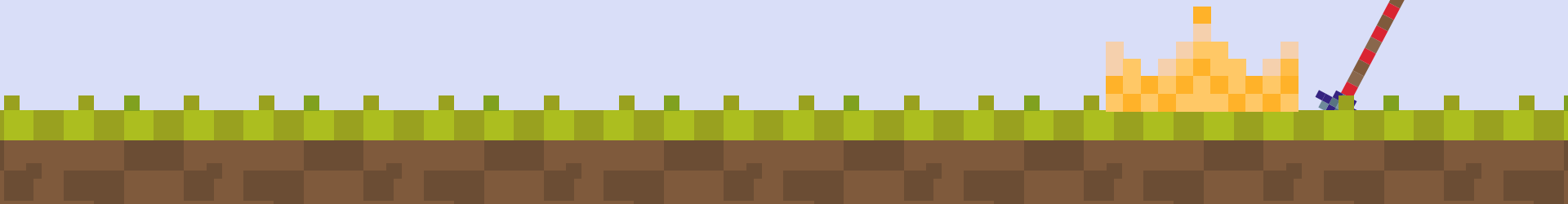




02.

# Intro to HTML5 Canvas

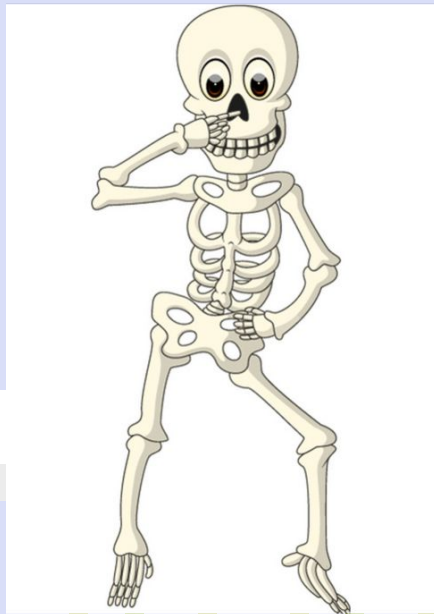
Overview of HTML5 and 2D game development, canvas coordinate system, drawing shapes, animations.





# How Web Pages work

HTML



CSS



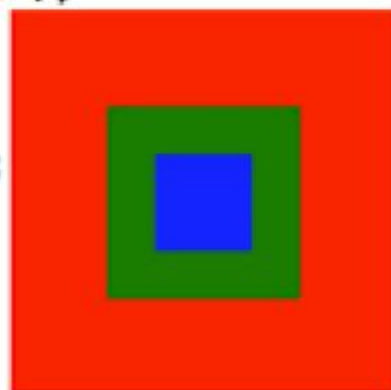
JavaScript





# HTML5 Canvas

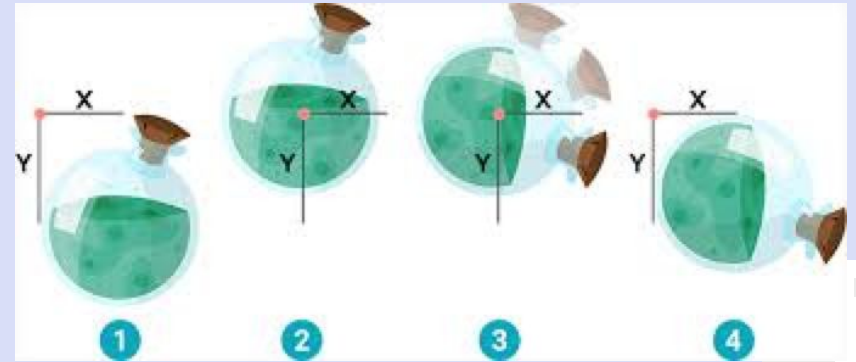
```
<html>
<body>
  <canvas id="myCanvas" width="200" height="200" />
  <script>
    var canvas = document.getElementById("myCanvas");
    var ctx = canvas.getContext("2d");
    ctx.fillStyle = "red";
    ctx.fillRect(0, 0, 200, 200);
    ctx.fillStyle = "green";
    ctx.fillRect(50, 50, 100, 100);
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





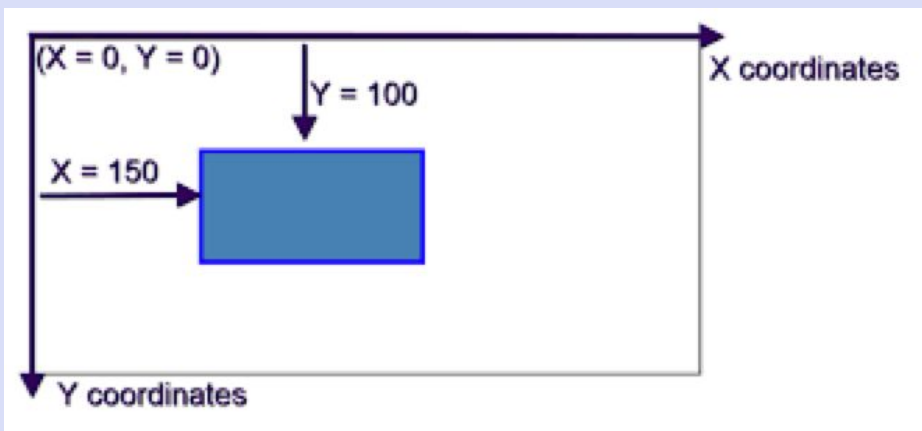
# TABLET APP

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

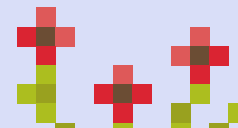




# Canvas Coordinate System



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





# Open up StackBlitz

The screenshot displays the StackBlitz web editor interface for a Vite project. The interface is divided into several panels:

- Menu Bar:** Located at the top, it includes options for Fork, Share, and a user profile icon. It also shows the project name "Vitejs (forked)" and a dropdown menu.
- Activity Bar:** On the left side, it contains icons for switching between views: Explorer, Search, Source Control, and Settings.
- Side Bar:** This panel shows the project's file structure under the "FILES" section. The files listed are: `.vscode`, `public`, `src`, `_gitignore`, `index.html`, `package-lock.json`, `package.json`, `README.md`, `tsconfig.json`, `tsconfig.node.json`, and `vite.config.ts`.
- Editor:** The central area for writing code. It shows the `index.html` file with the following content:

```
<!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>
```
- Terminal:** Located at the bottom left, it shows the output of the Vite development server:

```
vite v2.8.6 dev server running at:
> Local: http://localhost:3000/
> Network: use '--host' to expose

ready in 1265ms.
```
- Preview:** On the right side, it shows a live preview of the application. The address bar indicates the URL `vitejs-vite-aahkib--3000`. The preview area is currently blank.
- Console:** Located at the bottom right, it shows the output of the development server:

```
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

```
● ● ●  
  
// 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);
```

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

```
● ● ●  
  
* {  
  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);

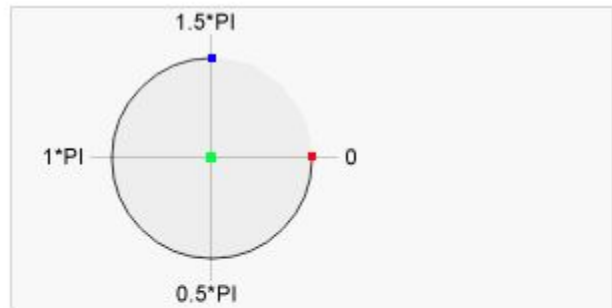
  // draw the shape at the new position
  ctx.fillStyle = 'red';
  ctx.fillRect(x, 50, 50, 50);

  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();
```



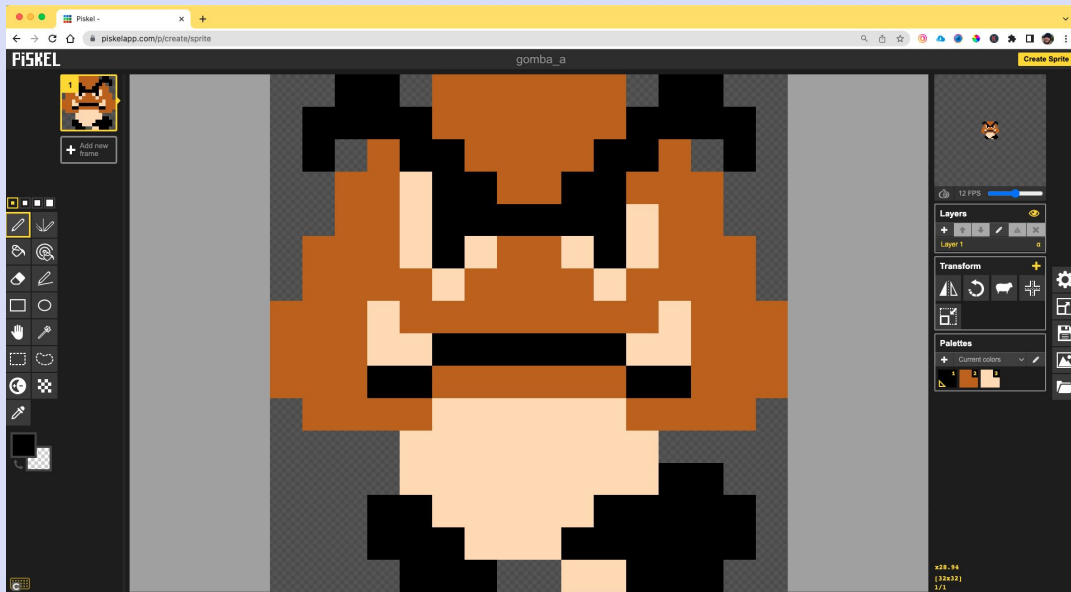
# Create a Triangle



```
ctx.beginPath();  
ctx.moveTo(100, 100);  
ctx.lineTo(125, 125);  
ctx.lineTo(75, 125);  
ctx.fillStyle = "red";  
ctx.fill();
```



# Load an Image



```
let cannon: string =  
'data:image/png;base64,iVBORw0KGgoAAAANSUHEugAAACAA  
AAAgCAYAAABzenr0AAAAAXNSR0IArs4c6QAAAG9JREFUWEdjZBh  
gwDjA9j0M0oDsEPC8KPuf0fq26z8jyyyyNIEsHnXAgIUAusXo2Z  
jUtEByGhh1wGgIjIYAzU0AkAWUVt/o5QRG0TBgDqC1xbhKTHgID  
JgD6G0xekgwjjpgNARGQ2A0BEZDYKBDAAB9wlqFoTfm5AAAAABJ  
RU5ErkJgggAA';  
  
let myImage = new Image();  
myImage.src = cannon;  
  
/* image, x, y, width, height */  
ctx.drawImage(myImage, 50, 50, 32, 32);
```



# Add Text



```
@import url('https://fonts.googleapis.com/css2?family=Press+Start+2P&display=swap');
```



```
ctx.font = "14px 'Press Start 2P'";  
ctx.fillStyle = '#FFE737';  
ctx.fillText('PacMan is coming', 100, 200);
```



# Canvas Background



```
/* -- 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,iVBORw0KGgoAAAANSUHEUgAAACAAAAgCAYAAABzenr0AA
AAAXNSR0IArs4c6QAAAHBJREFUWEdjZBhgwDjA9jOM0gAzBGS//adqtDzmwhvKow4YhCF
AKAEQSiME4hzdeNKz4agDRkNg0IcAoVyElkuonwtGHTDgITBaEo6GwNAPAUJFLaFshi4/
2iYceiFAahxTqJ702pBCCylvLA43BwAA4lIIcuQYG0AAAAASUVORK5CYII=';

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



```
// create an image
let myImage = new Image();
myImage.src =
'data:image/png;base64,iVBORw0KGgoAAAANSUHEUgAAACAAABACAYAAAB7jnwU
AAAAAXNSR0IArs4c60AAAZhJREFUePTUuSgyAmlNM4G/eesHtv42noEvmF8AngEvt
eumoVY6QEOaqIfejm01PAqCHAUZErmNp8rW8HAALEEq4t01jDABuCqrcxM0tKSA2
wa7jZcCLF0Ba3tZgmV8QAPWQ9scjAMUZFQDn4WZ2E/PDn4tEwM48P2adK
nx634q36uFAI9UP0QAZrOYLW0BeY+A1ALDnAAYeP84ACUDvm8m+R1x6qyb48KKVPQ
3NRrALjgEGrFXM41rLRDXvAABsAMiyBRJu2kpmo8KacclFFmE35AewrE1SDCOYUzL
DXm0YITxL8noeKTjhZYEDz3TmPA4Vg15IqJLTgErRuv04ehYZA7gK59fteKgnV016
3kG18AtIML05pplRkLwFQ8nGLTg0Lp5PmAJKn3oyrCg7IMho0pplAL-e4n0e1S
6grXS01AFZG3pn5fBV1OpY1LF6nTz20lhPRLLjL8L4McT9Vrb9VyYunNSRCQq+8H
sgyMBoCBefzrTIOy/7mf886um/7LCgn2a4QUm3QAAAABJRUSErkJggg==';

// setup frames
let frameIndex = 0;

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

// due to using animations all canvas draws 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 = '#00F';
  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

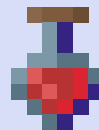
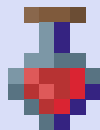
  // loop the code again
  requestAnimationFrame(gameLoop);
}

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:

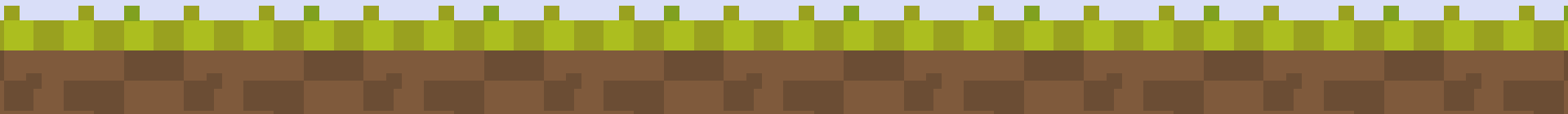
Create a circle in canvas MDN

## 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.

