

Experimental Web Design

Elements rotating to mouse movement using
Trigonometry.

By © Roger Benjume. All Rights Reserved. 2023.

About me

My name is Roger, and I am in my Senior Year studying Computer Science at Eastern Washington University. (Class of Winter Quarter 2024)

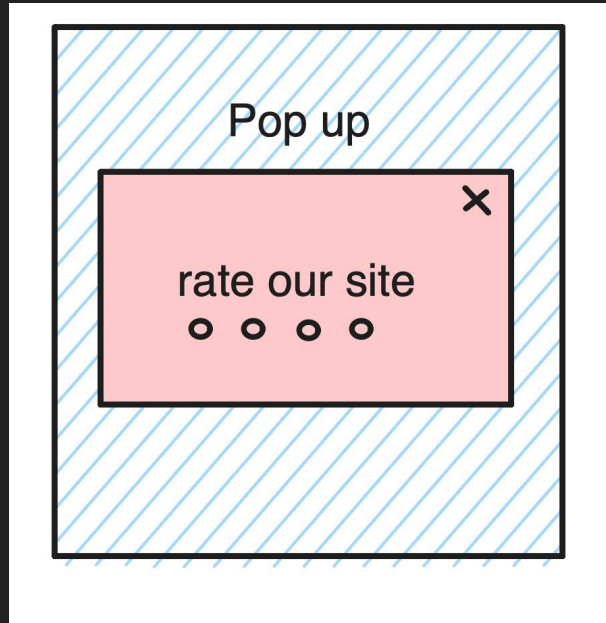
I enjoy playing the guitar and the pacific northwest.

Web design sparked my interest to study Computer Science.

I am always interested in how math is relevant to web design.

Websites employ user interaction

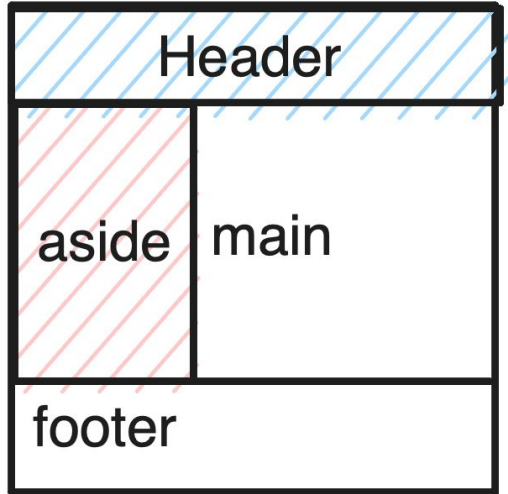
For example, you might get unwanted pop-ups.



Traditional Interaction

Help us read information

Common Layout



What is experimental web design?

Experimental web design is finding new way of interacting with websites/content in creative, untraditional, and useful ways.

Break norms of layout

Interactivity on scroll, hover, or click events

3D Websites, using WebGL (libraries or build from scratch)

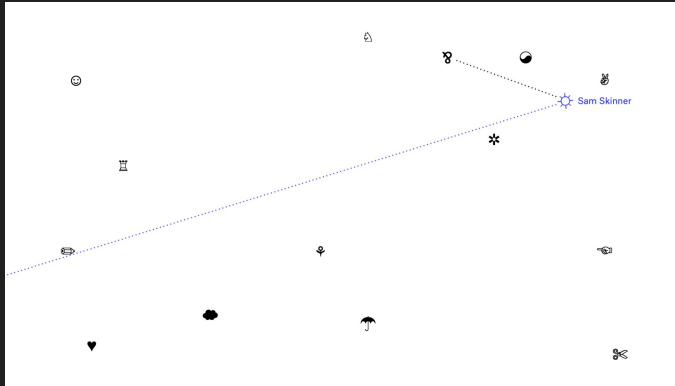
Websites may use Math :)

Always consider accessibility, and how web effects might affect users.

Experimental Interaction

samskinner by Jake Dow-Smith (Designer & Developer)

<http://samskinner.net>



What we will create, and experimental interface!

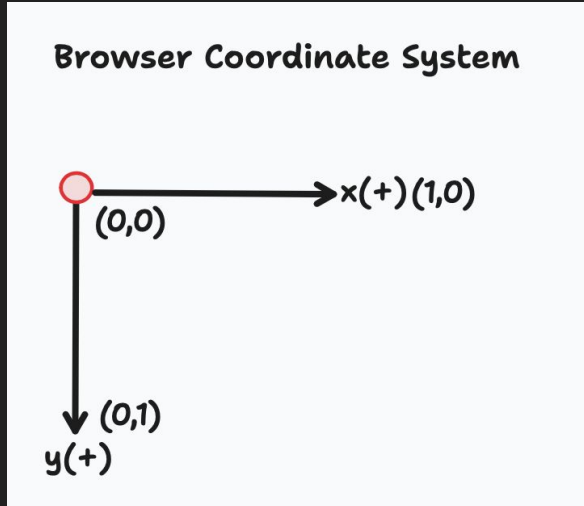
Starships that point and follow the cursor location. We will use Trigonometry! Math is useful hooray!



There are several things to consider

The browser coordinate system is different than cartesian coordinates because the y-axis is switched. However, the x-axis is the same!

Browser System: positive y-axis is downwards, and negative y-axis is upwards.

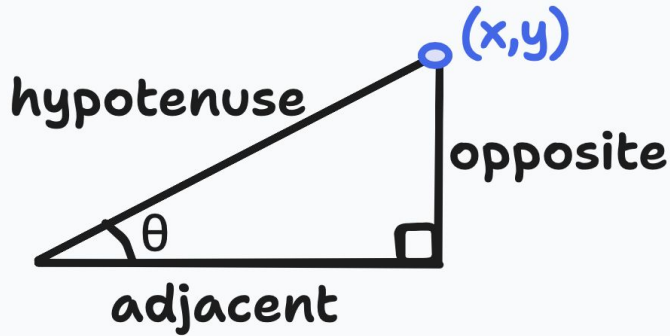


(Drawn by Roger)

Just a thought - we will use atan2 instead

We want theta (Θ) angle which will be calculated by the (x, y) coordinates. In our case it will be our (xDiff, yDiff) variables.

$$\tan\theta = \text{opposite}/\text{adjacent}$$



We need tangent to get angle!

There are several things to consider

We will not be working with a regular unit circle since we will use the built-in `Math.atan2(y,x)` function.

Returns: angle in Radians *between $[-\pi, 0]$

*Technically $[-\pi, \pi]$ according to the documentation

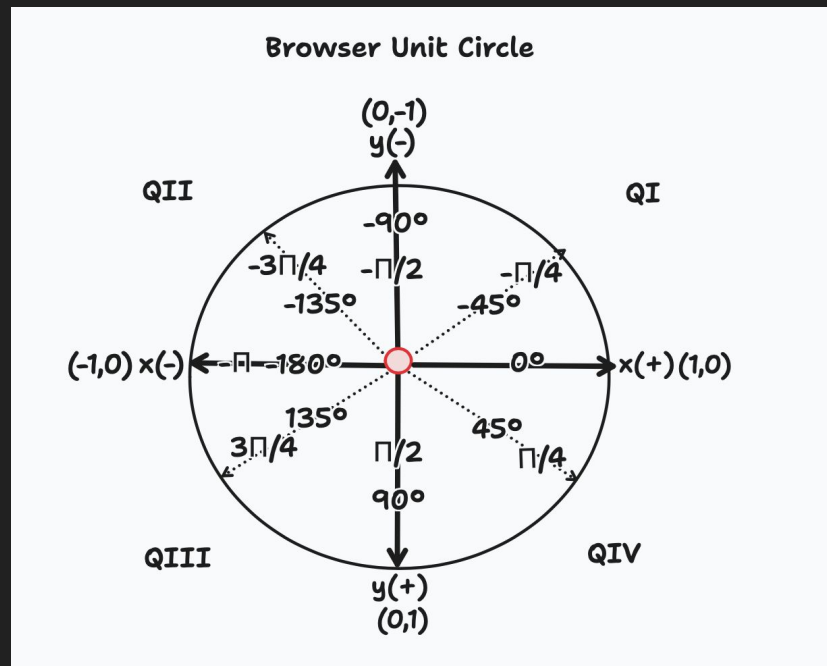
But, we will convert to degrees to make

CSS rotations possible with JavaScript.

(Drawn by Roger)

[Read about atan2 function](#)

[MDN read about CSS rotations](#)



There are several things to consider

To find horizontal center of an image: `img.getBoundingClientRect().left + img.width / 2`.

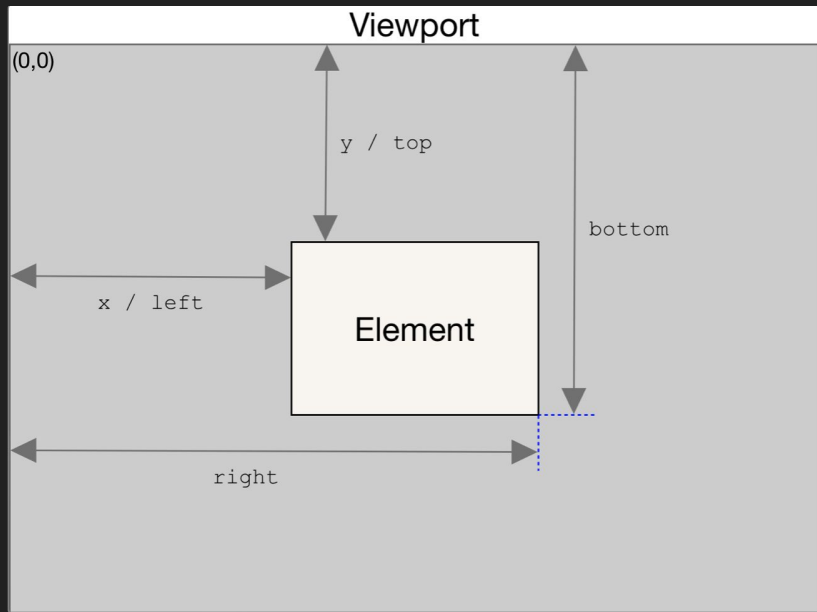
To find vertical center of an image: `img.getBoundingClientRect().top + img.height / 2`;

The difference of an image's center and cursor position results in distance calculations which output angle rotations within $[-180^\circ, 0^\circ]$ for each image at any given hover.

An equation: convert radians to degrees

$$\text{degrees} = \text{radians} \times 180 / \pi$$

Browser `getBoundingClientRect()` method



Provides the size of element, and it's position relative to the viewport.

Image by Mozilla Developer Network: [read about getBoundingClientRect](#)

Some possible rotations with the difference equation

At any given cursor location images will be within these rotation positions (left, right, up, down) and more! At the same time.

Result:

Each image will spin and point directly at the cursor location.

Possible Rotations		
(diffX,diffY)	degrees	rotation
(-90, 0)	-180	leftward
(0, 90)	90	downward
(0,-90)	-90	upward
(90, 0)	0	rightward

(4 major rotations calculated on a spreadsheet)

Possible image rotations visualized

Distance is calculated using difference of the image's center relative to the mouse position, which will ultimately output image angle rotations. Essentially, each image will have it's unit own unit circle.



rightward rotation

$$(xDiff, yDiff) = (90, 0) = 0^\circ$$



upward rotation

$$(xDiff, yDiff) = (0, -90) = -90^\circ$$



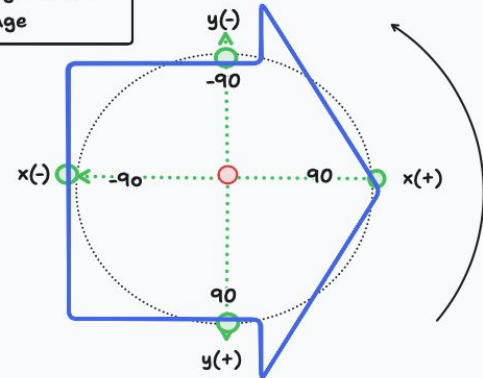
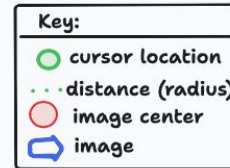
leftward rotation

$$(xDiff, yDiff) = (-90, 0) = -180^\circ$$



downward rotation

$$(xDiff, yDiff) = (0, 90) = 90^\circ$$



counter clockwise angles θ are measured

In conclusion

Thanks for attending!

Math can be a useful for web design!

Websites do not have to be boring.

What's something we can bring from our surroundings and employ it as interactivity for the web?

[Code and Slides](#)

Let's connect:

[LinkedIn](#)