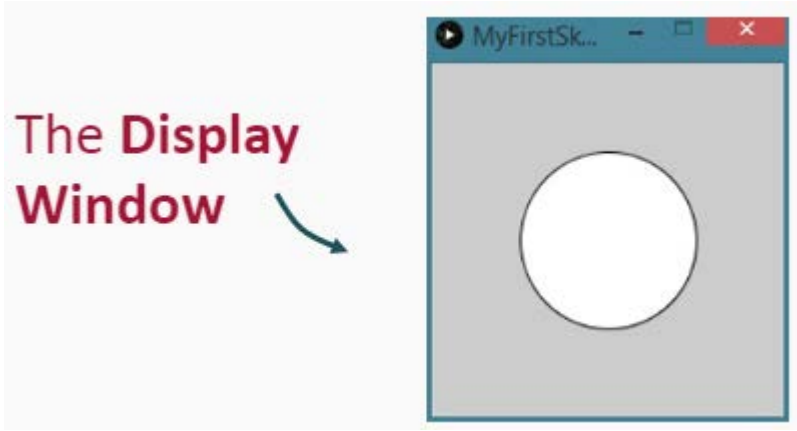
Drawing Functions

Processing Functions

- Processing contains many built-in **Functions** for drawing on the screen.
- **Functions** are the **actions** or **verbs** of the code sentence.
- **Functions** sometimes need details, called **Arguments** to perform their actions.
- When we use functions we say we are "**Calling**" the function.

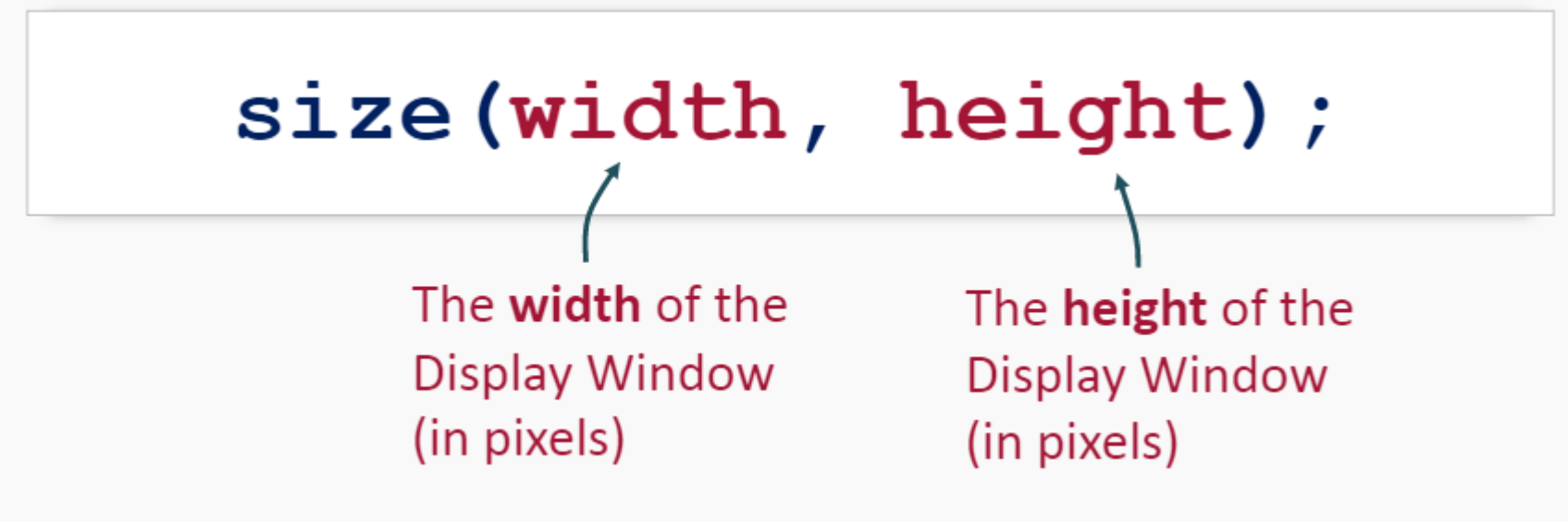
The Display Window

- Before we can draw, we must create a canvas. In Processing, this canvas is called the **Display Window.**
- The **size()** function sets the size of the display window we can draw on.
- The **Display Window** is measured in **pixels.**
- The default size is **100px** by **100px**.



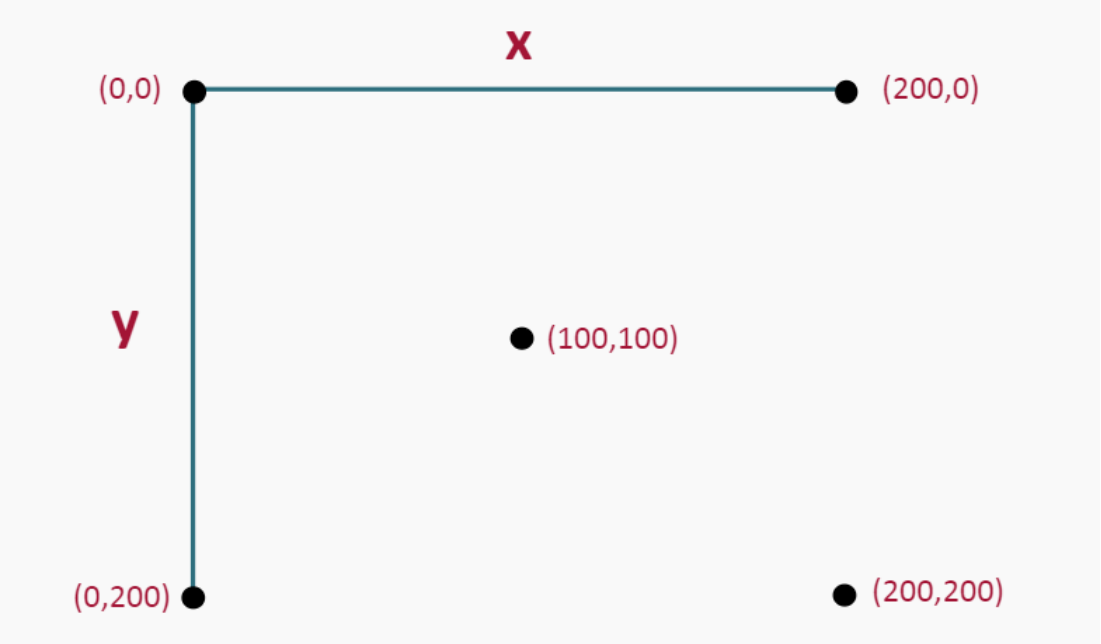
Setting the Window Size

- The **size()** built-in function defines the size of the display window in pixels.
- The **size()** function takes two **Arguments:**



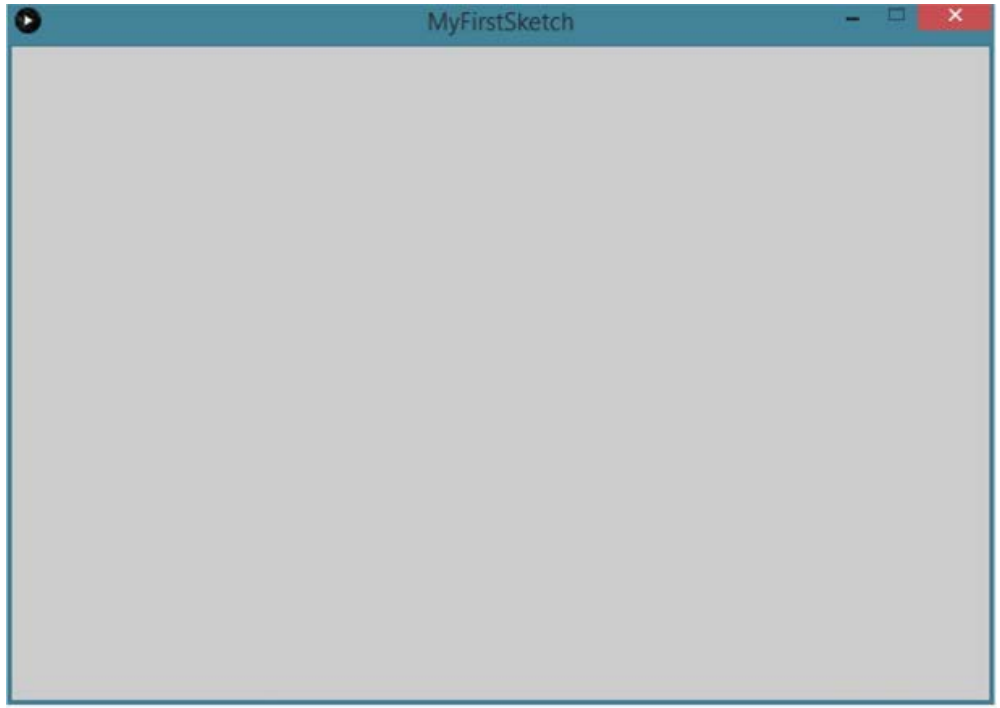
The Display Window

- The display window is like a grid. For example, a window that is 200px by 200px would look like this:

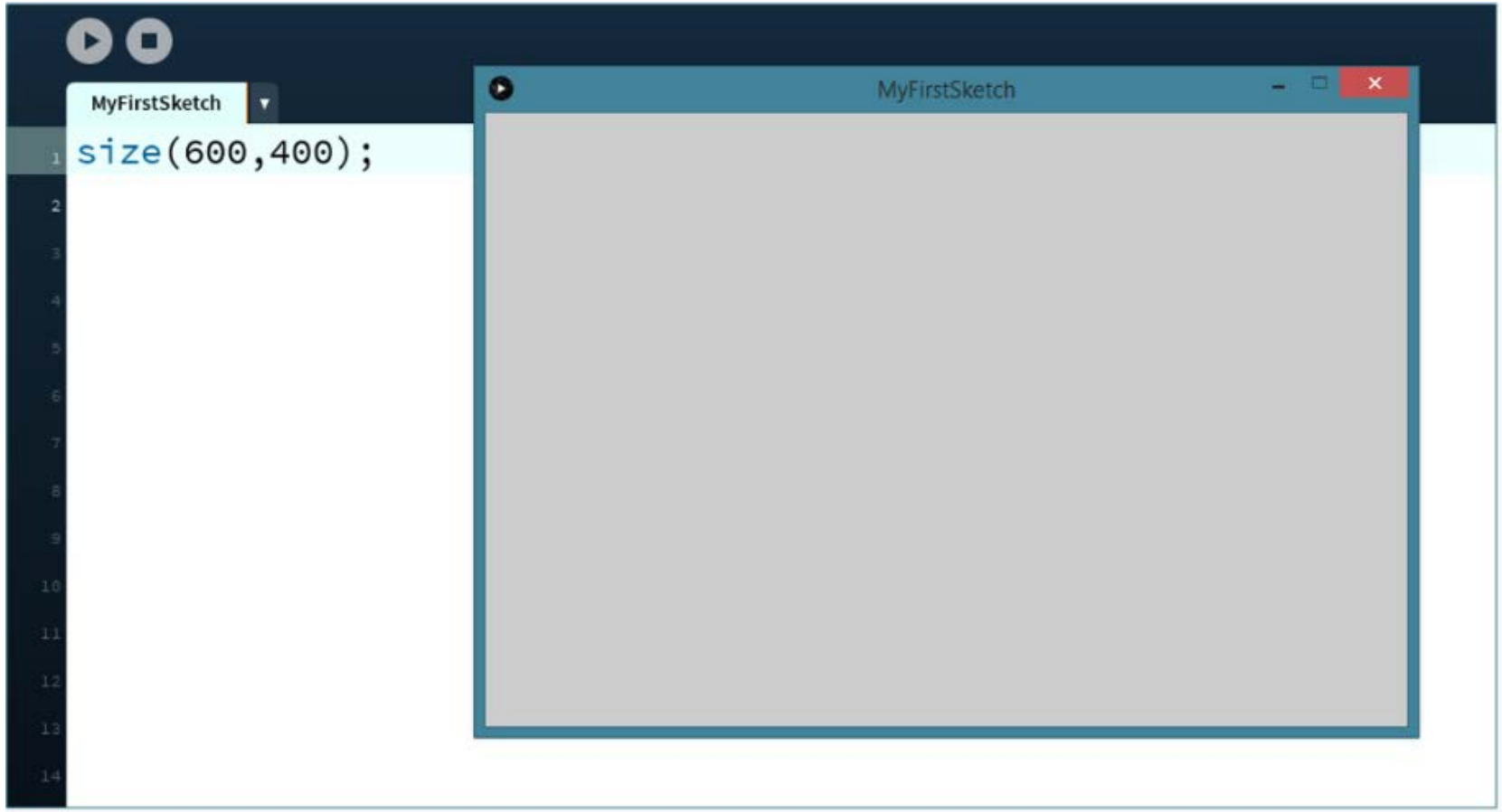


Try It Yourself

- Create a blank window that is 600px by 400px.

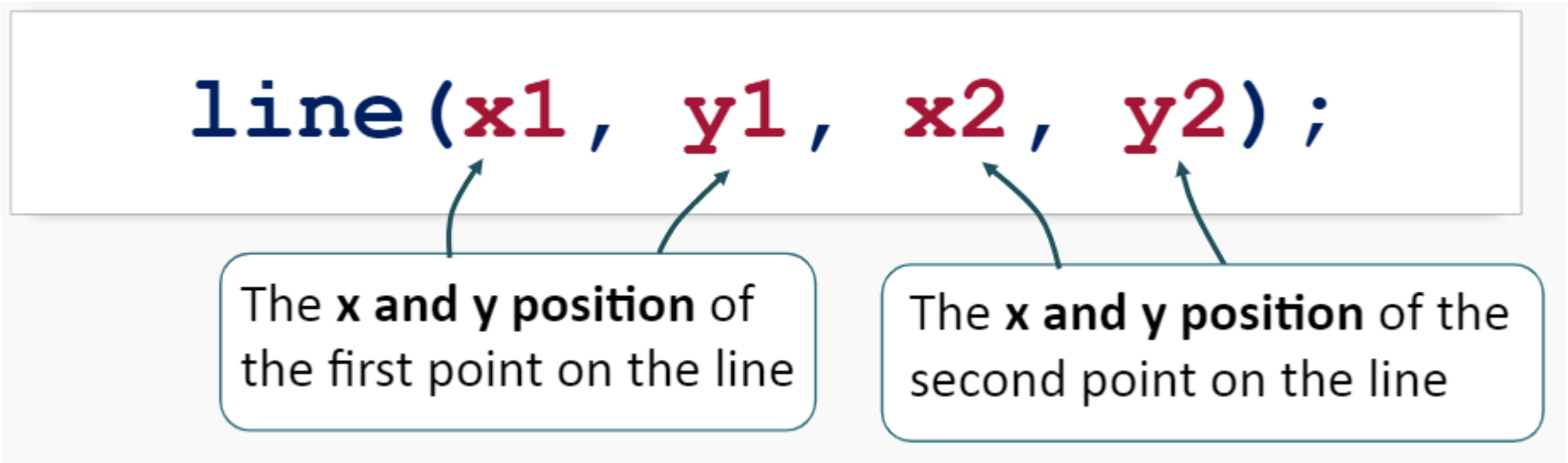


Try It Yourself Solution



Drawing a Line

- We can use the built in **line()** function to draw a line.
- The **line()** function needs four arguments:



Drawing a Line

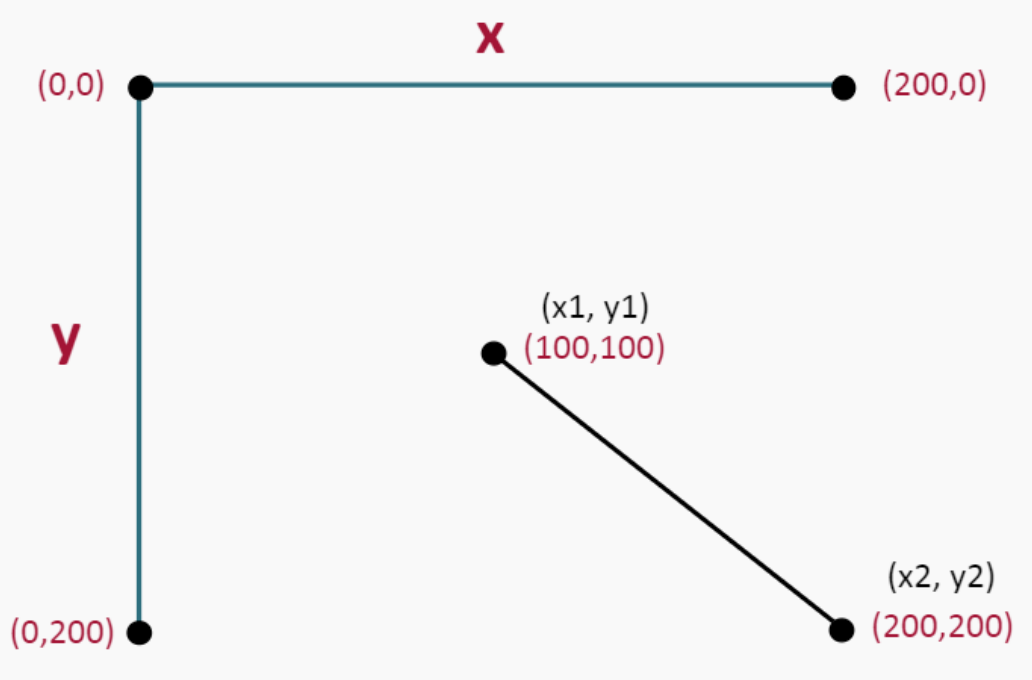
- For example, the code:

```
line(100, 100, 200, 200);
```

- Will draw a line from point (100,100) to point(200,200).
- Try it yourself in Processing

Drawing a Line

- In a window that is **200px** by **200px**, the line looks like this:



Drawing Lines Practice

- Fill in the blanks in the code snippet below to draw a line from point (100, 200) to point (300, 300).

```
size(800, 800);

line(                    );
```

Drawing Lines Practice

- Fill in the blanks in the code snippet below to draw a line from point (200, 200) to point (400, 300).

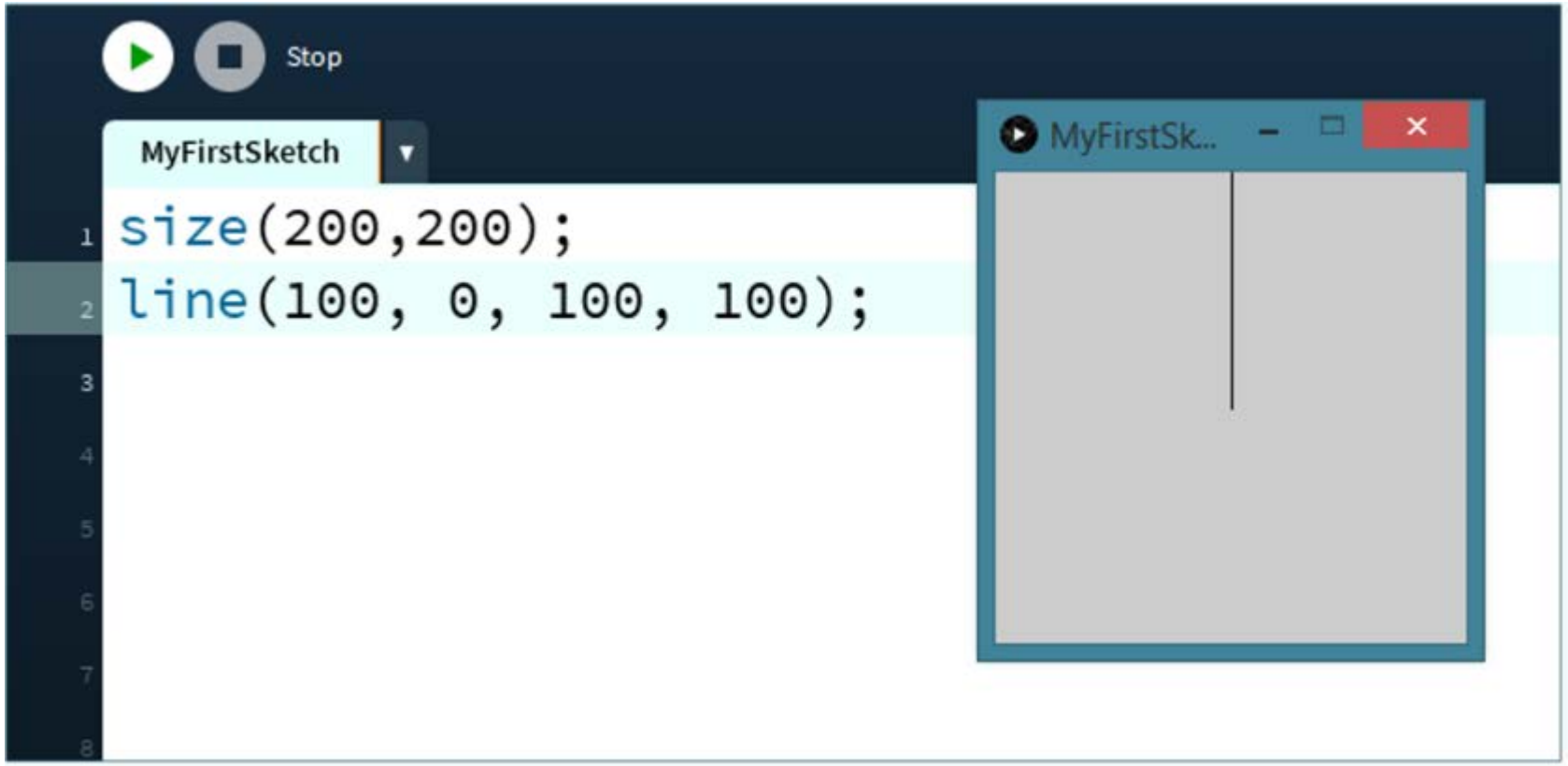
```
size(800, 800);

.
```

Try It Yourself

- Draw a line from point (100,0) to point(100,100) in a window that is 200px by 200px.

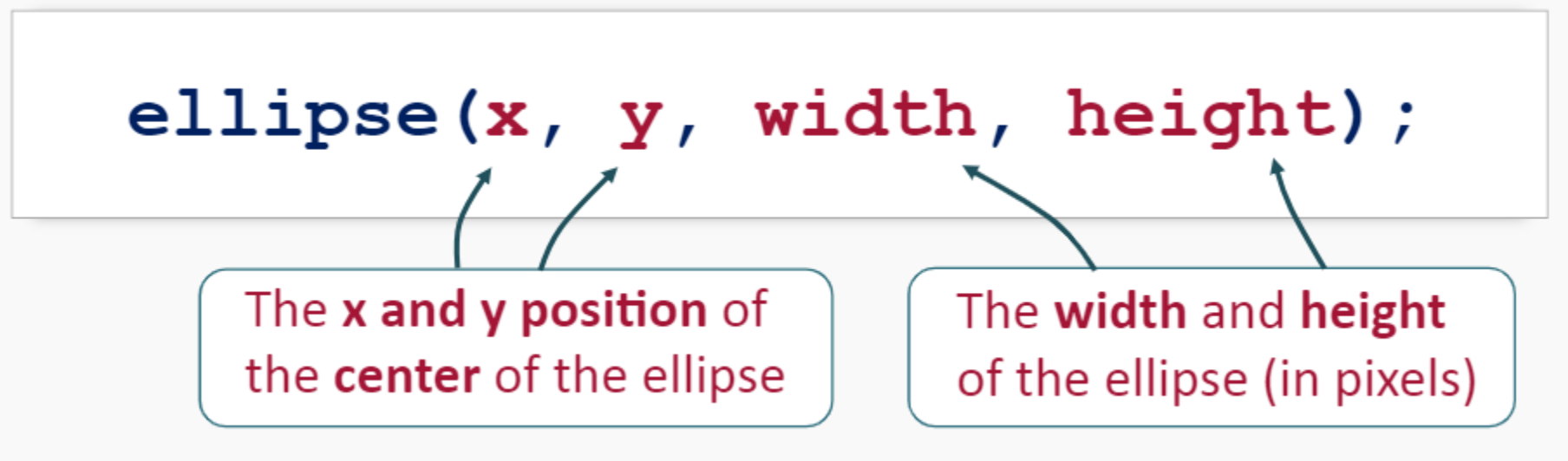
1. Set the window size to 200px by 200px
2. Draw a line from point (100,0) to point (100,100).

Try It Yourself Solution



Drawing an Ellipse

- The **ellipse()** function draws ovals and circles.
- This function takes 4 arguments:



Drawing an Ellipse

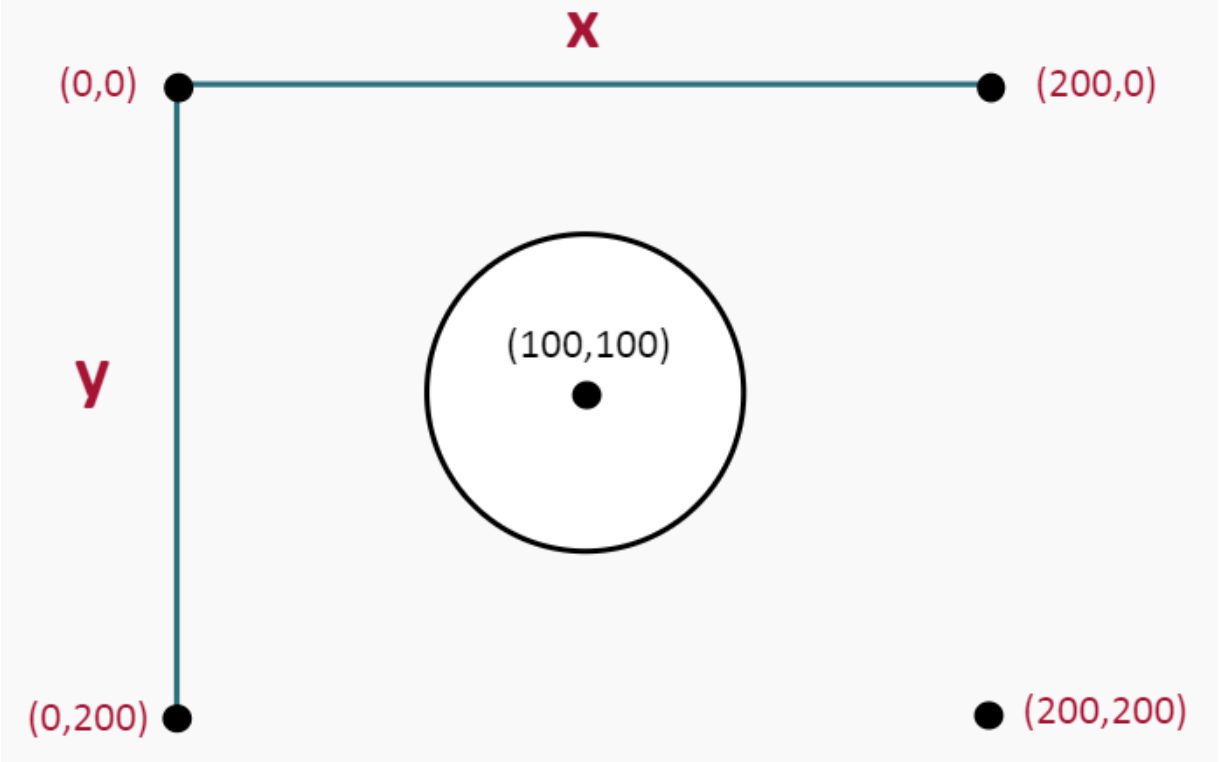
- For example, the code:

```
ellipse(100, 100, 100, 100);
```

- Will draw an ellipse with the centre at the point (100, 100) that is 100px wide and 100px high.
- Try it yourself in Processing

Drawing an Ellipse

- On a stage that is 200px by 200px, the ellipse looks like this:



Drawing an Ellipse Practice

- Fill in the blanks in the code snippet below to create an ellipse at point (100, 200) that is 200px wide and 100px high.

```
size(400, 400);

_____(____,____,____,____);
```

Drawing an Ellipse Practice

- Fill in the blanks in the code snippet below to create an ellipse that is 100px wide by 100px high in the centre of the display window.

```
size(400, 400);

(  ,  ,  ,  );
```

Answer

```
ellip… (200 , 200 , 100 , 100 );
```

Drawing an Ellipse Practice

- Fill in the blanks in the code snippet below to create an ellipse at point (0,0) that is 150px wide by 150px high in a window that is 600px by 600px.

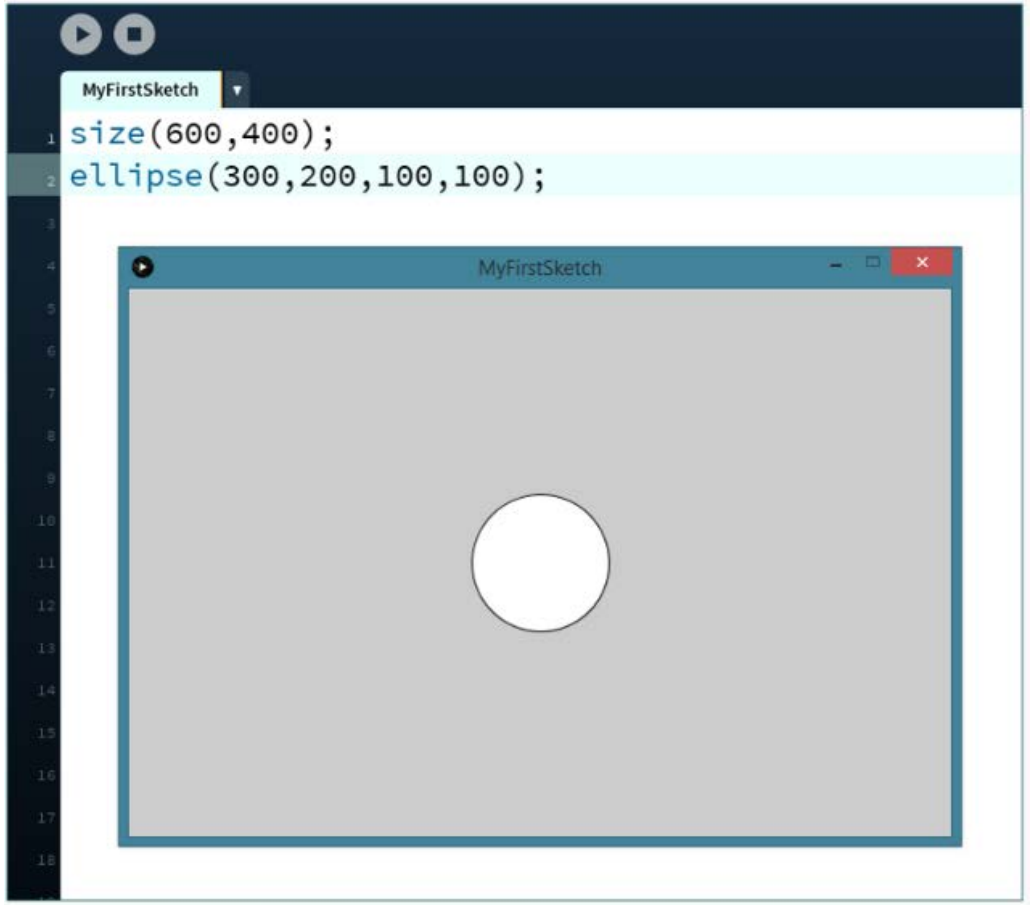
```
size(          ,          );

          (     ,     ,     ,     );
```

Try It Yourself

- In a window that is 600px by 400px, draw an ellipse at the point (300, 200) that is 100px wide by 100px high.

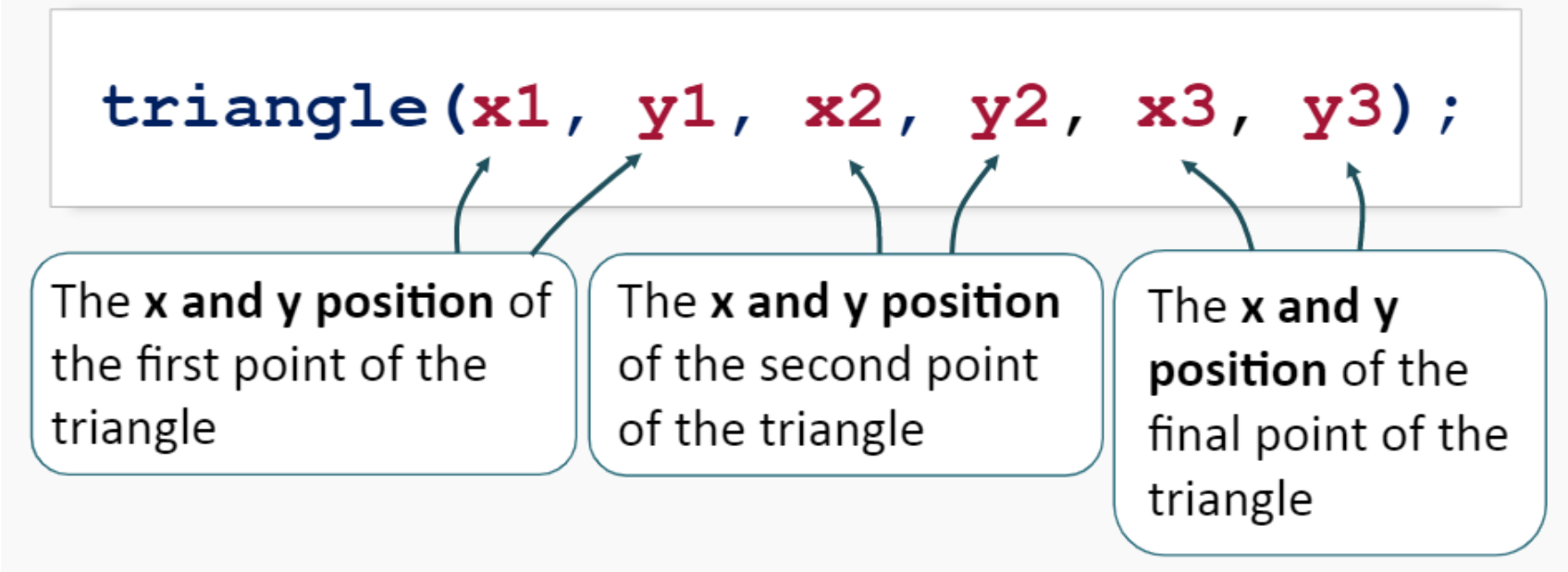
1. Set the size of the window to 600px by 400px.
2. Draw an ellipse that is 100px by 100px at point (300, 200)

Try It Yourself Solution



Drawing a Triangle

- We can use the built in **triangle** function to draw a triangle.
- The **triangle()** function needs **six** arguments:



Drawing a Triangle

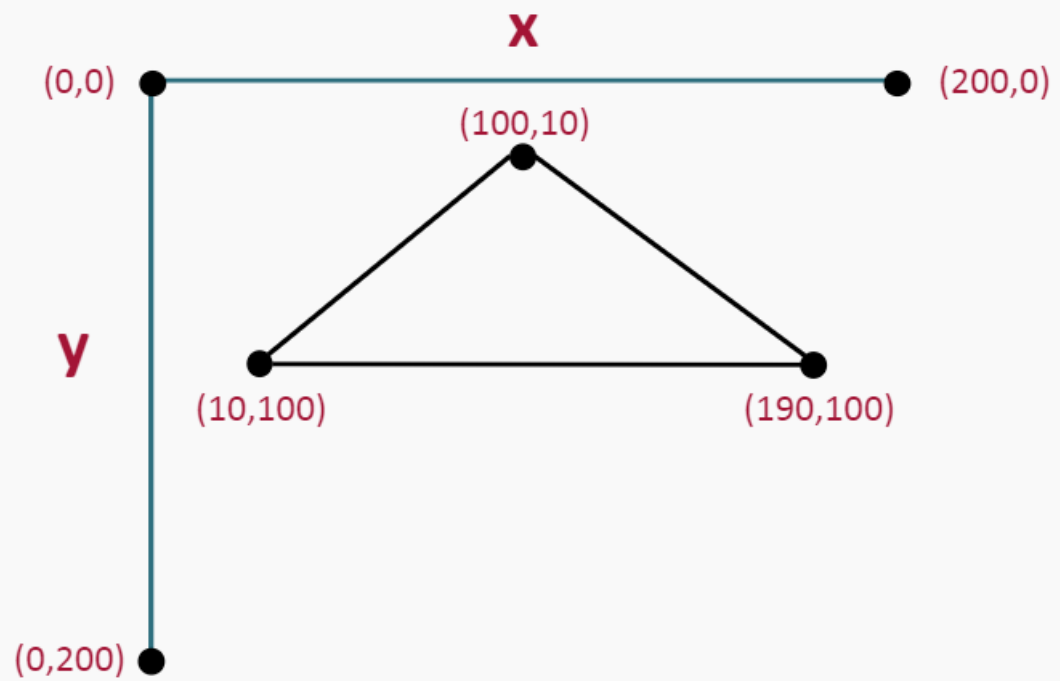
- For example, the code:

```
triangle(10,100,100,10,190,100);
```

- Will draw a line from point (10,100) to point (100,10) and then to point(190,100) creating a triangle.
- Try it yourself in Processing.

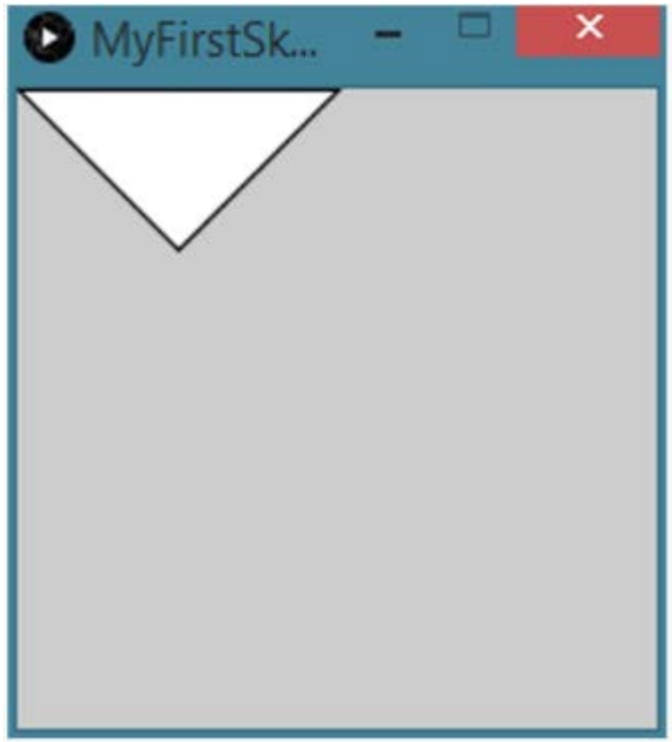
Drawing a Triangle

```
triangle(10,100,100,10,190,100);
```

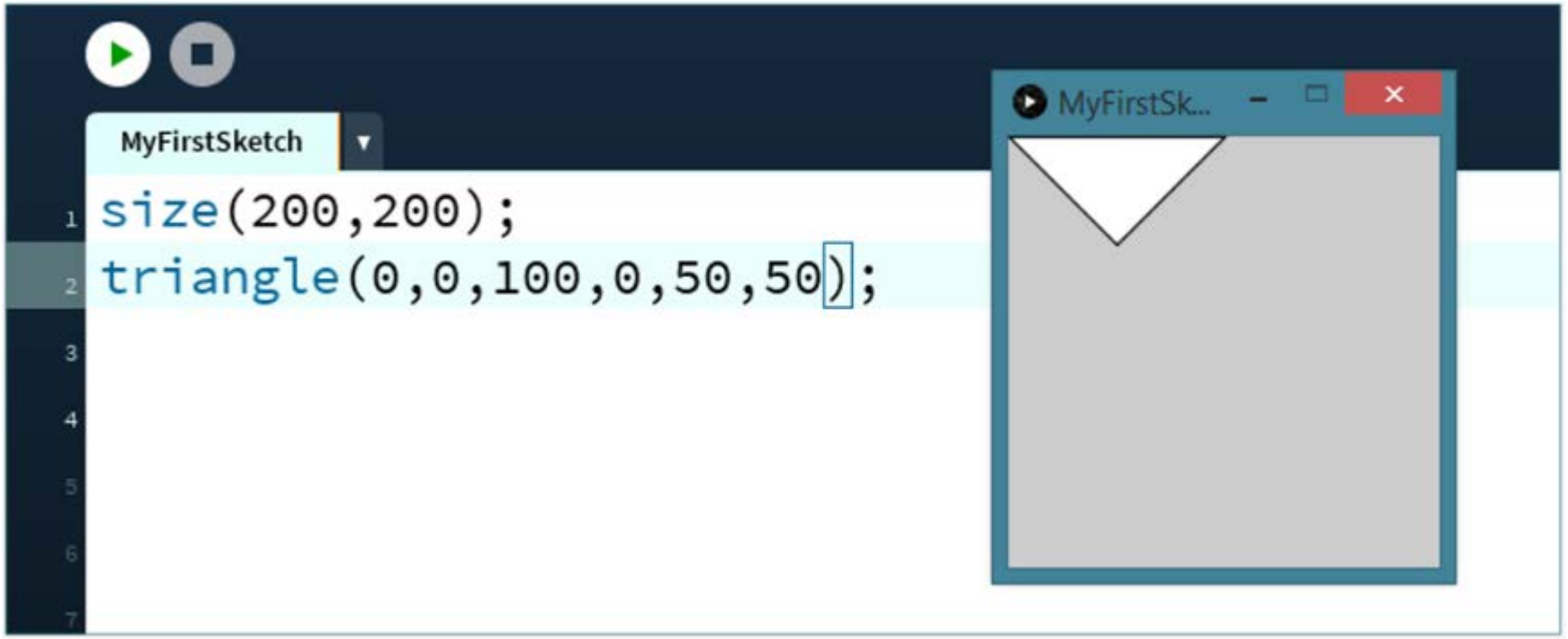


Try It Yourself

- Draw a triangle with points at: (0,0) , (100,0) and (50,50).

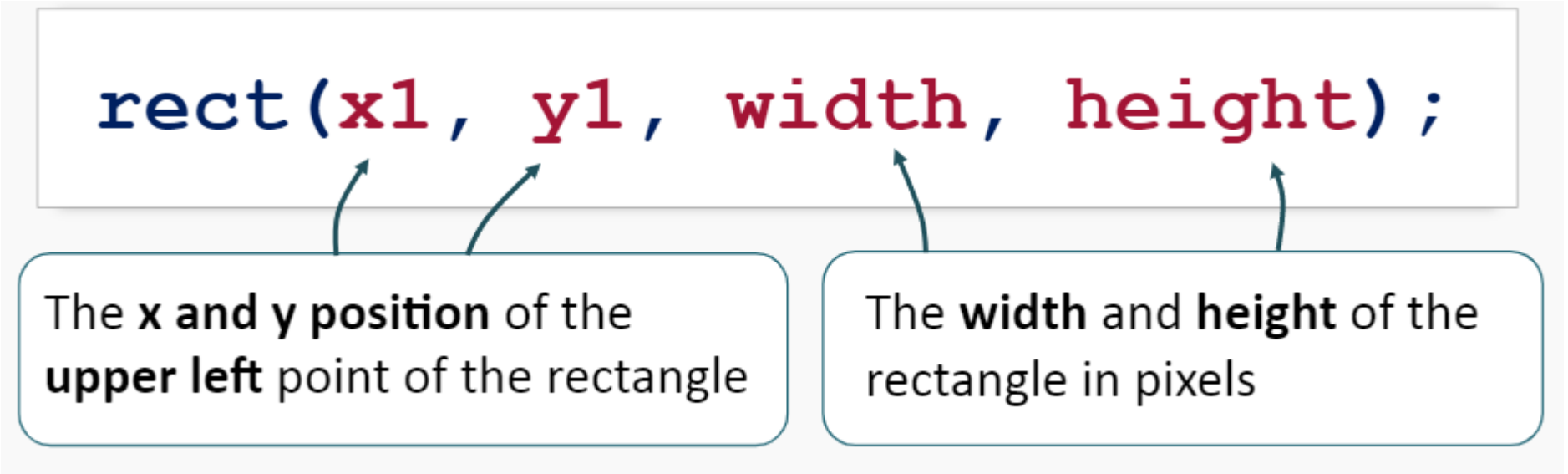


Try It Yourself Solution



Drawing a Rectangle

- We can use the built in **rect()** function to draw a rectangle.
- The **rect()** function needs **four** arguments:



Drawing a Rectangle

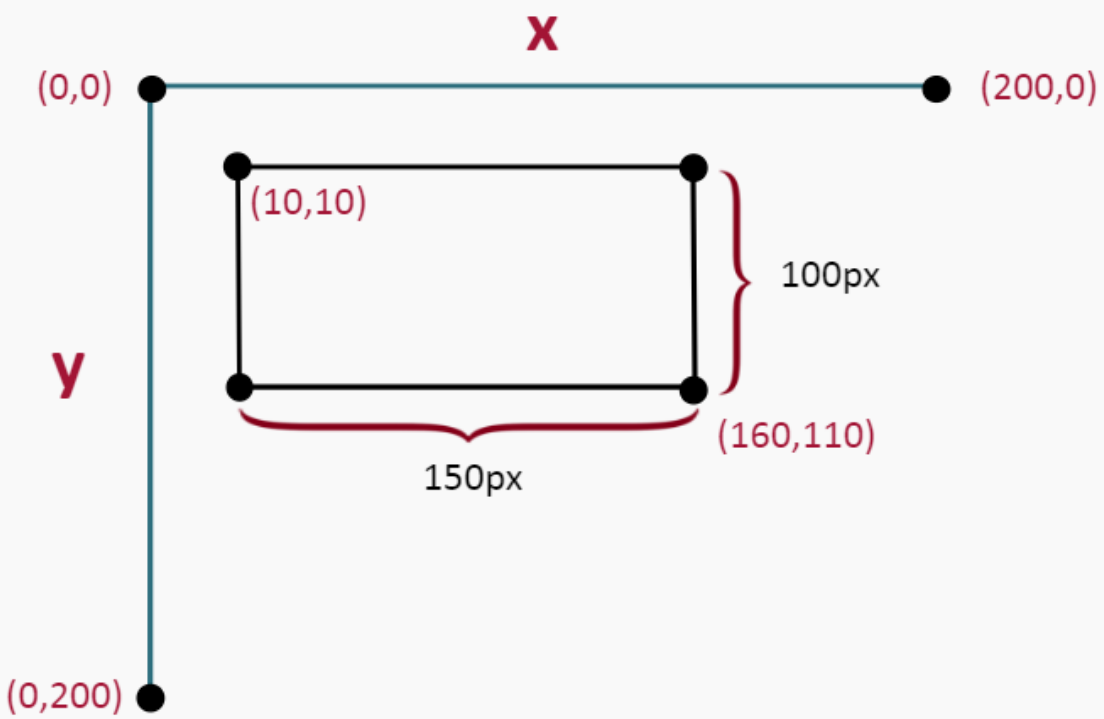
- For example, the code:

```
rect(10,10,150,100);
```

- Will draw a rectangle that has a top left point at (10,10) and has a width of 150px and a height of 100px.
- Try it yourself in Processing.

Drawing a Rectangle

```
rect(10,10,150,100);
```



Drawing Rectangles Practice

- Fill in the blanks in the code snippet below to draw a rectangle that has a corner at (100,100) and is 300px wide and 200px tall.

```
size(800, 800);

rect(                    );
```

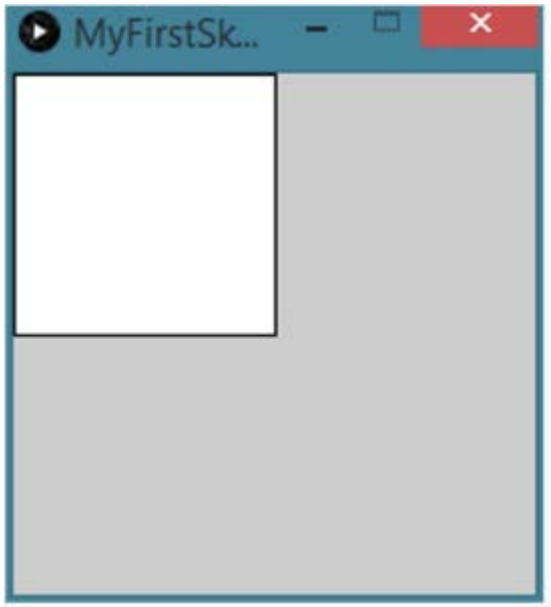
Drawing Rectangles Practice

- Fill in the blanks in the code snippet below to draw a rectangle that has a corner at (0,0) and is 500px wide and 600px tall.

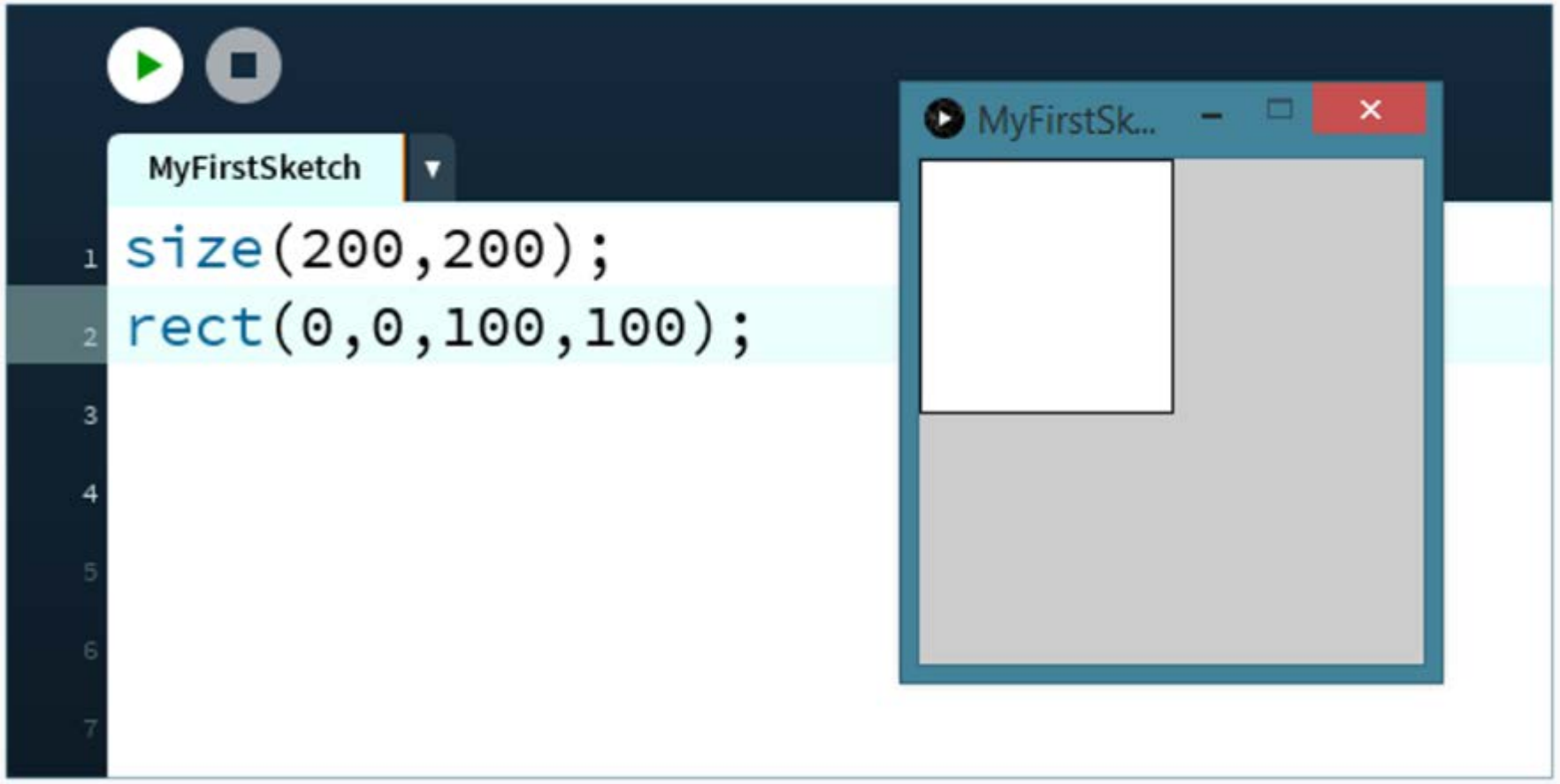
```
size(800, 800);

.(                                    )
```

Try It Yourself

- Draw a rectangle with the top left corner at (0,0) with a width of 100px and a height of 100px.

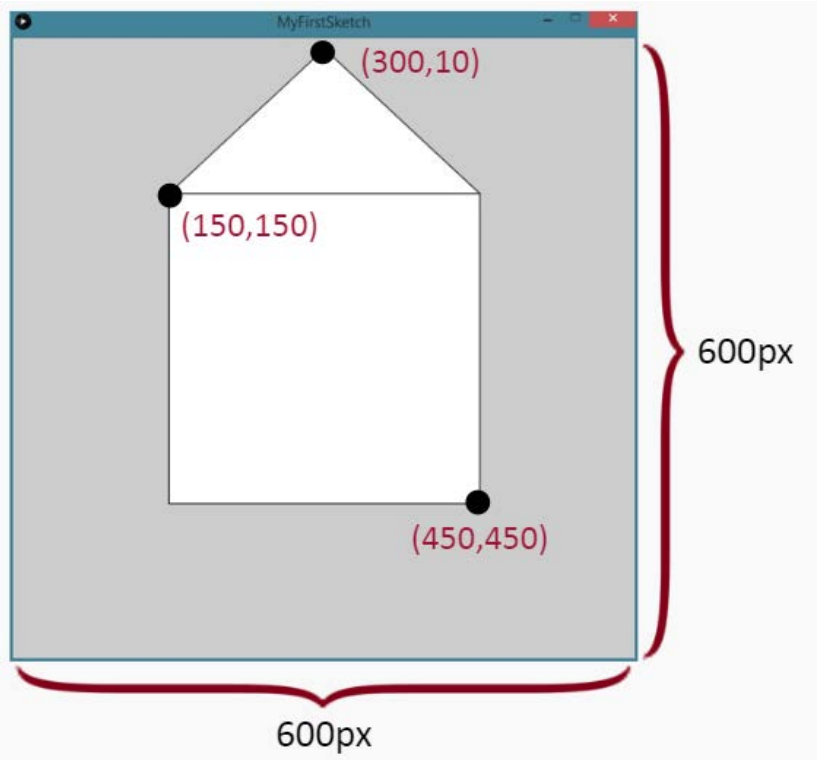


Try It Yourself Solution



Try It Yourself Challenge

- Challenge! Use the built in drawing functions to draw:



Summary

- Change the size of the display window:

```
size(width, height)
```

- Draw an ellipse:

```
ellipse(x, y, width, height)
```

- Draw a line from point (x1, y1) to (x2, y2):

```
line(x1, y1, x2, y2)
```

Summary

- Draw a triangle from (x1, y1) to (x2, y2) to (x3, y3):

```
triangle(x1, y1, x2, y2, x3, y3)
```

- Draw a rectangle with the top left point (x, y):

```
rect(x, y, width, height)
```