02.

Intro to HTML5 Canvas

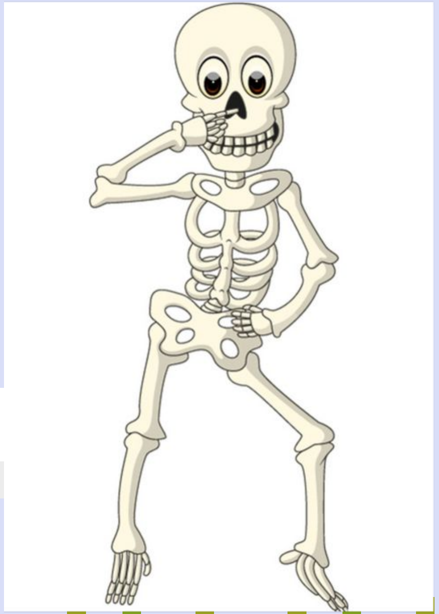
Overview of HTML5 and 2D game development, canvas coordinate system, drawing shapes, animations.

How Web Pages work

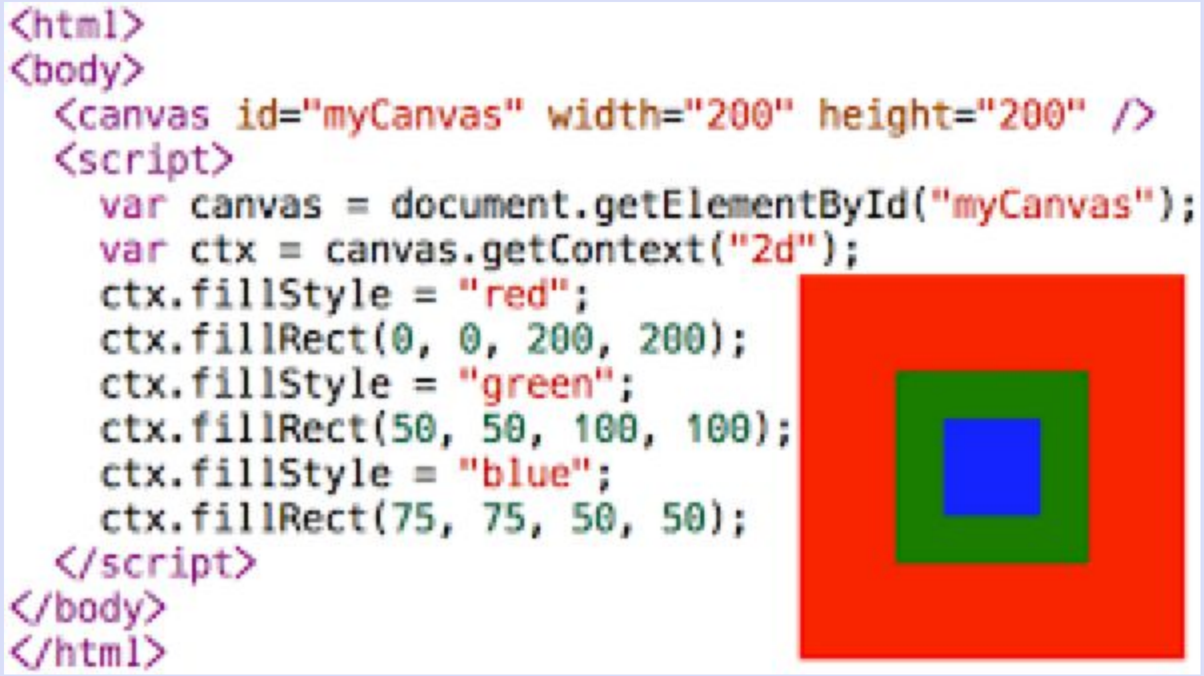
HTML

CSS

JavaScript



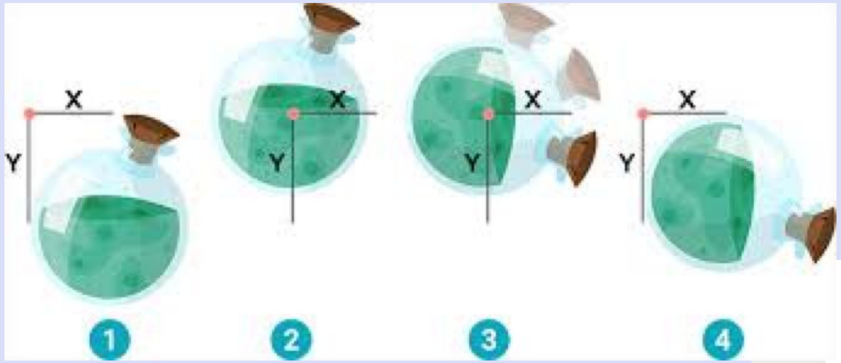
HTML5 Canvas



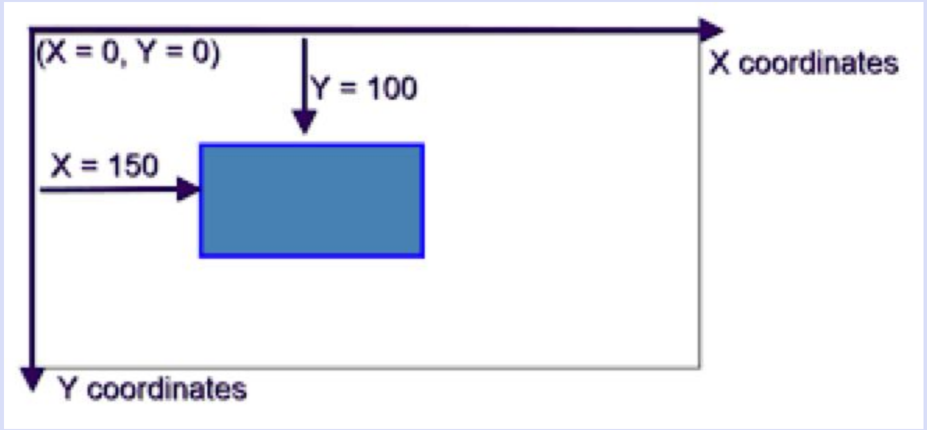
```
<html>
<body>
  <canvas id="myCanvas" width="200" height="200" />
  <script>
    var canvas = document.getElementById("myCanvas");
    var ctx = canvas.getContext("2d");
    ctx.fillStyle = "red";
    ctx.fillRect(0, 0, 200, 200);
    ctx.fillStyle = "green";
    ctx.fillRect(50, 50, 100, 100);
    ctx.fillStyle = "blue";
    ctx.fillRect(75, 75, 50, 50);
  </script>
</body>
</html>
```

TABLET APP

Canvas...is a powerful tool for developers to create rich and interactive games and apps on the web.

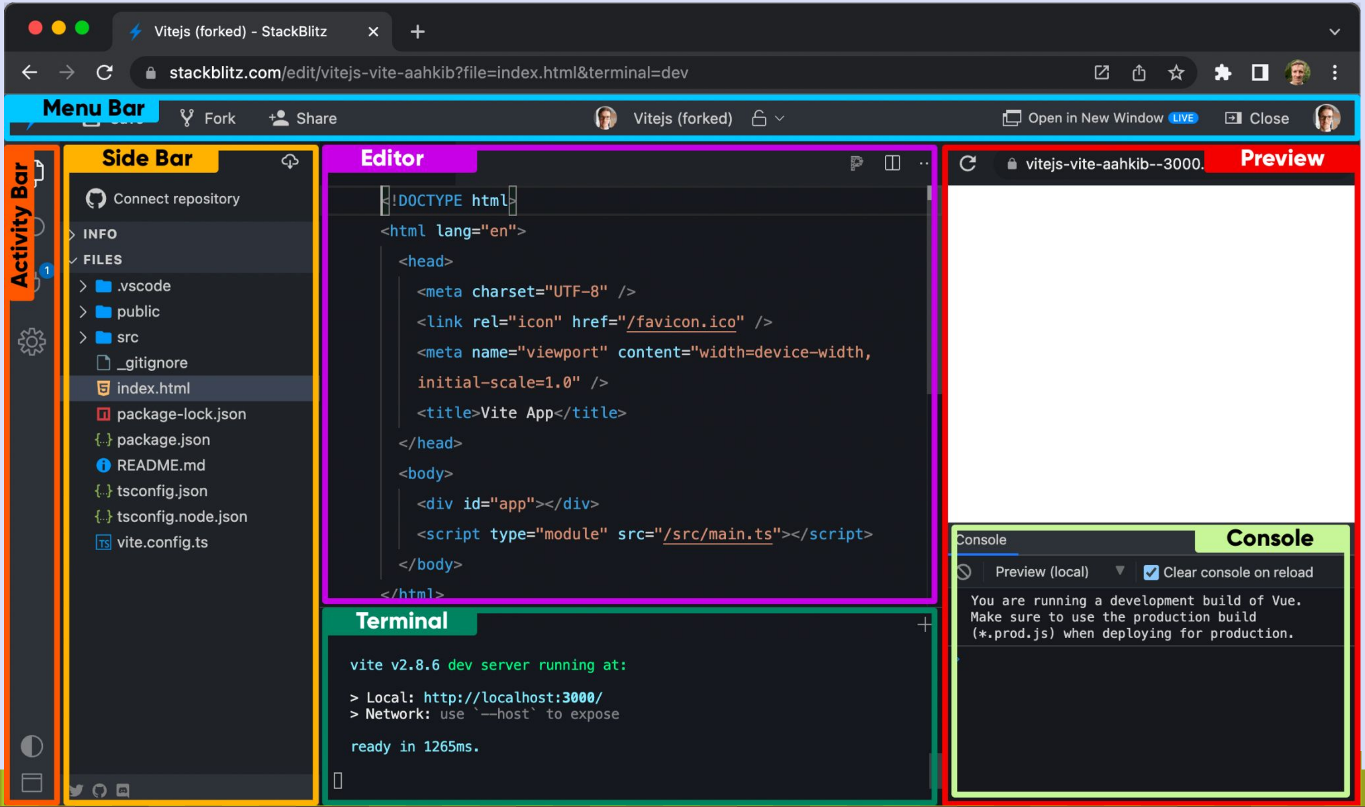


Canvas Coordinate System



Note that unlike the cartesian system the canvas is (0,0) in the top left corner.

Open up StackBlitz

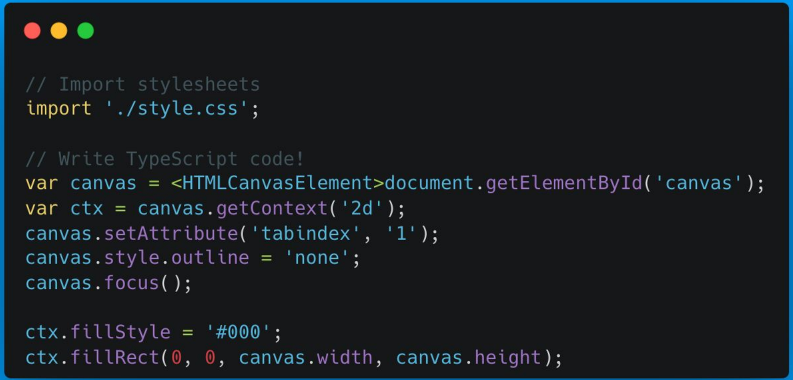


```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" href="/favicon.ico" />
    <meta name="viewport" content="width=device-width,
    initial-scale=1.0" />
    <title>Vite App</title>
  </head>
  <body>
    <div id="app"></div>
    <script type="module" src="/src/main.ts"></script>
  </body>
</html>
```

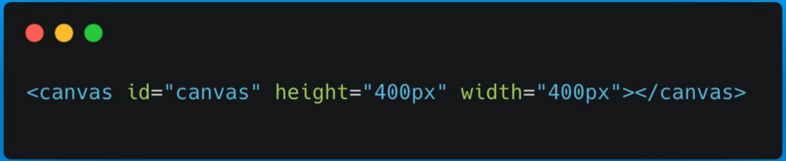
```
vite v2.8.6 dev server running at:

> Local: http://localhost:3000/
> Network: use `--host` to expose

ready in 1265ms.
```

```
You are running a development build of Vue.
Make sure to use the production build
(*.prod.js) when deploying for production.
```

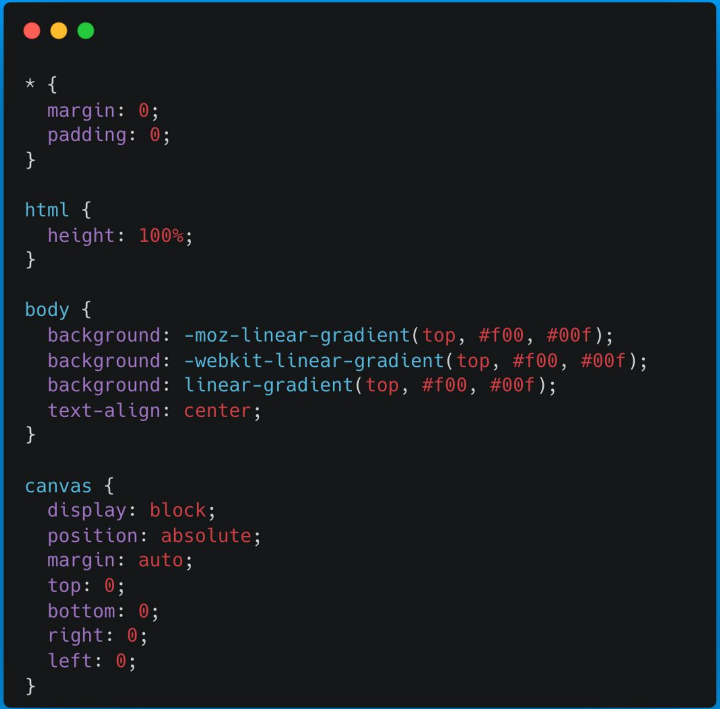
Start of Game Engine



```typescript
// Import stylesheets
import './style.css';

// Write TypeScript code!
var canvas = <HTMLCanvasElement>document.getElementById('canvas');
var ctx = canvas.getContext('2d');
canvas.setAttribute('tabindex', '1');
canvas.style.outline = 'none';
canvas.focus();

ctx.fillStyle = '#000';
ctx.fillRect(0, 0, canvas.width, canvas.height);
```

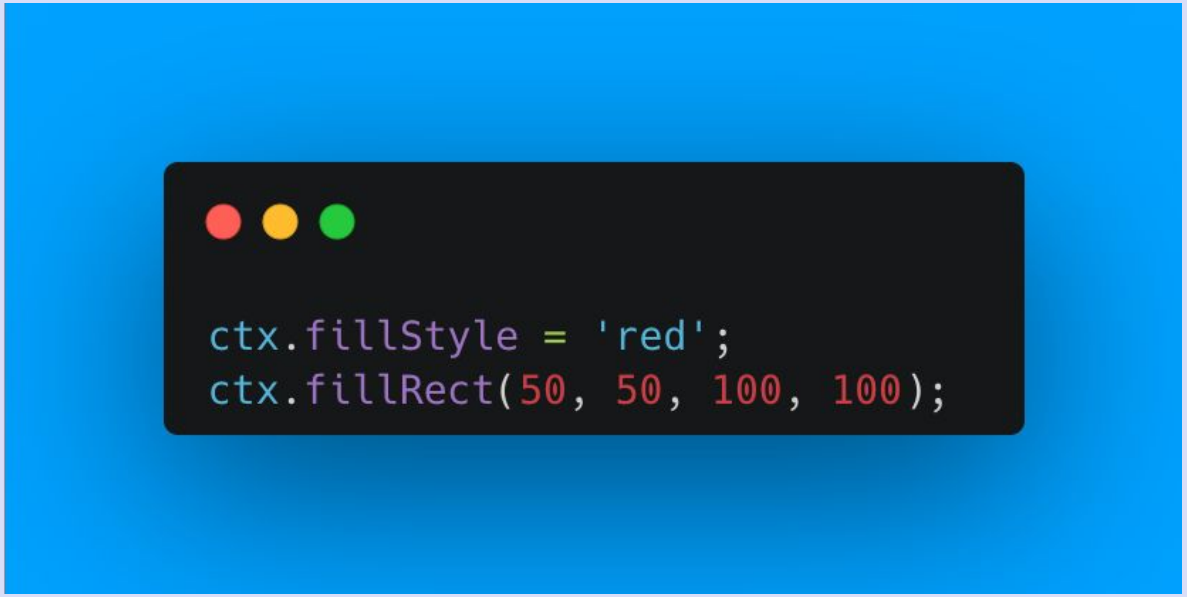


```html
<canvas id="canvas" height="400px" width="400px"></canvas>
```



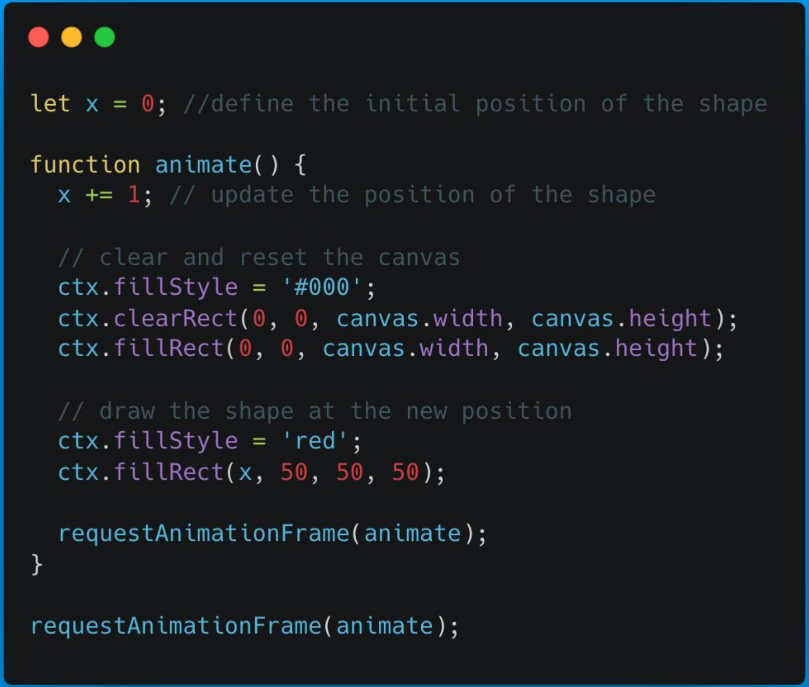
```css
* {
  margin: 0;
  padding: 0;
}

html {
  height: 100%;
}

body {
  background: -moz-linear-gradient(top, #f00, #00f);
  background: -webkit-linear-gradient(top, #f00, #00f);
  background: linear-gradient(top, #f00, #00f);
  text-align: center;
}

canvas {
  display: block;
  position: absolute;
  margin: auto;
  top: 0;
  bottom: 0;
  right: 0;
  left: 0;
}
```

Create a Rectangle



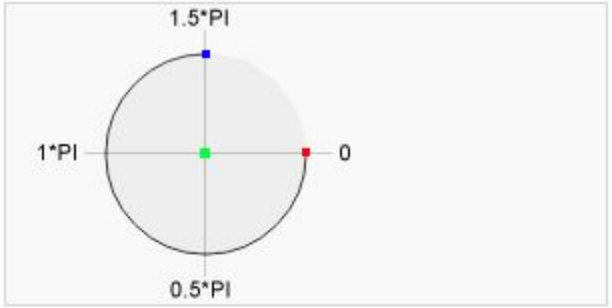
```
ctx.fillStyle = 'red';
ctx.fillRect(50, 50, 100, 100);
```

Animate a Rectangle

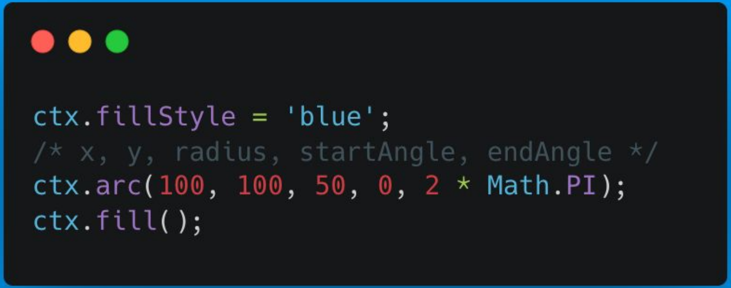


```
let x = 0; //define the initial position of the shape

function animate() {
  x += 1; // update the position of the shape

  // clear and reset the canvas
  ctx.fillStyle = '#000';
  ctx.clearRect(0, 0, canvas.width, canvas.height);
  ctx.fillRect(0, 0, canvas.width, canvas.height);

  // draw the shape at the new position
  ctx.fillStyle = 'red';
  ctx.fillRect(x, 50, 50, 50);

  requestAnimationFrame(animate);
}

requestAnimationFrame(animate);
```

Create a Circle

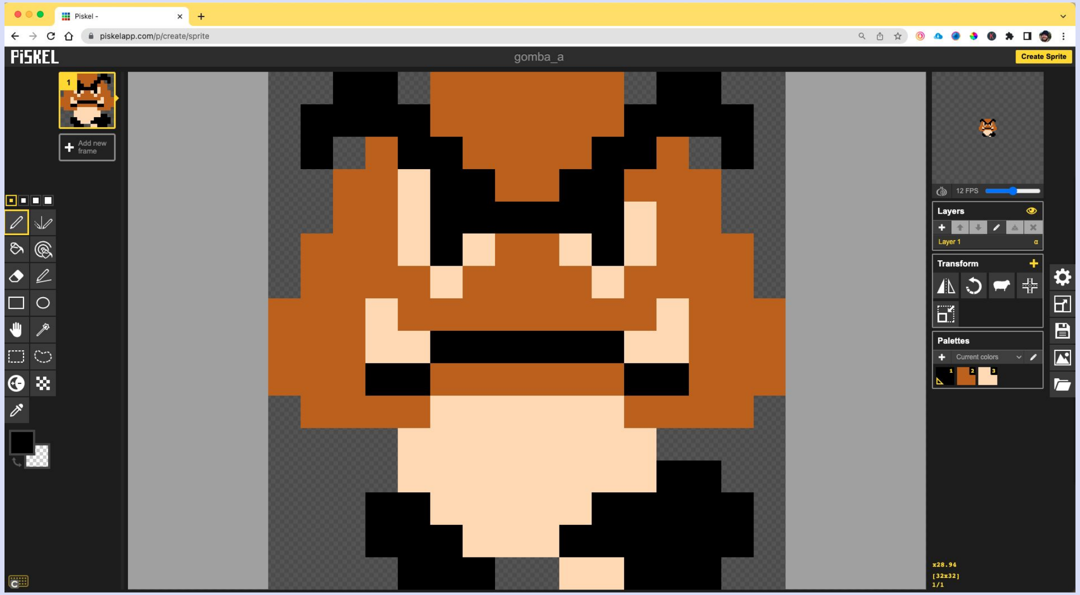


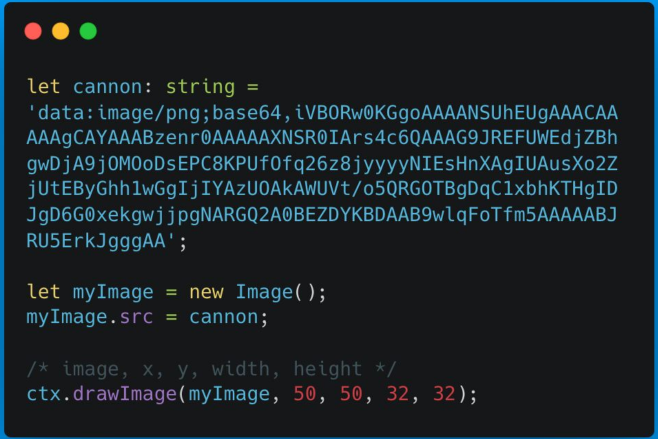
🟩 Center	arc(**100,75**,50,0*Math.PI,1.5*Math.PI)
🟥 Start angle	arc(100,75,50,**0**,1.5*Math.PI)
🟦 End angle	arc(100,75,50,0*Math.PI,**1.5*Math.PI**)



```
ctx.fillStyle = 'blue';
/* x, y, radius, startAngle, endAngle */
ctx.arc(100, 100, 50, 0, 2 * Math.PI);
ctx.fill();
```

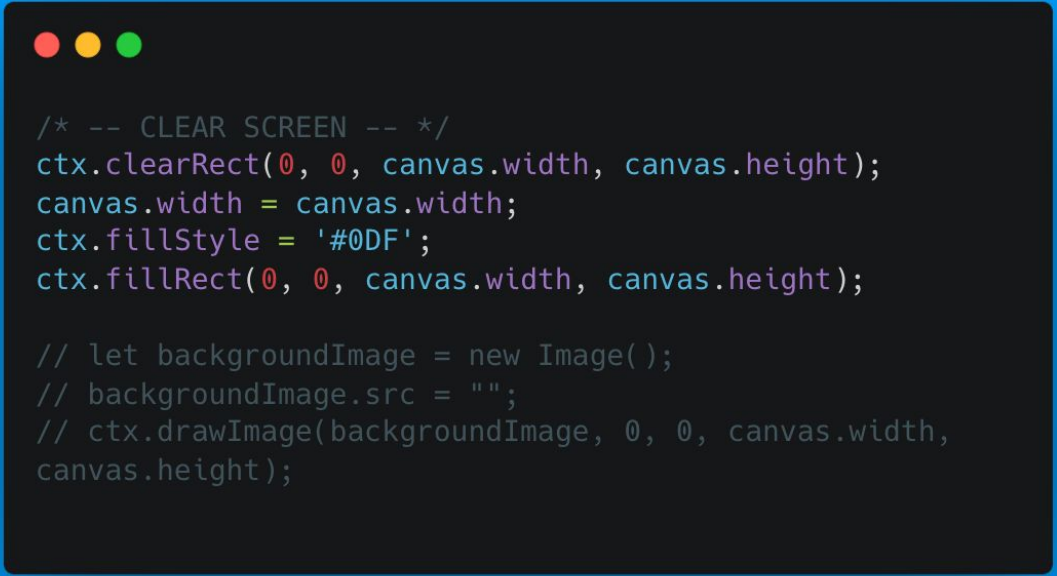
Load an Image





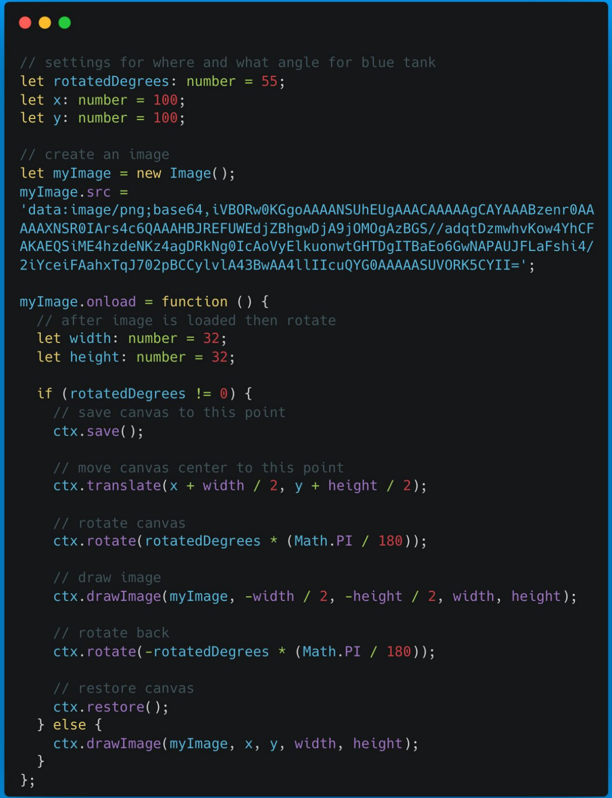
```
let cannon: string =
'data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAACAA
AAAgCAYAAABzenr0AAAAAXNSR0IArs4c6QAAAG9JREFUWEdjZBh
gwDjA9jOMOoDsEPC8KPUfOfq26z8jyyyyNIEsHnXAgIUAusXo2Z
jUtEByGhh1wGgIjIYAzUOAkAWUVt/o5QRGOTBgDqC1xbhKTHgID
JgD6G0xekgwjjpgNARGQ2A0BEZDYKBDAAB9wlqFoTfm5AAAAABJ
RU5ErkJgggAA';

let myImage = new Image();
myImage.src = cannon;

/* image, x, y, width, height */
ctx.drawImage(myImage, 50, 50, 32, 32);
```

Canvas Background



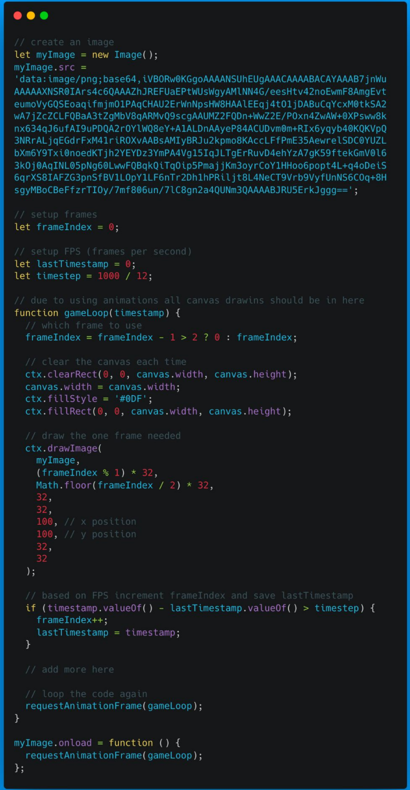
```javascript
/* -- CLEAR SCREEN -- */
ctx.clearRect(0, 0, canvas.width, canvas.height);
canvas.width = canvas.width;
ctx.fillStyle = '#0DF';
ctx.fillRect(0, 0, canvas.width, canvas.height);

// let backgroundImage = new Image();
// backgroundImage.src = "";
// ctx.drawImage(backgroundImage, 0, 0, canvas.width,
canvas.height);
```

Rotate an Image



```
// settings for where and what angle for blue tank
let rotatedDegrees: number = 55;
let x: number = 100;
let y: number = 100;

// create an image
let myImage = new Image();
myImage.src =
'data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAACAAAAAgCAYAAABzenr0AA
AAAXNSR0IArs4c6QAAAHBJREFUWEdjZBhgwDjA9jOMOgAzBGS//adqtDzmwhvKow4YhCF
AKAEQSiME4hzdeNKz4agDRkNg0IcAoVyElkuonwtGHTDgITBaEo6GwNAPAUJFLaFshi4/
2iYceiFAahxTqJ702pBCCylvlA43BwAA4llIIcuQYG0AAAAASUVORK5CYII=';

myImage.onload = function () {
  // after image is loaded then rotate
  let width: number = 32;
  let height: number = 32;

  if (rotatedDegrees != 0) {
    // save canvas to this point
    ctx.save();

    // move canvas center to this point
    ctx.translate(x + width / 2, y + height / 2);

    // rotate canvas
    ctx.rotate(rotatedDegrees * (Math.PI / 180));

    // draw image
    ctx.drawImage(myImage, -width / 2, -height / 2, width, height);

    // rotate back
    ctx.rotate(-rotatedDegrees * (Math.PI / 180));

    // restore canvas
    ctx.restore();
  } else {
    ctx.drawImage(myImage, x, y, width, height);
  }
};
```

Sprite Animation



```javascript
// create an image
let myImage = new Image();
myImage.src =
'data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAACAAAABACAYAAAB7jnWu
AAAAAXNSR0IArs4c6QAAAZhJREFUaEPtWUsWgyAMlNN4G/eesHtv42noEwmF8AmgEvt
eumoVyGQSEoaqifmjmO1PAqCHAU2ErWnNpsHW8HAAlEEqj4tO1jDABuCqYcxM0tkSA2
wA7jZcZCLFQBaA3tZgMbV8qARMvQ9scgAAUMZ2FQDn+WwZ2E/POxn4ZwAW+0XPsww8k
nx634qJ6ufAI9uPDQA2rOYlWQ8eY+A1ALDnAAyeP84ACUDvm0m+RIx6yqyb40KQKVpQ
3NRrALjqEGdrFxM41riROXvAABsAMIyBRJu2kpmo8KAccLFfPmE35AewrelSDC0YUZL
bXm6Y9Txi0noedKTjh2YEYDz3YmPA4Vg15IqJLTgErRuvD4ehYzA7gK59ftekGmV0l6
3kOj0AqINL05pNg60LwwFQBqkQiTqOip5PmajjKm3oyrCoY1HHoo6popt4L+q4oDeiS
6qrXS8IAFZG3pnSfBV1LOpY1LF6nTr2Dh1hPRiljt8L4NeCT9Vrb9VyfUnNS6COq+8H
sgyMBoCBeFfzrTIOy/7mf806un/7lC8gn2a4QUNm3QAAAABJRU5ErkJggg==';

// setup frames
let frameIndex = 0;

// setup FPS (frames per second)
let lastTimestamp = 0;
let timestep = 1000 / 12;

// due to using animations all canvas drawins should be in here
function gameLoop(timestamp) {
  // which frame to use
  frameIndex = frameIndex - 1 > 2 ? 0 : frameIndex;

  // clear the canvas each time
  ctx.clearRect(0, 0, canvas.width, canvas.height);
  canvas.width = canvas.width;
  ctx.fillStyle = '#0DF';
  ctx.fillRect(0, 0, canvas.width, canvas.height);

  // draw the one frame needed
  ctx.drawImage(
    myImage,
    (frameIndex % 1) * 32,
    Math.floor(frameIndex / 2) * 32,
    32,
    32,
    100, // x position
    100, // y position
    32,
    32
  );

  // based on FPS increment frameIndex and save lastTimestamp
  if (timestamp.valueOf() - lastTimestamp.valueOf() > timestep) {
    frameIndex++;
    lastTimestamp = timestamp;
  }

  // add more here

  // loop the code again
  requestAnimationFrame(gameLoop);
}

myImage.onload = function () {
  requestAnimationFrame(gameLoop);
};
```

How to Explore More of Canvas

Mozilla Developer (MDN)

When googling how to do things if you add MDN at the end it will give you resources to help.

Example:
Create a circle in canvas MDN

W3Schools

The same technique for searching the web for help can apply with W3Schools. Or you can search up Canvas W3Schools for a list of resources to help

ChatGPT

You can always ask ChatGPT a specific question and it will show you how to accomplish it. You can even ask it to correct your code.

Assignment

Your assignment is to create on Canvas about 3-5 different shapes and images. They can depict a simple game scene or they can be random objects on the canvas. Please show at next class for a reward.