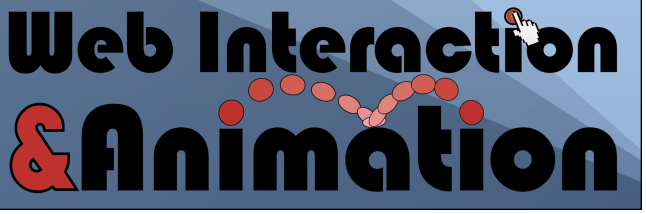
HTML5

Advanced Stuff

Web Interaction & Animation
Web Design & Development Bachelor of Science Degree



Creating Shadows

- Shadows can be colored, offset (both x& y) and have a blur
- All drawing operations on the canvas are affected by shadows

Shadows Attributes

Attributes	Description
`shadowColor`	The color to use for the shadow. Use any CSS color string. Defaults to transparent black
`shadowOffsetX`	Horizontal offset of the shadow (defaults to 0)
`shadowOffsetY`	Vertical offset of the shadow (defaults to 0)
`shadowBlur`	Blur value of the shadow. Defaults to 0. Must be set to greater than 0 to have an effect

Shadows

- Example Time

Using Patterns

- Patterns can be used in any drawing operation that has a fill or stroke property
- A Pattern can be created from an image, video or another canvas element.
  - If image is animated it will grab the poster frame or the first frame
  - For Video, the current frame is used
- Patterns can repeat:
  - In both directions
  - Only in x or y dimensions
  - Or NOT repeat at all

Pattern Function

Function	Description
`createPattern(elem, repeat)`	Creates a pattern from the given element. The first argument must be an `img`, `video`, or `canvas` element. The repeat argument can be `no-repeat`, `repeat`, `repeat-x`, or `repeat-y`

Patterns

- Example Time

FULL SAIL
UNIVERSITY

Gradients

Two Types - Two Steps

- Two types of gradients: linear & radial
- Creating gradients has 2 steps:
  1. Use the appropriate function to create the gradient
  2. Add color stops to the gradient to create color transitions
- After the gradient is created, it can be used anywhere a stroke or fill style is used.

FULL SAIL UNIVERSITY

Gradients

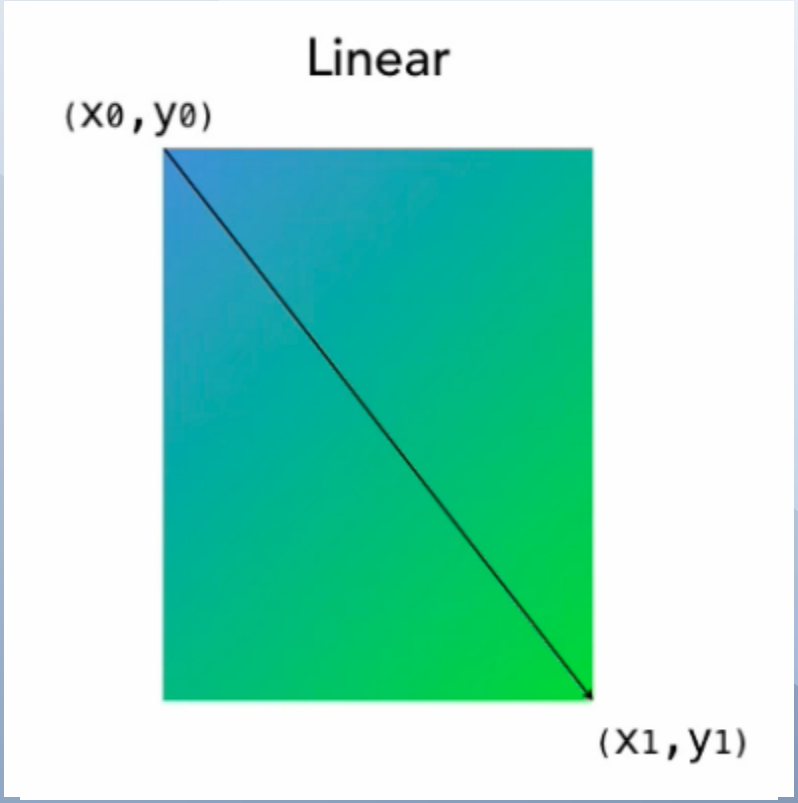
Functions

Function	Description
`createLinearGradient(x0,y0,x1,y1)`	Defines a linear gradient that starts at point `(x0, y0)` and travels to `(x1, y1)`
`createRadialGradient (x0,y0,r0,x1,y1,r1)`	Defines a radial gradient that begins with the circle whose center is at `(x0, y0)` and has radius `r0`, and travels to the circle whose center is at `(x1, y1)` and has radius `r1`
`addColorStop(position, color)`	Adds a `color` stop at the given position and has the given color. `position` is a floating point number from 0.0 to 1.0

FULL SAIL UNIVERSITY

Gradients

Linear

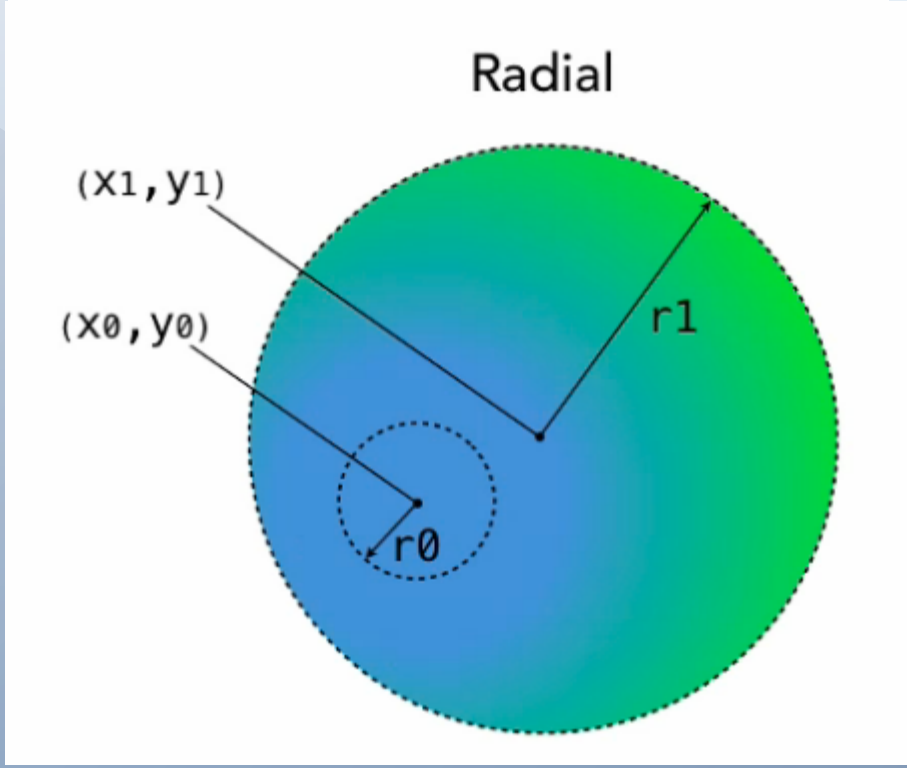


Web Interaction &Animation

FULL SAIL UNIVERSITY

Gradients

Radial



Web Interaction &Animation

Gradients

- Example Time

FULL SAIL
UNIVERSITY

Images & Video On Canvas

Placing a picture

- Images & Video can be programmatically drawn onto a canvas
- You can draw multiple ways:
  - From an img or video element in the page or one that's dynamically loaded
  - From another canvas element in the document
- Source image can be drawn, resized or cropped

Images Drawing

Functions

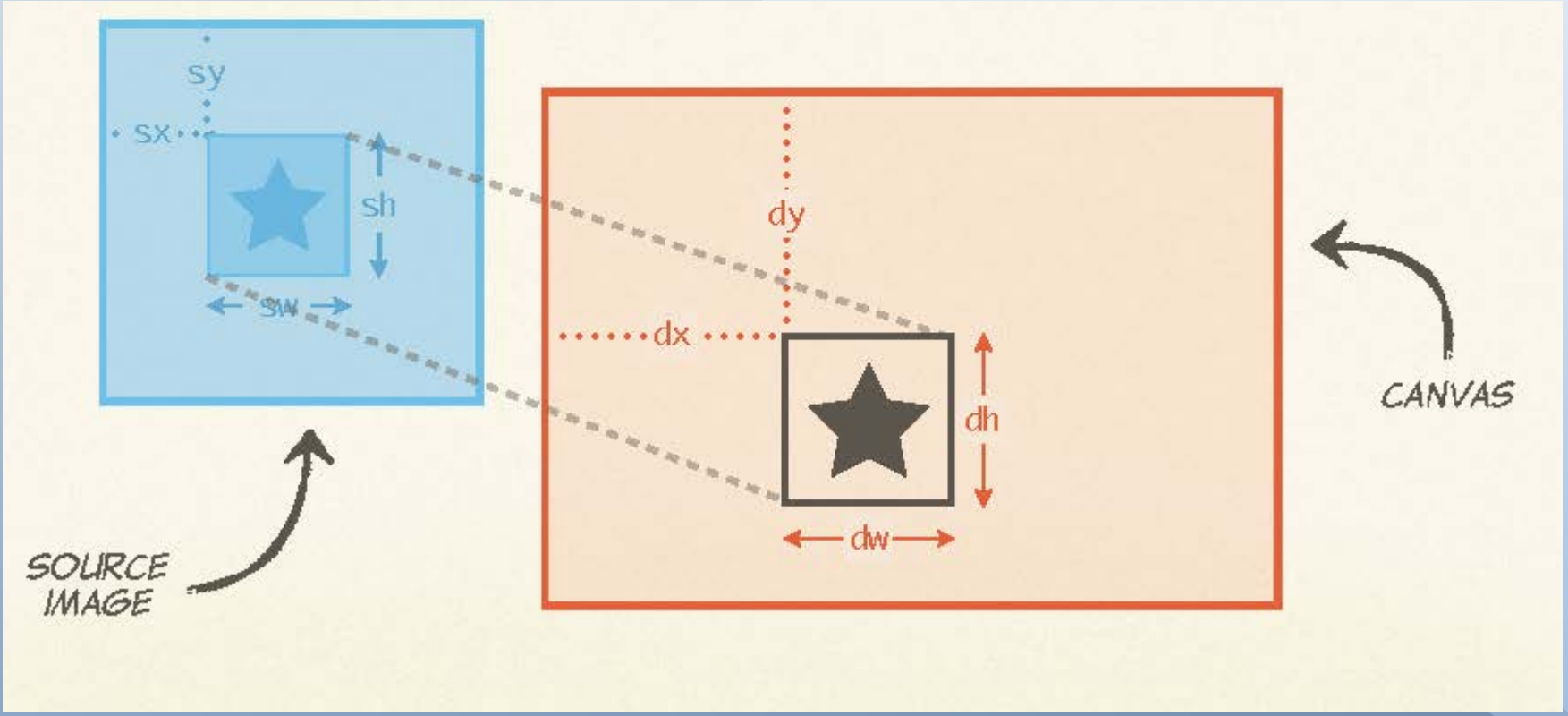
Function	Description
`drawImage(srcImg, dx, dy)`	Draws the source image onto the destination canvas at the location point (dx, dy)
`drawImage(srcImg, dx, dy, dw, dh)`	Draws the source image onto the destination canvas at the location point (dx, dy), but scales the image to fit width dw and height dh
`drawImage(srcImg, sx, sy, sw, sh, dx, dy, dw, dh)`	Draws the portion of the source image starting from point (sx, sy) and within the area defined by width sw and height sh onto to destination canvas at the location point (dx, dy), but scales the image to fit width dw and height dh

FULL SAIL UNIVERSITY

Image Drawing

Basic

```
ctx.drawImage(image, dx, dy);
```



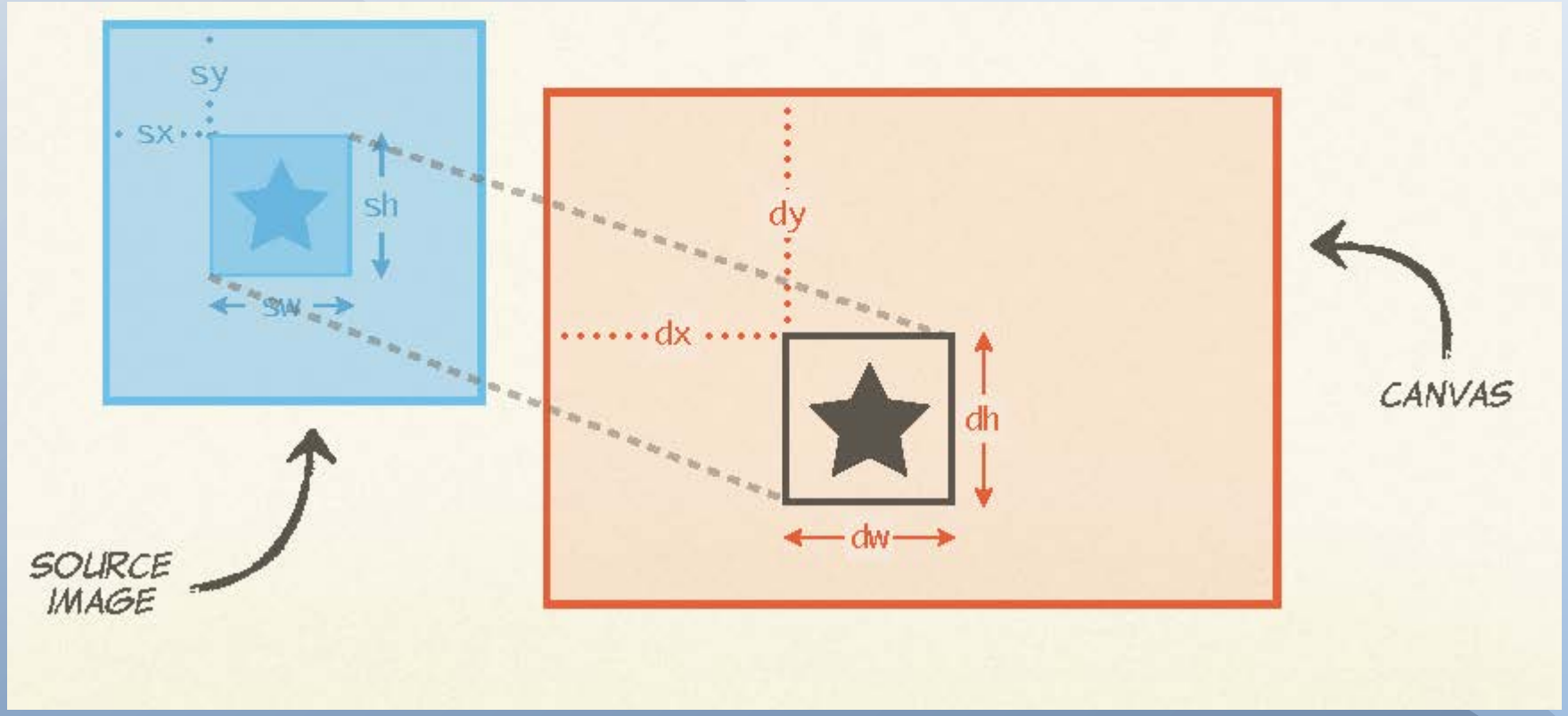
Web Interaction &Animation

FULL SAIL UNIVERSITY

Image Drawing

Scaling

```
ctx.drawImage(image, dx, dy, dw, dh);
```

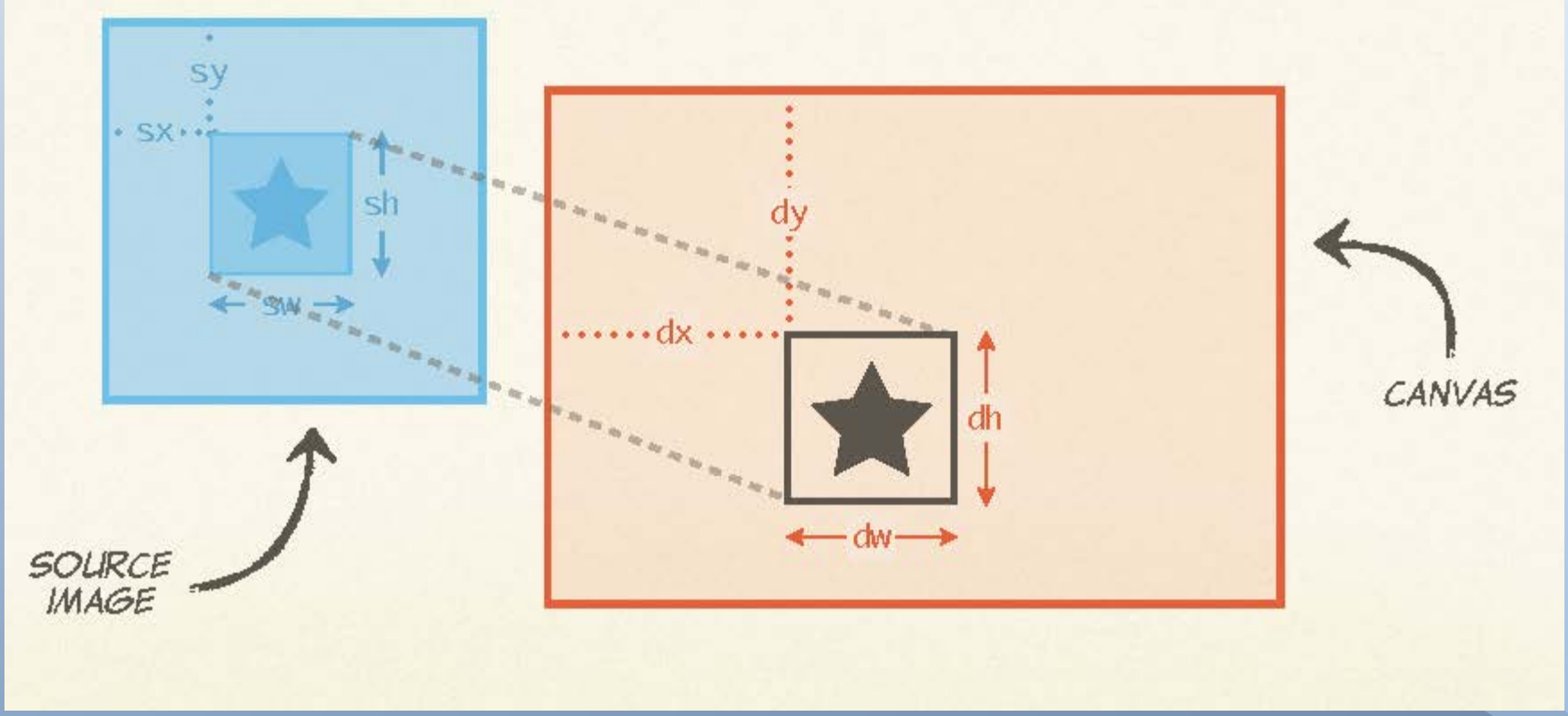


Web Interaction &Animation

Image Drawing

Slicing

```
ctx.drawImage(image, sx, sy, sw, sh, dx,
dy, dw, dh);
```



FULL SAIL
UNIVERSITY

Image Drawing

Example

Code Time

Web Interaction & Animation
Web Design & Development Bachelor of Science Degree

Web Interaction &Animation

FULL SAIL
UNIVERSITY

Clipping

Clipping allows masking of Canvas areas so anything drawn only appears in clipped areas

Web Interaction & Animation
Web Design & Development Bachelor of Science Degree

Web Interaction &Animation

FULL SAIL
UNIVERSITY

Clipping

- By Default, the entire canvas is the current clipping path.
- Its a region inside of which drawing takes place and outside of which drawing has no effect.
- Any path can be defined as a clipping path

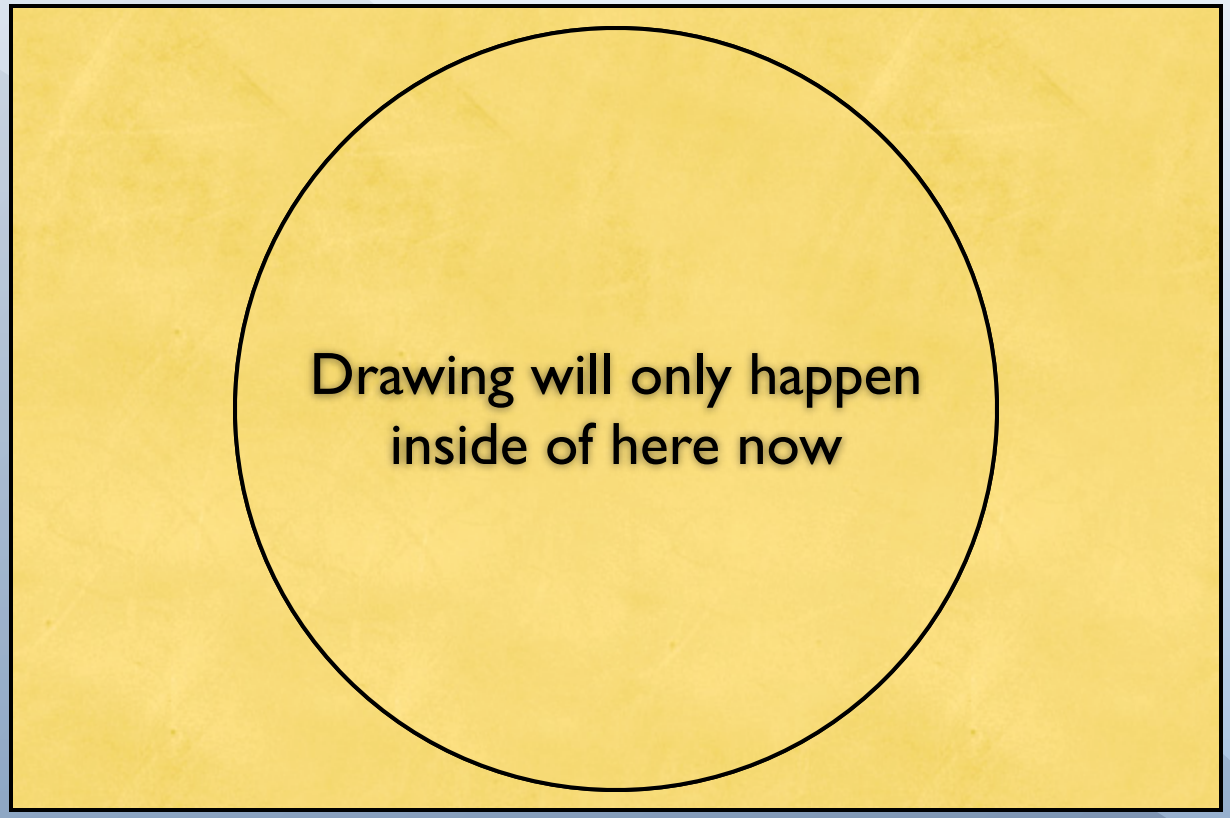
Clipping Function

Path Function	Description
`clip()`	Creates a new clipping region by calculating the intersection of the current clipping region and the area described by the current path. The new clipping region replaces the current clipping region.

`clip()` function defines the current path as a clipping path.

Each Context can have only ONE clipping region.

Clipping Function



Clipping Examples

FULL SAIL UNIVERSITY

- Code Time

Web Interaction & Animation
Web Design & Development Bachelor of Science Degree

Web Interaction & Animation

Transformations

- Transformations provide a way to affect how objects are drawn on the canvas.
- Simulates very common, but difficult to program effects

Transformations

- 3 basic transformations:
  - translate
  - scale
  - rotate
- Define your own free-form transforms

FULL SAIL
UNIVERSITY

Transformations

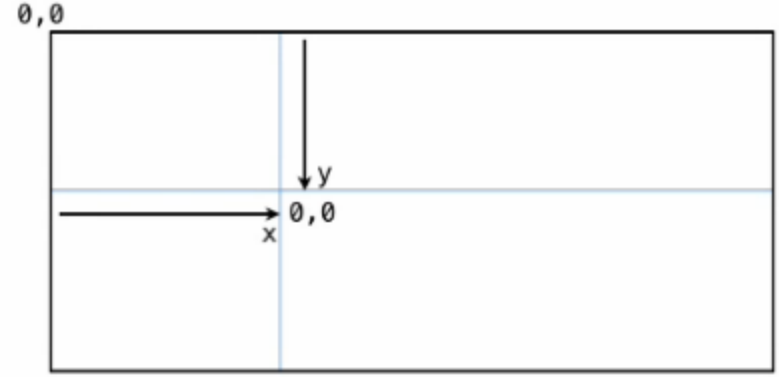
- Transforms affect ALL of the drawing operations that come after them
- They are additive - each transform is added to the previous one
- Good to save() and restore() the context state when using them

Translate Transform

¿Cómo se dice?

- **translate() transform is the simplest**
- **Moves the canvas origin to a new location**

Path Function	Description
`translate(x,y)`	Moves the origin by the amounts x and y



Translate Transform

Code

- **Example Time**

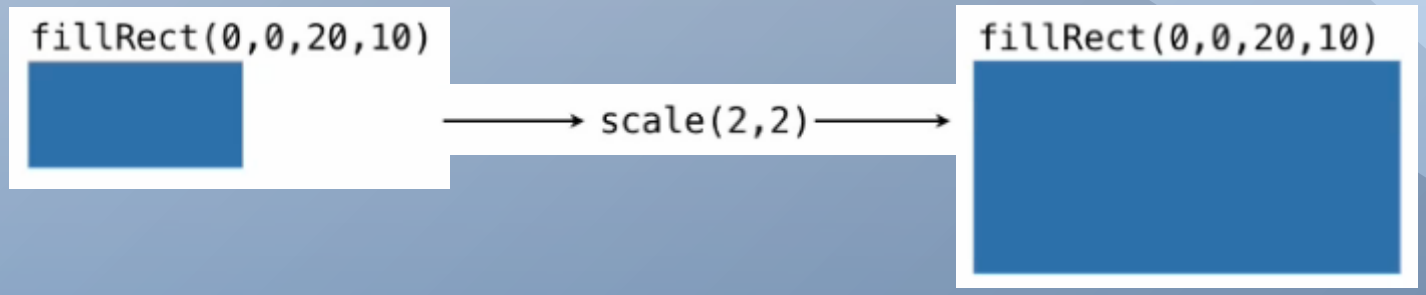
FULL SAIL UNIVERSITY

Scale Transform

Big, Bigger, Biggest?

- **scale() transform causes drawing operations to be multiplied by a given scale factor in the x & y directions independently**

Path Function	Description
`scale(x,y)`	Scale drawing operations by multiples x and y



Web Interaction &Animation

FULL SAIL UNIVERSITY

Scale Transform

Code

Example Time

Web Interaction & Animation

Web Design & Development Bachelor of Science Degree

Web Interaction & Animation

FULL SAIL UNIVERSITY

Rotate Transform

Here we go round

- rotate() transform causes drawing operations to be rotated by a given angle in radians, not degrees

- Rotation takes place around current origin....NOT an objects center!
  - To do this, use translate() to move the origin to that object's center and then rotate() it.

Path Function	Description
`rotate(angle)`	Rotate subsequent drawing operations by the given angle

FULL SAIL UNIVERSITY

Canvas Rotations

Rotating around the origin

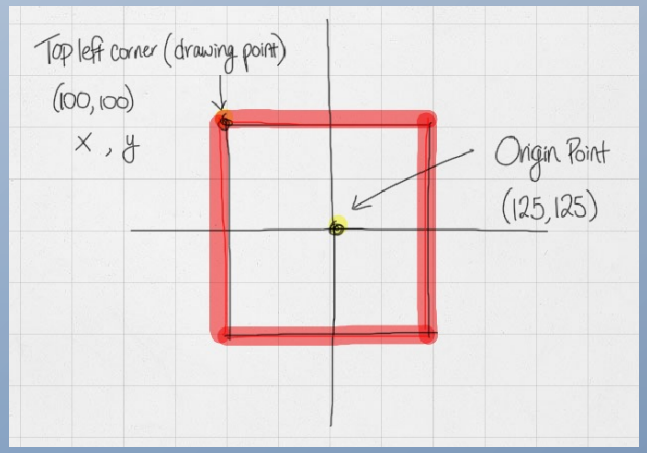
What about rotating about the origin?

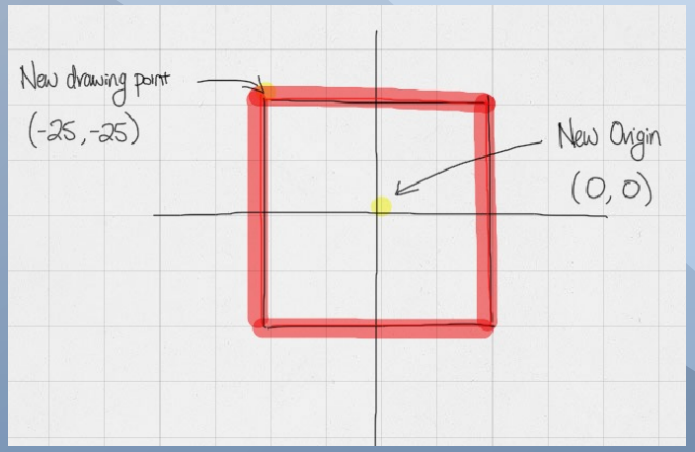
The point of origin to the center of our shape must be translated to rotate it around its own center

- Change the origin of the canvas to be the centre of the square
  - context.translate(x+.5*width, y+.5*height);

OR

- Draw the object starting with the correct upper-left coordinates
  - context.fillRect(-.5*width,-.5*height , width, height);





Web Interaction & Animation

FULL SAIL
UNIVERSITY

Rotate Transform

Code

- Example Time

Web Interaction & Animation
Web Design & Development Bachelor of Science Degree

Web Interaction &Animation

FULL SAIL UNIVERSITY

Custom Transform

This time, we are making it personal!

- **In addition to the built-in transforms, you can define your own**

- **A transform is defined as a matrix, with this format:**

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} ace \\ bdf \\ 001 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Function	Description
`transform(a,b,c,d,e,f)`	Add the given transform to the current one
`setTransform(a,b,c,d,e,f)`	Set the current transform to the given arguments

setTransform - erases all existing transforms.

FULL SAIL
UNIVERSITY

Custom Transform

Code

- **Example Time**

Web Interaction & Animation
Web Design & Development Bachelor of Science Degree

Web Interaction &Animation

FULL SAIL UNIVERSITY

Compositing

Layering

Compositing is the control of transparency and layering of objects.

This is controlled by

- **globalAlpha**
- **globalCompositeOperation**

FULL SAIL UNIVERSITY

Compositing

globalAlpha

globalAlpha

- The opacity setting that affects all drawing operations
- Defaults to 1 (completely opaque)
- Set before an object is drawn to Canvas

FULL SAIL UNIVERSITY

Compositing

globalCompositeOperation

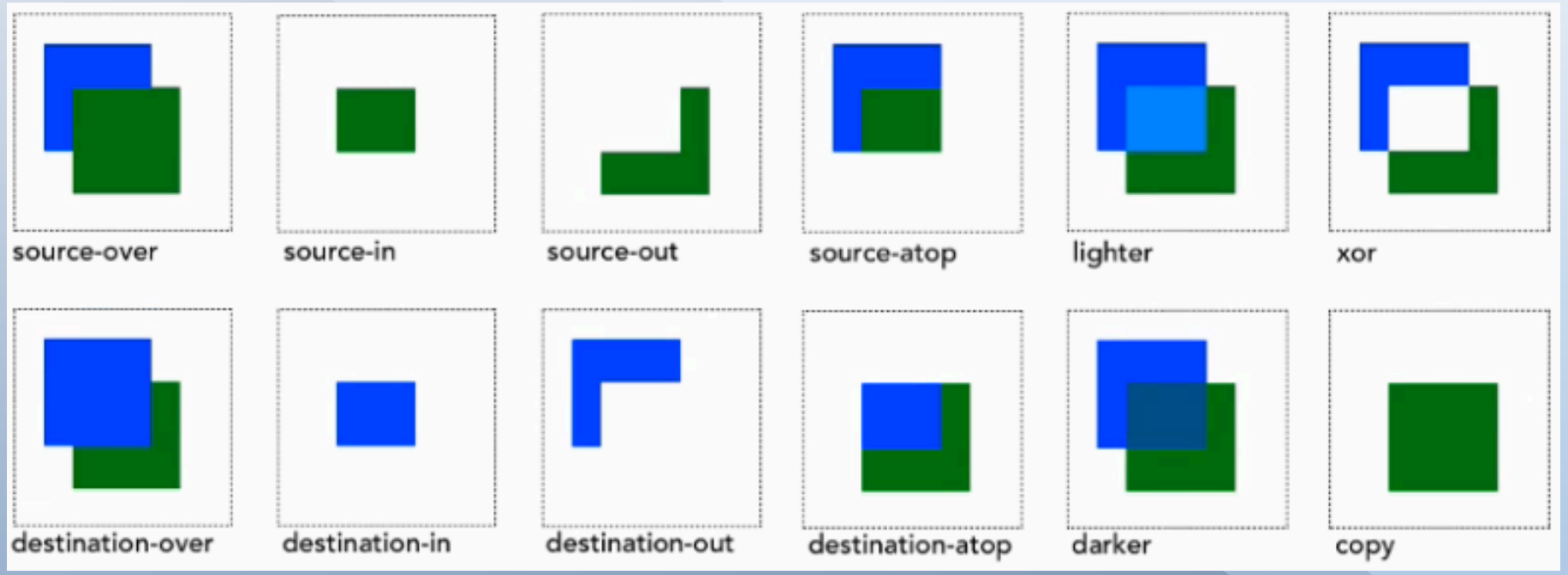
globalCompositeOperation

- Controls how new content is drawn onto the canvas surface
- Twelve different compositing methods

FULL SAIL UNIVERSITY

globalCompositeOperation

Method Roll Call



source-over is default

Web Interaction &Animation

FULL SAIL
UNIVERSITY

Custom Transform

Code

- **Example Time**

Web Interaction & Animation
Web Design & Development Bachelor of Science Degree

Web Interaction &Animation

FULL SAIL
UNIVERSITY

Accessing Raw Pixel Data

It's Hip To Be Square

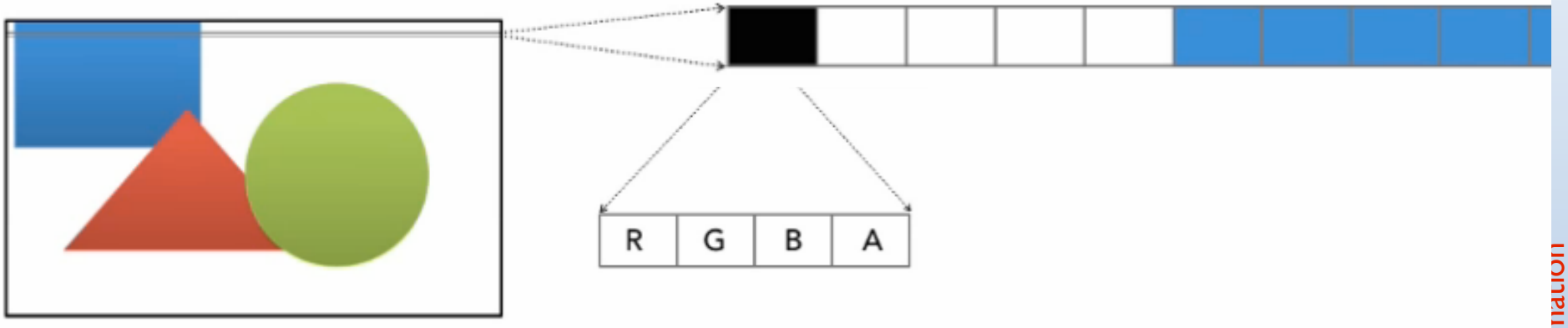
- **Canvas provides access to the individual pixel data as an array of bytes**
- **This data can be manipulated and then put back into the canvas**
- **Each row in an image is composed of a 4-byte pixel.**

Web Interaction &Animation

FULL SAIL UNIVERSITY

Accessing Raw Pixel Data

4-Byte Pixel?? Say Whaaaa??



Size of the array would be:

- **height * width * 4**

Accessing Raw Pixel Data

Methods

Attribute/Function	Description
`width, height`	Width and height of the canvas pixel data
`data`	The single-dimension array of raw pixel data
`createImageData(sw,sh)`	Creates a new image data with width `sw` and height `sh`
`createImageData(imgData)`	Creates new image data from an existing one
`getImageData(sx,sy,sw,sh)`	Gets image data within the given bounds
`putImageData(imgData, dx, dy, [dirtyDx, dirtyDy, dirtyW, dirtyH])`	Puts modified data back into the image. If the `dirty` rectangle data is supplied, then only the bits inside that rectangle are updated

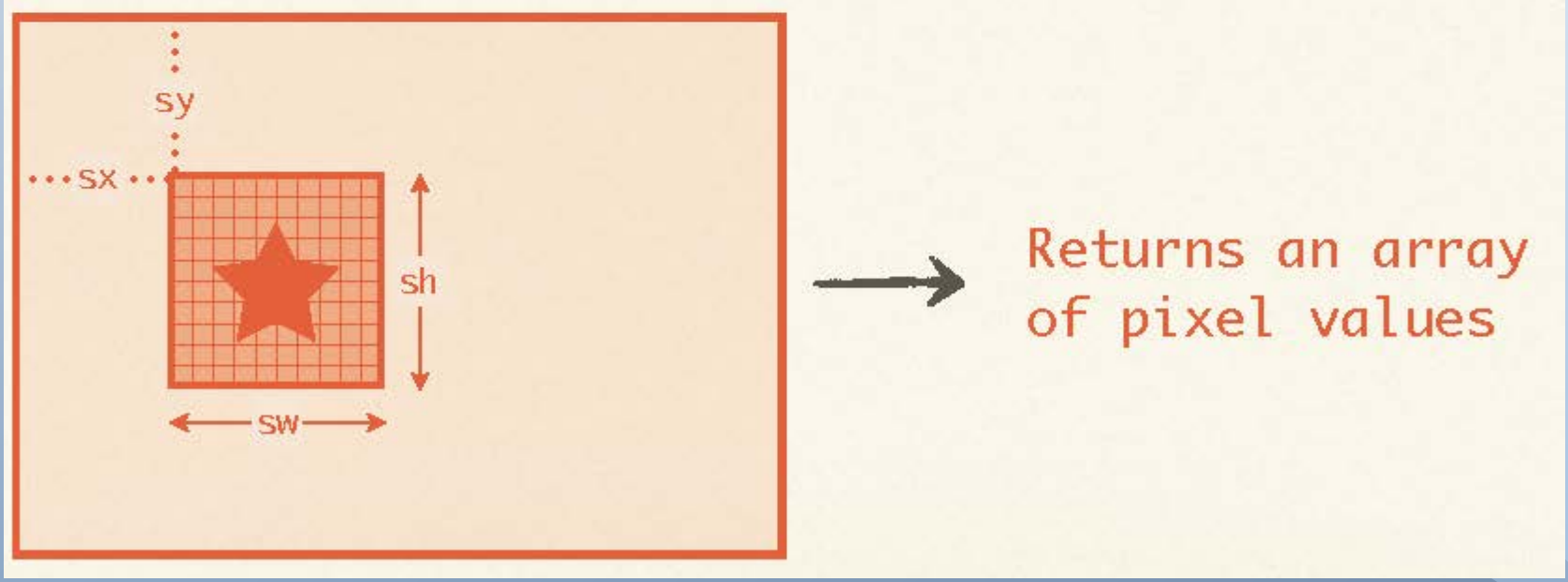
FULL SAIL UNIVERSITY

Pixel Values

Code

Individual pixel values can be retrieved

```
ctx.getImageData(sx, sy, sw, sh);
```



Web Interaction &Animation

FULL SAIL UNIVERSITY

Pixels

Code

Example Time

Web Interaction & Animation

Web Design & Development Bachelor of Science Degree

Web Interaction & Animation

FULL SAIL UNIVERSITY

Animation

Basic, but doable

- **Simulate animation by re-drawing the canvas**
- **We use a setInterval() method**

```
setInterval(function, milliseconds)
```

FULL SAIL UNIVERSITY

Animation

Code

Example Time

Web Interaction & Animation
Web Design & Development Bachelor of Science Degree

Web Interaction &Animation