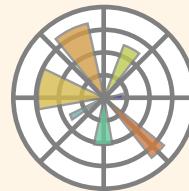


```
import numpy as np  
  
ax.set_title("Sine Wave")
```



# MATPLOTLIB CHEAT SHEET

SCALER  
*Topics*

```
plt.scatter(x, y)  
plt.show()
```

```
plt.barh(x, y)
```

# What is Matplotlib?

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

**We can create various types of plots in Matplotlib:**

- (1) Line Plots
- (2) Scatter Plots
- (3) Bar Plots
- (4) Histograms
- (4) Pie Charts
- (5) Box Plots
- (6) Heatmaps
- (7) 3D Plotting

## Installation

### Prerequisite:

An array is a collection of elements identified by index or key values. The elements are stored contiguously in memory.

Open your command prompt and run the following command:

```
pip install matplotlib
```

Install numpy and pandas as well (Recommended):

```
pip install numpy pandas
```

# Basic Plotting

## Plotting Lines and Markers

### Import required libraries

```
import numpy as np  
import matplotlib.pyplot as plt
```

### Create empty figure and plot

```
fig, ax = plt.subplots()
```

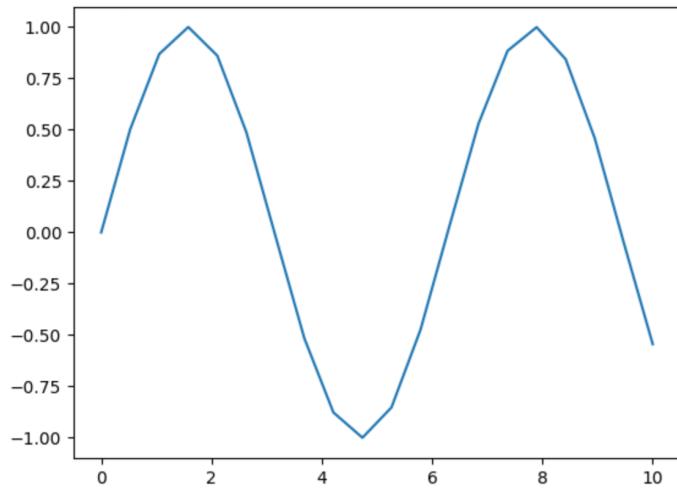
### Create the data using numpy

```
x = np.linspace(0, 10, 100)  
y = np.sin(x)
```

### Plot and Display

```
ax.plot(x, y)  
plt.show()
```

### Output



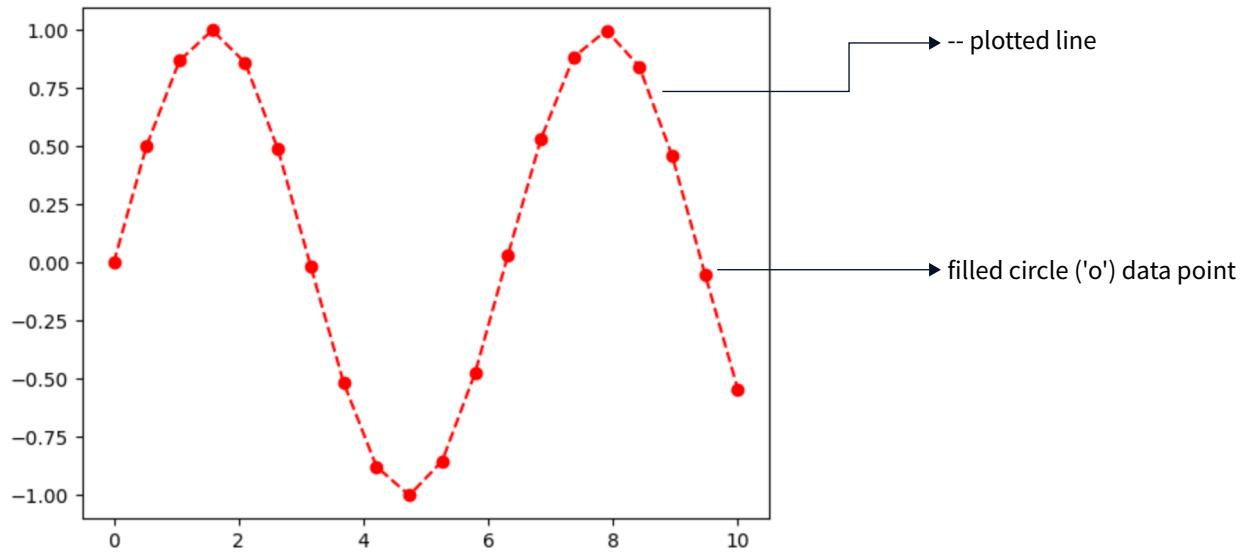
## plot() Options

- \* **linestyle:** Specifies the line style
- \* **color:** Specifies color of the plotted lines and markers
- \* **marker:** Specifies marker options to plot data points.

### Example

```
ax.plot(x, y, linestyle='--', color='red', marker='o')
```

## Output



Commonly used values of linestyle:

- \* '-' -> Solid line
- \* '--' -> Dashed line
- \* '-.' -> Dash-dot line
- \* ':' -> Dotted line

Commonly used values of marker:

1. '!' - Point marker
2. 'o' - Circle marker
3. 's' - Square marker
4. 'p' - Pentagon marker
5. '\*' - Star marker
6. 'h' - Hexagon1 marker
7. '+' - Plus marker
8. 'x' - Cross marker
9. 'D' - Diamond marker

Commonly used values of color:

- \* 'red'
- \* 'blue'
- \* 'green'
- \* 'orange'
- \* 'purple'

You can use single characters as well:

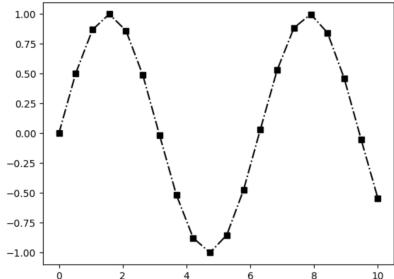
- \* 'r' for red
- \* 'b' for blue
- \* 'g' for green
- \* 'k' for black

You can use hexadecimal color code as well:

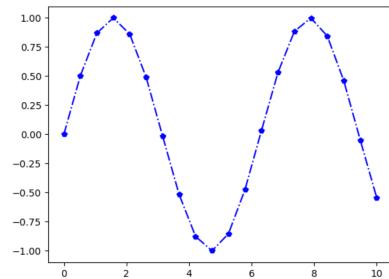
- \* '#fa0714'
- \* '#0ffa07' and so on.

## More Examples

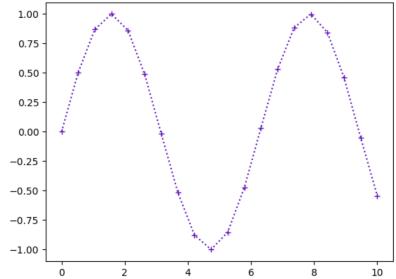
```
ax.plot(x, y, linestyle='-.',  
color='black', marker='s')  
plt.show()
```



```
ax.plot(x, y, linestyle='-.',  
color='blue', marker='p')  
plt.show()
```



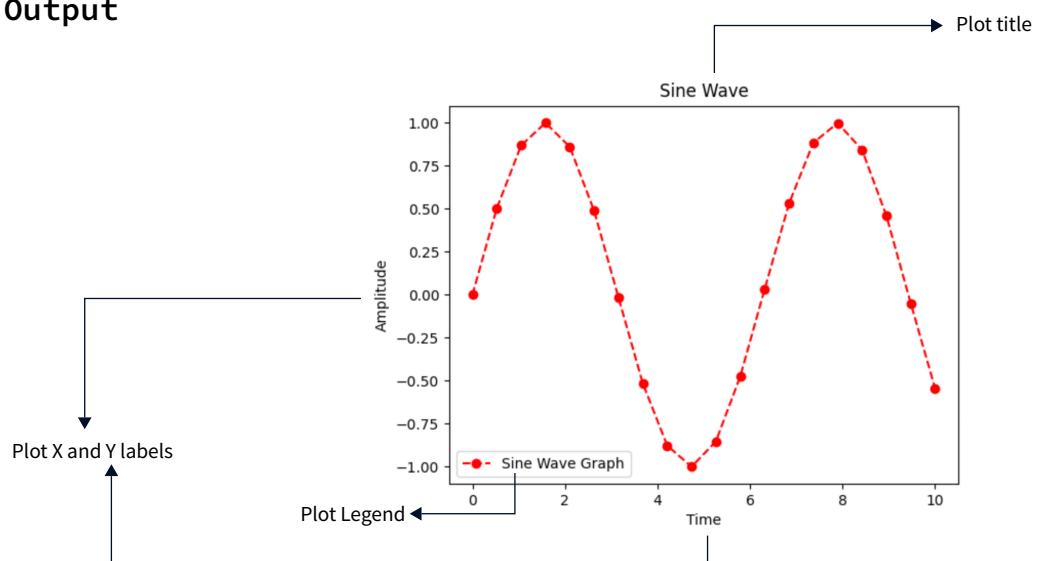
```
ax.plot(x, y, linestyle=':',  
color='#5e0fb8', marker='+')  
plt.show()
```



## Plot title, Labels and Legends

```
// Assign labels to the axes  
ax.set_xlabel("Time")  
ax.set_ylabel("Amplitude")  
  
// Legend  
ax.legend(["Sine Wave Graph"])  
  
// Plot title  
ax.set_title("Sine Wave")
```

## Output



# Line Plots

## Basic Line Plots and Multiple Lines on the Same Plot

### Create the data using numpy

```
import numpy as np
import matplotlib.pyplot as plt

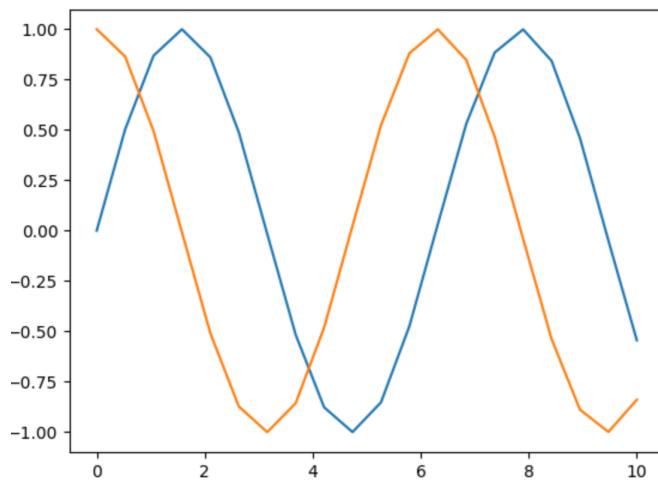
x = np.linspace(0, 10, 100)
y1 = np.sin(x)
y2 = np.cos(x)
```

### Plot the lines

```
ax.plot(x, y1, label='Sine')
ax.plot(x, y2, label='Cosine')
```

### Output

```
plt.show()
```

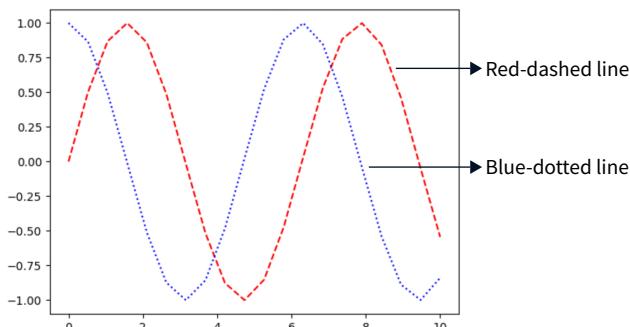


## Line Styles and Colors

### Examples

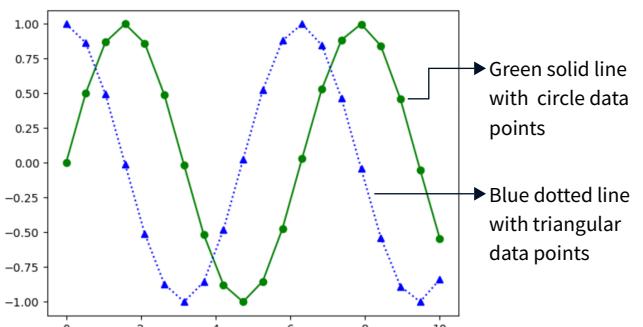
```
ax.plot(x, y1, 'r--', label='Sine')
# Red dashed line

ax.plot(x, y2, 'b:', label='Cosine')
# Blue dashed line
```



```
ax.plot(x, y1, 'go-', label='Sine')
# Green solid line with circle markers

ax.plot(x, y2, 'b^:', label='Cosine')
# Blue dotted line with triangle_up markers
```



# Scatter Plots

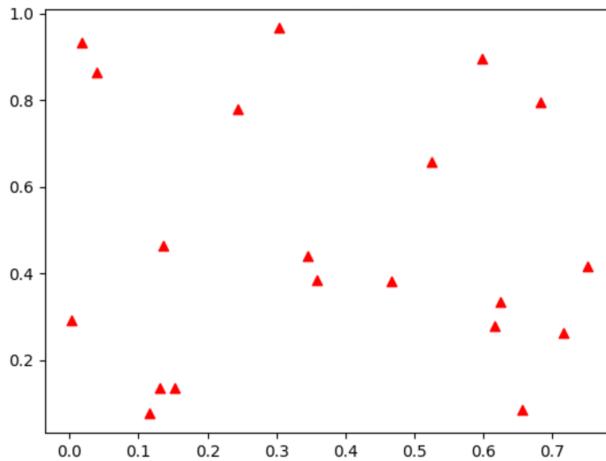
## Basic Line Plots and Multiple Lines on the Same Plot

Create the data using numpy

```
x = np.random.rand(20)           → Random x and y points  
y = np.random.rand(20)
```

Plot the data points

```
plt.scatter(x, y)  
plt.show()
```



## Adding color and size

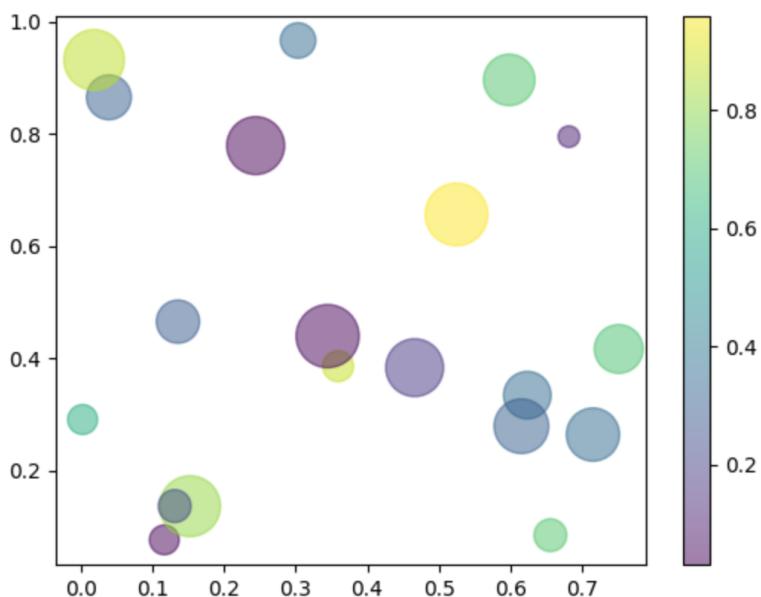
Create the colors and size arrays

```
colors = np.random.rand(50)          → Random colors and sizes  
sizes = 1000 * np.random.rand(50)
```

Plot with a Color Bar

```
plt.scatter(x, y, c=colors, s=sizes, alpha=0.5, cmap='viridis')  
plt.colorbar()                      → Adds a colorbar to the plot  
plt.show()
```

## Output



## Options

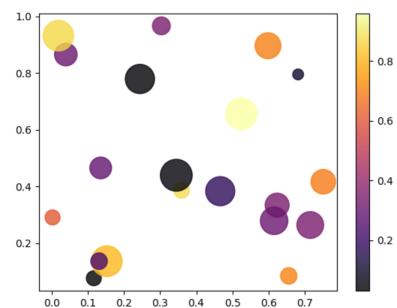
c: Specifies color for each data point  
s: Specifies size of the markers  
alpha: Specifies the transparency level of markers  
cmap: Specifies the colormap used for mapping

Commonly used values of cmap:

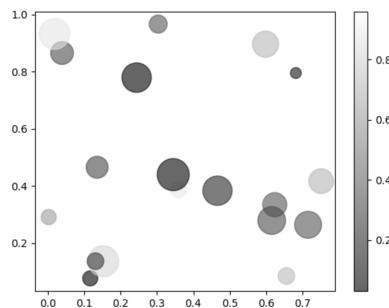
* plasma	* summer
* viridis	* cividis
* inferno	* hot
* magma	* winter
* cool	* gray

## More Examples

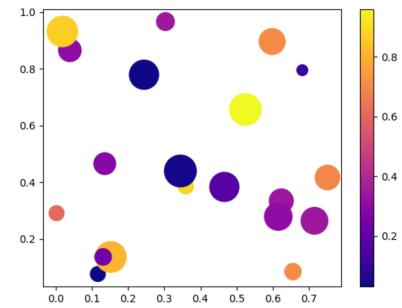
```
plt.scatter(x, y, c=colors,  
           s=sizes, alpha=0.8, cmap='inferno')  
  
plt.colorbar()  
plt.show()
```



```
plt.scatter(x, y, c=colors,  
           s=sizes, alpha=0.6, cmap='gray')  
  
plt.colorbar()  
plt.show()
```



```
plt.scatter(x, y, c=colors,  
           s=sizes, alpha=1, cmap='plasma')  
  
plt.colorbar()  
plt.show()
```



## Bar Plots

### Creating bar plots

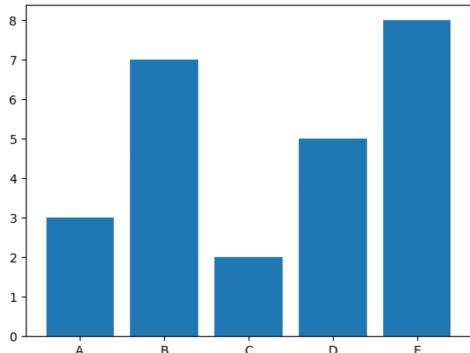
#### Create the data

```
x = ['A', 'B', 'C', 'D', 'E']
y = [3, 7, 2, 5, 8]
```

#### Plot the data points

```
plt.bar(x, y)
plt.show()
```

#### Output



### Grouped Bar Plots

#### Create the data

```
N = 5
men_means = (20, 35, 30, 35, 27)
women_means = (25, 32, 34, 20, 25)
```

#### Create the bar plot

```
ind = np.arange(N) # Create an array of numbers from 0 to N - 1
width = 0.35 # Width of the bars
plt.bar(ind, men_means, width, label='Men') # Plotting first bar at x values: 0 to N - 1
plt.bar(ind + width, women_means, width, label='Women') # Plotting second bar at x values + width
```

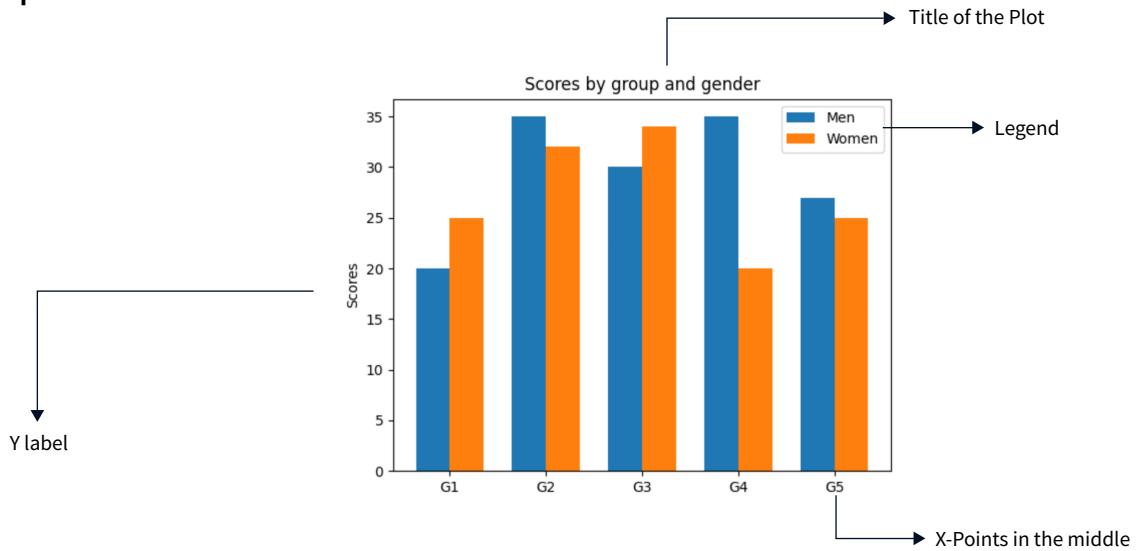
### Set label, legend and title

```
plt.ylabel('Scores')
plt.legend(loc='best')
plt.title('Scores by group and gender')
```

### Set x-ticks

```
# Create x-ticks in the middle of the bars.
plt.xticks(ind + width / 2, ('G1', 'G2', 'G3', 'G4', 'G5'))
plt.show()
```

### Output



## Stacked Bar Plots

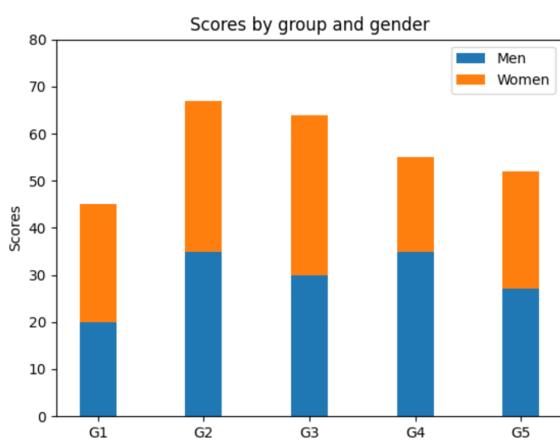
### Create the stacked bar plot

```
p1 = plt.bar(ind, men_means, width)
p2 = plt.bar(ind, women_means, width, bottom=men_means)
```

### Set legend, x and y ticks and plot

```
plt.xticks(ind, ('G1', 'G2', 'G3', 'G4', 'G5')) # Give x-ticks at ind points on x-axis
plt.yticks(np.arange(0, 81, 10)) # Give y-ticks at intervals of 10
plt.legend((p1[0], p2[0]), ('Men', 'Women'))
plt.show()
```

## Output

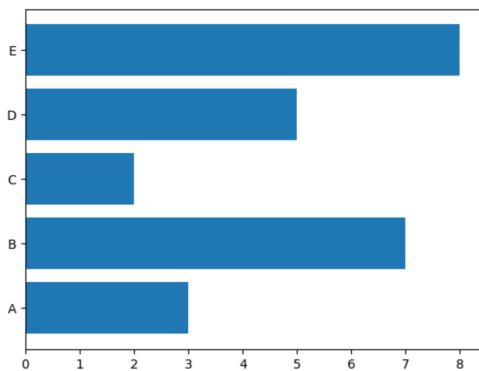


## Horizontal bar plots

Create the stacked bar plot

```
plt.barh(x, y)  
plt.show()
```

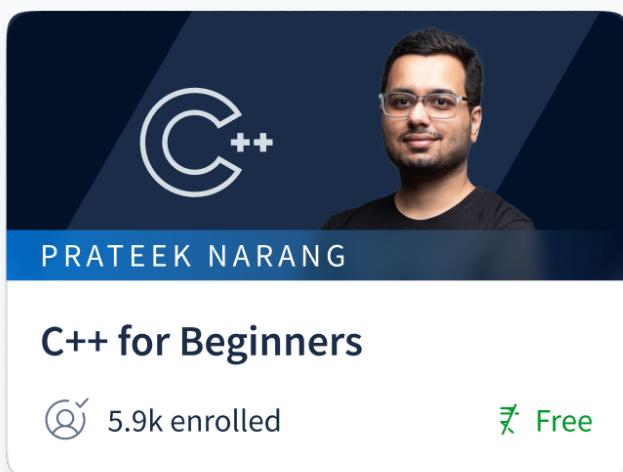
## Output



# SCALER TOPICS

Unlock your potential in software development with  
**FREE COURSES** from SCALER TOPICS!

Register now and take the first step towards your future Success!



**C++ for Beginners**

PRATEEK NARANG

5.9k enrolled

₹ Free



**Java for Beginners**

TARUN LUTHRA

6.8k enrolled

₹ Free

That's not it. Explore 20+ Courses by clicking below

Explore Other Courses

Practice **CHALLENGES**  
and become 1% better everyday



**CIFAR-10 Image Classification Using PyTorch**

Article

No. Of Questions : 3

[Go to Challenge >](#)



**How to Build a Snake Game in JavaScript?**

Article

No. Of Questions : 3

[Go to Challenge >](#)

Explore Other Challenges