SOUTHERN ILLINOIS UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

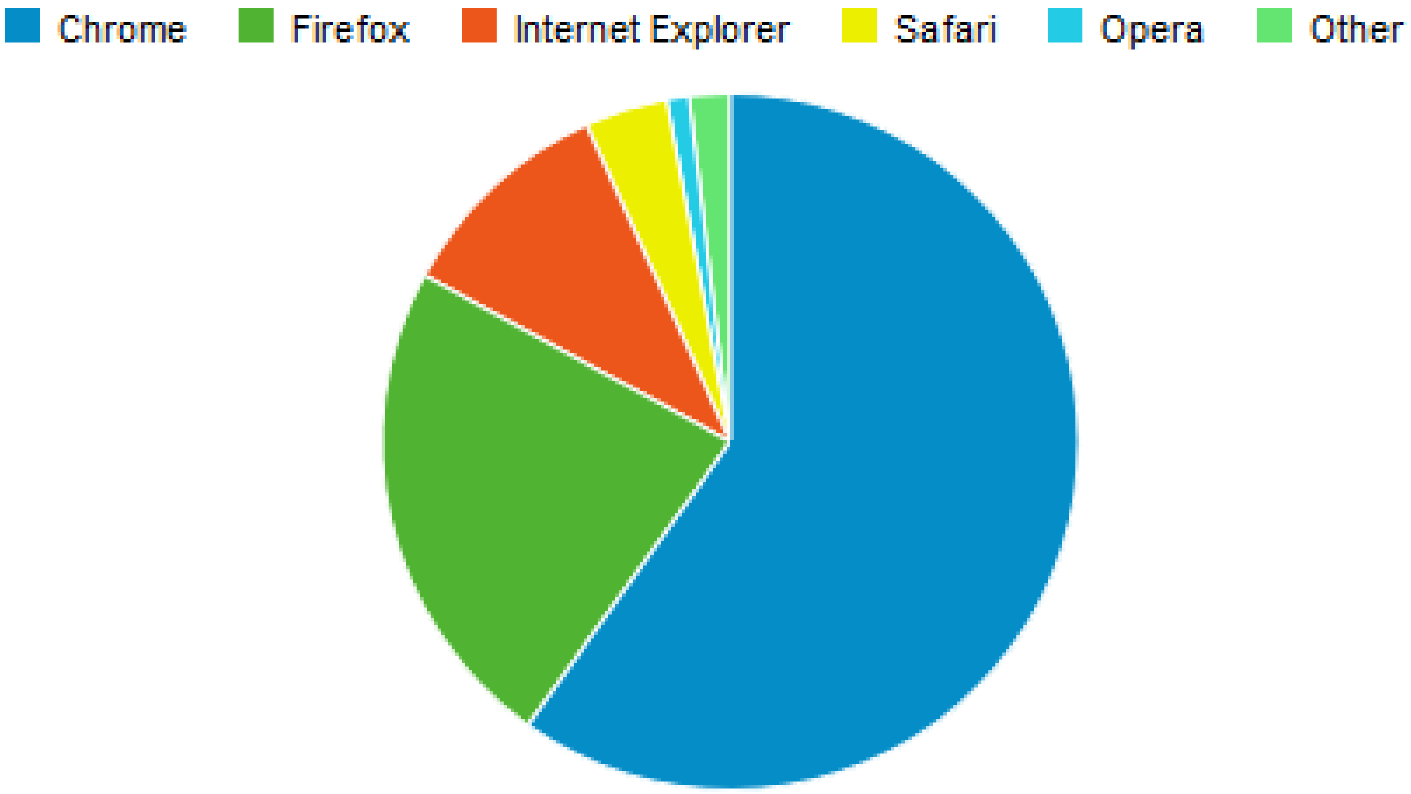
SIU CARBONDALE

CS-350/491 WEB APPLICATION DEVELOPMENT

Agenda

- Browser Support
- Multiple Styles
- Transform
- Transition
- Animation
- Keyframes

Browser Usage



Source: http://www.w3schools.com/browsers/default.asp

CSS

- Transform
- Transition
- Animations

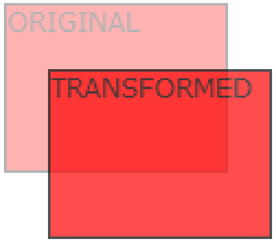
CSS Samples

1. http://dev.sencha.com/deploy/css3-ads/
2. http://media.24ways.org/2009/14/5/index.html
3. http://zurb.com/playground/sliding-vinyl
4. http://zurb.com/playground/css3-polaroids
5. http://zurb.com/playground/osx-dock
6. http://mrdoob.github.io/three.js/examples/css3d_periodictable.html
7. http://tympanus.net/Tutorials/OriginalHoverEffects/index8.html
8. http://www.webkit.org/blog-files/leaves/index.html
9. http://hompimpa.googlecode.com/svn/trunk/CSS3-Windows7-Start-Menus/index.html#
10. http://leaverou.github.io/animatable/
11. http://codepen.io/juliangarnier/pen/idhuG
12. http://tympanus.net/Tutorials/AnimatedButtons/index.html
13. http://tympanus.net/Development/FullscreenImage3DEffect/
14. http://tympanus.net/Tutorials/CreativeCSS3AnimationMenus/
15. http://tympanus.net/Tutorials/CircleNavigationEffect/
16. http://cssdeck.com/labs/google-doodle-in-css-follow-up
17. http://www.romancortes.com/ficheros/css-coke.html
18. http://www.clicktorelease.com/code/

2D Transforms

- We can translate, scale, turn, spin and stretch elements
- Chrome uses –webkit- prefix
- CSS properties:
  - transform : transform-function1 transform-function2 ..

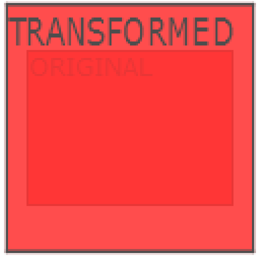
    -> applies a 2D or 3D transformation to an element
  - transform-origin : <percentage> | <length> | left | center | right | top | bottom -> allows the user to change the position on transformed elements (moves the point of origin of transformation)
- 2D transform functions - translate:



  - translate(x,y) – moves the element along the X and Y axes
  - translateX(x) – moves the element along the X axis
  - translateY(y) – moves the element along the X axis

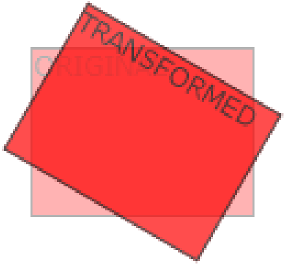
2D Transforms (2)

- 2D transform functions – scale, skew, rotate, general :
  - scale(x,y) – changes the width to x times the original and the height to y times the original
  - scaleX(x) – changes the width to x times the original
  - scaleY(y) – changes the height to x times the original

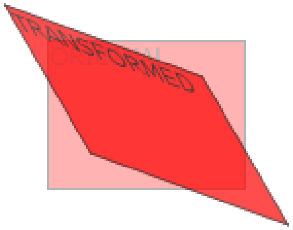


  - rotate(angle) – rotate element by angle;

    ex.: rotate(45deg);



  - skew(x-angle,y-angle) – skew transform along the X and Y axes
  - skewX(x-angle) – skew transform along the X axis
  - skewY(y-angle) – skew transform along the Y axis



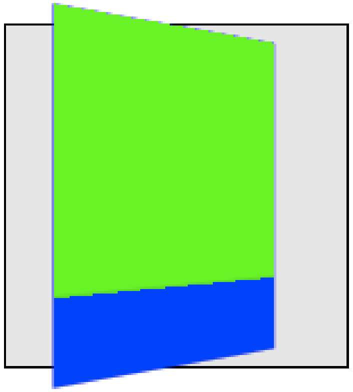
  - matrix(n,n,n,n,n,n) – general transformation

3D Transforms

- Apply 3D transforms to elements
- Chrome uses –webkit- prefix
- CSS properties:
  - transform : transform-function1 transform-function2 ..
  - transform-origin : <percentage> | <length> | left | center | right | top | bottom
  - transform-style : flat | preserve-3d -> a nested child element will not preserve its 3d position (‘flat’) or it will preserve it position (‘preserve-3d’)
  - perspective : none | number -> how many pixels the element is placed from the viewport
  - perspective : x y -> define the view’s x- and y-axis for nested elements; x,y= left | center | right | length | percent

3D Transforms (2)

- 3D transform functions:
  - translate3d(x,y,z)
  - translateX(x)
  - translateY(y)
  - translateZ(z)
  - scale3d(x,y,z)
  - scaleX(x)
  - scaleY(y)
  - scaleZ(z)
  - rotate3d(x,y,z,angle)
  - rotateX(angle)
  - rotateY(angle)
  - rotateZ(angle)
  - perspective(n)
  - matrix3d(n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n) – 3D transform 4x4 matrix



Transform

- **To better understand the transform property, view a demo available in week 6 folder.**

- We now look at the Demo1 code:

Transform Demo (Completed Animation Demos available in week 6 folder)

- Create a folder animation
- Create a file demo1.html
- Create a folder css
- Create a file style1.css inside the css folder
- Create a folder images
- Copy the sample images from Windows > Pictures folder
  - Desert.jpg
  - Koala.jpg
  - Lighthouse.jpg

demo1.html

```
<h1>CSS Animation</h1>

<div id="div1">
        <img src="images/Desert.jpg">
        <h3>Desert</h3>
</div>
<div id="div2">
        <img src="images/Koala.jpg">
        <h3>Koala</h3>
</div>
<div id="div3">
        <img src="images/Lighthouse.jpg">
        <h3>Lighthouse</h3>
</div>
```

style1.css

```css
h1 {
        margin-bottom: 100px;
}
div {
        width: 200px;
        height: 200px;
        float: left;
        background-size: 200px 200px;
        box-shadow: 0 0 5px 5px #888;
        background-color: black;
        color: white;
}
div img {
        width: 100%;
}
```

style1.css

```
div h3 {
        margin-top: -5px;
        text-align: center;
        padding: 5px;
}
#div1 {
        transform: rotate(30deg);
        -webkit-transform: rotate(30deg); /* Safari and Chrome */
}
#div2 {
        transform: rotate(10deg);
        -webkit-transform: rotate(10deg); /* Safari and Chrome */
}
```

style1.css

```
#div3 {
        transform: rotate(180deg);
        -webkit-transform: rotate(-20deg); /* Safari and Chrome */
}
#div1:hover, #div2:hover, #div3:hover {
        transform: rotate(0deg);
        -webkit-transform: rotate(0deg); /* Safari and Chrome */
        transform: scale(1.5);
        -webkit-transform: scale(1.5); /* Safari and Chrome */
        z-index: 1;
        position: relative;
}
```

Transitions - properties

- Add effects when changing from a style to another (e.g. when :hover is used), like flash or javascript
- Style properties used:
  - transition-property : comma separated list of property names to which transition is applied
  - transition-duration : how long it take the transition to be completed (ex.: transition-duration : 2s)
  - transition-delay : when transition will start (ex.: transition-delay: 1s)
  - transition-timing-function : defines the speed of the transition; values: linear|ease|ease-in|ease-out|ease-in-out|cubic-bezier(*n*,*n*,*n*,*n*)
  - transition : shorthand property for the above properties

Sample Transition

```
div {
        transition-property: width;
        transition-duration: 1s;
        transition-timing-function: linear;
        transition-delay: 2s;
}
```

OR using shorthand transition

```
div {
        transition: width 1s linear 2s;
}
```

Source: http://www.w3schools.com/css/css3_transitions.asp

Transition Demo

- **To better understand the transition property, view a demo available in week 6 folder.**
- We now take a look at the Demo1 code:

- Create a file demo2.html
- Create a file style2.css inside the css folder

demo2.html

```
<div id="div1">
        <img src="images/Desert.jpg">
        <h3>Desert</h3>
</div>
```

style2.css

```
#div1 {
        width: 200px;
        height: 200px;
        float: left;
        background-size: 200px 200px;
        box-shadow: 0 0 5px 5px #888;
        background-color: black;
        color: white;
        transition: width 2s, background-color 3s;
        -webkit-transition: width 2s, background-color 3s;  /* Safari & Chrome */
}
```

style2.css

```css
#div1 img {
        width: 100%;
}
#div1 h3 {
        margin-top: -5px;
        text-align: center;
        padding: 5px;
}
#div1:hover {
        width:600px;
        height: 500px;
        background-color: red;
}
```

Animations - properties

- @keyframes : defines the frames of the animation
- animation-name : defines the animation name, used in @keyframes
- animation-duration : duration of the animation
- animation-timing-function : defines the speed of the transition; values: linear|ease|ease-in|ease-out|ease-in-out|cubic-bezier(*n*,*n*,*n*,*n*)
- animation-delay : startup delay (in seconds)
- animation-iteration-count : how many times the animation is played
- animation-direction : the direction in which animation is played (normal | reverse | alternate | alternate-reverse)
- animation-play-state : running or pausing the animation
- animation – shorthand property

Animations - @keyframes

```
@keyframes name-of-animation {
        keyframe-selector {
                property: value;
                …
                property: value;
        }
        …
         keyframe-selector {
                property: value;
                …
                property: value;
        }
}
```

keyframe-selector is either ‘from’ (=0%), ‘to’ (=100%) or a percent of animation Duration from 0% to 100%.

property is a CSS property.

Animations Demo

- **To better understand the animation property, view a demo available in week 6 folder.**
- We now take a look at the Demo3 code:

- Create a file demo3.html
- Create a file style3.css inside the css folder

demo3.html

```
<div id="div1">
        <img src="images/Desert.jpg">
        <h3>Desert</h3>
</div>
```

style3.css

```
#div1 {
        width: 200px;
        height: 200px;
        float: left;
        background-size: 200px 200px;
        box-shadow: 0 0 5px 5px #888;
        background-color: black;
        color: white;
        animation: myfirst 5s;
        -webkit-animation: myfirst 5s; /* Safari and Chrome */position:relative;
}
#div1 img {
        width: 100%;
}
#div1 h3 {
        margin-top: -5px;
        text-align: center;
        padding: 5px;
}
```

style3.css

```
keyframes myfirst  {
            0%  {background: black; left:0px; top:0px;}
            25%  {background: red; left:400px; top:0px;}
            50%  {background: yellow; left:400px; top:400px;}
            75%  {background: blue; left:0px; top:400px;}
            100% {background: black; left:0px; top:0px;}
}
@-webkit-keyframes myfirst  { /* Safari and Chrome */
            0%   {background: black; left:0px; top:0px;}
            25%  {background: red; left:400px; top:0px;}
            50%  {background: yellow; left:400px; top:400px;}
            75%  {background: blue; left:0px; top:400px;}
            100% {background: black; left:0px; top:0px;}
}
```

Source: http://www.w3schools.com/css/css3_animations.asp

Lab 6

- Try the demo codes
- Try the examples from w3schools/CSS
- Please see the lab6.html