

HTML Graphics

Graphics HOME

HTML Plotting

- Plot Graphics
- Plot Canvas
- Plot Plotly
- Plot Chart.js
- Plot Google
- Plot D3.js

Google Maps

- Maps Intro
- Maps Basic
- Maps Overlays
- Maps Events
- Maps Controls
- Maps Types
- Maps Reference

SVG Tutorial

- SVG Intro
- SVG in HTML
- SVG Rectangle
- SVG Circle
- SVG Ellipse
- SVG Line
- SVG Polygon
- SVG Polyline
- SVG Path
- SVG Text
- SVG Stroking
- SVG Filters Intro
- SVG Blur Effects
- SVG Drop Shadows
- SVG Linear
- SVG Radial
- SVG Examples
- SVG Reference

Canvas Tutorial

- Canvas Intro
- Canvas Drawing
- Canvas Coordinates
- Canvas Lines
- Canvas Shapes
- Canvas Rectangles
- Canvas Circles
- Canvas Curves
- Canvas Gradients
- Canvas Text
- Canvas Images

Canvas Clock

- Clock Intro
- Clock Face
- Clock Numbers
- Clock Hands
- Clock Start

HTML Game

- Game Intro
- Game Canvas
- Game Components
- Game Controllers
- Game Obstacles
- Game Score
- Game Images
- Game Sound
- Game Gravity
- Game Bouncing
- Game Rotation
- Game Movement

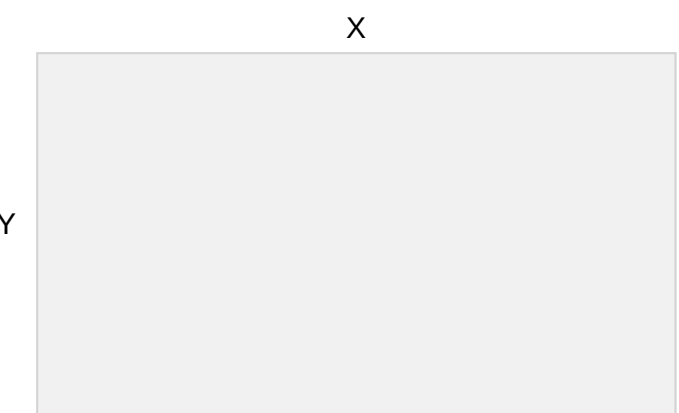
Try it Yourself »

Change the Position

We use x- and y-coordinates to position components onto the game area.

The upper-left corner of the canvas has the coordinates (0,0)

Mouse over the game area below to see its x and y coordinates:



You can position the components wherever you like on the game area:

Example

```
function startGame() {  
  myGameArea.start();  
  myGamePiece = new component(30, 30, "red", 2, 2);  
}
```

Try it Yourself »

Many Components

You can put as many components as you like on the game area:

Example

```
var redGamePiece, blueGamePiece, yellowGamePiece;  
  
function startGame() {  
  redGamePiece = new component(75, 75, "red", 10, 10);  
  yellowGamePiece = new component(75, 75, "yellow", 50, 60);  
  blueGamePiece = new component(75, 75, "blue", 10, 110);  
  myGameArea.start();  
}  
  
function updateGameArea() {  
  myGameArea.clear();  
  redGamePiece.update();  
  yellowGamePiece.update();  
  blueGamePiece.update();  
}
```

Try it Yourself »

Moving Components

Make all three components move in different directions:

Example

```
function updateGameArea() {  
  myGameArea.clear();  
  redGamePiece.x += 1;  
  yellowGamePiece.x += 1;  
  yellowGamePiece.y += 1;  
  blueGamePiece.x += 1;  
  blueGamePiece.y -= 1;  
  redGamePiece.update();  
  yellowGamePiece.update();  
  blueGamePiece.update();  
}
```

Try it Yourself »

◀ Previous

Log in to track progress

Next ▶

Spaces

Upgrade

Newsletter

Get Certified

Report Error

Top Tutorials

- HTML Tutorial
- CSS Tutorial
- JavaScript Tutorial
- How To Tutorial
- SQL Tutorial
- Python Tutorial
- W3.CSS Tutorial
- Bootstrap Tutorial
- PHP Tutorial
- Java Tutorial
- C++ Tutorial
- jQuery Tutorial

Top References

- HTML Reference
- CSS Reference
- JavaScript Reference
- SQL Reference
- Python Reference
- W3.CSS Reference
- Bootstrap Reference
- PHP Reference
- HTML Colors
- Java Reference
- Angular Reference
- jQuery Reference

Top Examples

- HTML Examples
- CSS Examples
- JavaScript Examples
- How To Examples
- SQL Examples
- Python Examples
- W3.CSS Examples
- Bootstrap Examples
- PHP Examples
- Java Examples
- XML Examples
- jQuery Examples

Get Certified

- HTML Certificate
- CSS Certificate
- JavaScript Certificate
- Front End Certificate
- SQL Certificate
- Python Certificate
- PHP Certificate
- jQuery Certificate
- Java Certificate
- C++ Certificate
- C# Certificate
- XML Certificate

