comes into focus, change the size**



The Properties

- **transition-property**
 - **What is it you want to change? (size, color, position, etc.)**
- **transition-duration**
 - **How long should each transition last?**
- **transition-timing**
 - **Should it be a smooth transition (linear)? Or different?**
- **transition-delay**
 - **How long should the wait be before the transition begins?**

Setting up

1. **Define your element**
2. **Choose the elements for transition**
3. **Define the new values**
 - **You must combine this step with a pseudo-class**

Example (CSS3-transitions)

```
div{  
    color: #000000;  
    background: #2db34a;  
    line-height: 200px;  
    text-align: center;  
    width: 250px;  
    height: 200px;  
    border-radius: 6px;  
}
```

Example (CSS3-transitions)

```
div{  
  color: #000000;  
  background: #2db34a;  
  line-height: 200px;  
  text-align: center;  
  width: 250px;  
  height: 200px;  
  border-radius: 6px;  
  transition-property: color, width, background, border-radius;  
  transition-duration: .5s;  
  transition-timing-function: linear;  
  transition-delay: .5s;  
}
```


Example (CSS3-transitions)

```
div:hover{  
    color: #ffffff;  
    width: 350px;  
    background: #2D31B3;  
    border-radius: 50%;  
}
```

Using Shorthands

- If you have multiple properties transitioning, you can use shorthand:

```
transition: background .2s linear, border-radius  
           1s ease-in 1s;
```

Review

© Colleen van Lent
University of Michigan
School of Information

Unless otherwise noted, this work is licensed under the
CC BY-NC 4.0 license.

Transforms

Transforms

- **Provide option for changing the appearance of elements**
- **Two-dimensional**
- **Three-dimensional**

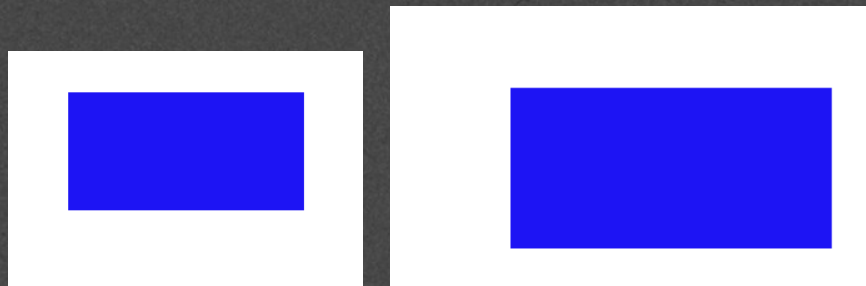
2D Transform Options

- **Options**
 - **translate**
 - **rotate**
 - **scale**
 - **skew**
 - **matrix**

translate

- **transform:translate(x, y);**
 - move x pixels to the left/right and y pixel up/down

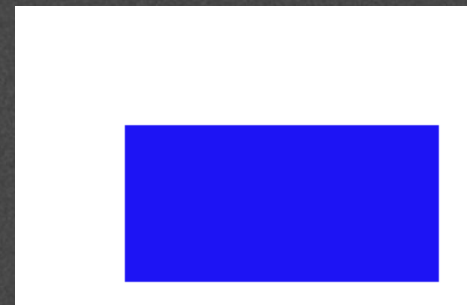
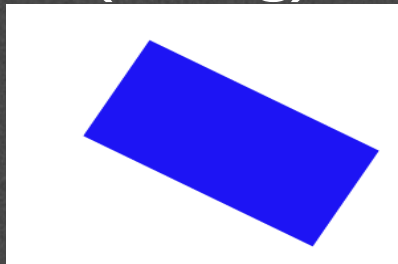
transform:translate(100, 75);



rotate

- `transform: rotate(deg);`
 - Rotate/"spin" the element a certain number of degrees

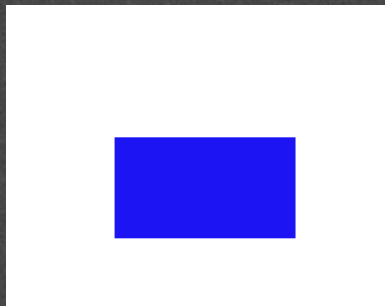
`transform: rotate(30deg);`



scale

- `transform:scale(width, height);`
 - Change the width and height of the element

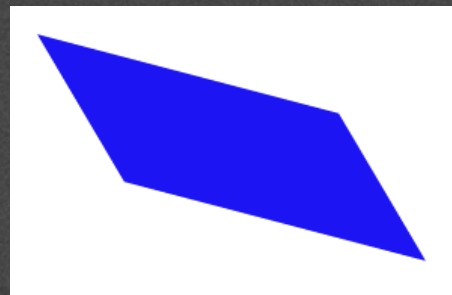
`transform:scale(2,3);`



skew

- `transform:skew(x-angle, y-angle);`
 - Rotate the element a certain number of degrees along the x and y axis

`transform:skew(30deg, 15deg)`



matrix

- **matrix()** - combines all of the 2D transform methods into one

3D rotate

- You can rotate along the x, y, or z dimension along a given degree
- `transform: rotateY(deg)`
- `transform: rotateX(deg)`
- `transform: rotateZ(deg)`
- `transform: rotate3d(x, y, z)`

Others

- **3D scale**
- **3D translate**

Review

- **Transforms are one more way to modify the look of your page.**
- **Often combined with state changes**
- **Will typically require browser prefixes.**

© Colleen van Lent
University of Michigan
School of Information

Unless otherwise noted, this work is licensed under the
CC BY-NC 4.0 license.

Accessible Navigation

Navigation

- **Navigation is a critical aspect of accessibility**
- **Sighted users have tried and true visual cues to orient them on a page**
 - **Banner**
 - **Search box**
 - **Main navigation box**
 - **Content well**
- **Blind and low-vision users rely on proper coding of page for orientation**

What if you can't see?

- **Title of page lets you know what page you're on when page loads**
- **Proper heading placement and hierarchy conveys organization of page and allows SR users to skip navigation**
- **Link descriptions convey content of page and organization of site**

Proper `<h1>` heading

- Screen readers can find and list headings
- `<h1>` heading uniquely identifies the page in the website
- Should be placed directly in front of the main content of the page
- The `<h1>` header should also match at least a subset of the the page `<title>`

Proper heading hierarchy

- Headings need to be properly nested to convey organization of the page
- `<h2>` tags follow the `<h1>` tag, the `<h3>` tags follow the `<h2>` tags, etc.

```
<h1></h1>
```

```
<h2></h2>
```

```
<h3></h3>
```

```
<h3></h3>
```

```
<h2></h2>
```

```
<h2></h2>
```


Off-page headings

- Useful when you want to give SR users a navigational aid without cluttering presentation
- Use CSS to position headings off-page

```
.offpage {  
    position: absolute;  
    left: -1000px;  
}
```

- Don't use {display: none} or {visibility: hidden}

Meaningful link text

- **Screen readers can find and list links**
- **Descriptions for the links must be meaningful out of context, via tabbing or presented in a list**
- **Don't use “here”, “click here”, “read this”, and “more”**
- **Don't use URL as a link description—will sound like gibberish, unless very short and intuitive**

Review

- **How easy is it to navigate your page?**
- **What would happen if the colors weren't there?**
- **What would happen if you couldn't use a mouse?**
- **Plan for everyone**

© Colleen van Lent
University of Michigan
School of Information

Unless otherwise noted, this work is licensed under the
CC BY-NC 4.0 license.

Final Project

Objective

- **Create your own unique site.**
- **Create a stylesheet that will be used by three different html files.**

Getting Started

- Using Homework Two as a starting point, or developing something completely from scratch create a three page site with your own HTML.
- Validate your html pages using the [w3validator](#).

Goal

Sample Site

Peer grading

- **Follow the written specifications.**
- **Grades will be based on level of completion.**
- **You can specify your preferred screen size for grading.**
- **Proper standards do apply – make sure everything passes the validators.**

Have Fun!

- **Don't aim for perfection, demonstrate growth.**
- **There is always more that you can do to style your site, that doesn't mean you should do it.**
- **Utilize Inspect Element!**

© Colleen van Lent
University of Michigan
School of Information

Unless otherwise noted, this work is licensed under the
CC BY-NC 4.0 license.

Closing

Congratulations

- You have come a long way from the plain pages we began with!

What next?

- **Consider creating your own site**
- **Continue to practice your skills**
 - **You are ready to join a Meet Up, or offer your skills as a TA at workshops.**
- **Begin to learn about using JavaScript to add interactive elements to your site**
- **Learn more about Responsive Design**

Thank you!

© Colleen van Lent
University of Michigan
School of Information

Unless otherwise noted, this work is licensed under the
CC BY-NC 4.0 license.