

# CSC 240

# Computer Graphics

Sara Mathieson  
Fall 2016  
Smith College

# Outline: 9/14

- Line algorithms (+ handout)
- Intro to HTML and JavaScript

Admin: Office Hours

- Lab 1

**Monday 4-5pm (location TBD)**  
**Tuesday 4-5pm (346 Ford)**

**TA hours Sun-Tues:**  
**7:30-9:30pm (241 Ford)**

# One more graphics application

Graphic Design Junction

AaBbCcDdEeFfGgHhIi

JjKkLlMmNnOoPpQqRr

SsTtUuVvWwXxYyZz

1234567890

[graphicdesignjunction.com](http://graphicdesignjunction.com)

AaBbCcDdEeFfGgHhIi

JjKkLlMmNnOoPpQqRr

SsTtUuVvWwXxYyZz

1234567890

[www.crafthubs.com](http://www.crafthubs.com)

ABCDEFGHIJKLM  
NOPQRSTUVWXYZ

0123456789!?

&\$@\*(/)

[smashinghub.com](http://smashinghub.com)

# Intro to HTML

(mostly used as a wrapper for JavaScript)

# Intro to HTML

(Hyper Text Markup Language)

```
<!DOCTYPE html>  
<html>  
<head>  
<title>Canvas Graphics</title>
```

```
</head>  
<body onload="init()">  
    <canvas id="theCanvas" width="640" height="480"></canvas>  
</body>  
</html>
```

# Intro to HTML

(Hyper Text Markup Language)

```
<!DOCTYPE html>
<html>
<head>
<title>Canvas Graphics</title>
```

HTML tags (start tag and end tag)



```
</head>
<body onload="init()">
    <canvas id="theCanvas" width="640" height="480"></canvas>
</body>
</html>
```

# Intro to HTML


(Hyper Text Markup Language)

```
<!DOCTYPE html>
<html>
<head>
<title>Canvas Graphics</title>
```

HTML tags (start tag and end tag)



after webpage has loaded (text, images, etc), call this function



```
</head>
<body onload="init()">
  <canvas id="theCanvas" width="640" height="480"></canvas>
</body>
</html>
```

# Intro to HTML

(Hyper Text Markup Language)

```
<!DOCTYPE html>  
<html>  
<head>  
<title>Canvas Graphics</title>
```

HTML tags (start tag and end tag)



after webpage has loaded (text, images, etc), call this function

```
</head>  
<body onload="init()">  
    <canvas id="theCanvas" width="640" height="480"></canvas>  
</body>  
</html>
```

size of the window (you choose)





# Intro to HTML

(Hyper Text Markup Language)

```
<!DOCTYPE html>  
<html>  
<head>  
<title>Canvas Graphics</title>
```

HTML tags (start tag and end tag)

after webpage has loaded (text, images, etc), call this function

```
</head>  
<body onload="init()">  
  <canvas id="theCanvas" width="640" height="480"></canvas>  
</body>  
</html>
```

size of the window (you choose)

id: name the element, which we can refer to later

# Intro to HTML

(Hyper Text Markup Language)

```
<!DOCTYPE html>
<html>
<head>
<title>Canvas Graphics</title>
```

Indentation: not required,  
but good style

HTML tags (start tag and end tag)

after webpage has loaded (text, images, etc), call this function

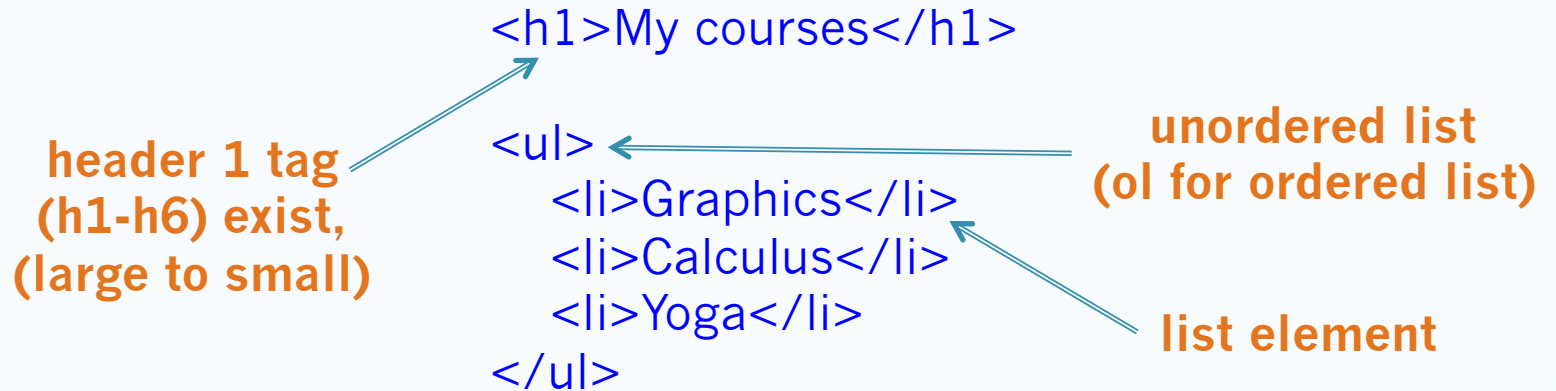
```
</head>
<body onload="init()">
  <canvas id="theCanvas" width="640" height="480"></canvas>
</body>
</html>
```

size of the window (you choose)

id: name the element, which we can refer to later

# HTML examples

## Lists



# HTML examples

## Lists

**header 1 tag (h1-h6) exist, (large to small)**

```
<h1>My courses</h1>
```

**unordered list (ol for ordered list)**

```
<ul>
```

- list element**
- Graphics
- Calculus
- Yoga

```
</ul>
```

## Tables

```
<table>
```

Graphics
2:40-4pm

```
</table>
```

**table row**

**table data**

# HTML examples

## Lists

**header 1 tag (h1-h6) exist, (large to small)**

```
<h1>My courses</h1>
```

**unordered list (ol for ordered list)**

```
<ul>
```

- list element**
- Graphics</li>
- Calculus</li>
- Yoga</li>

```
</ul>
```

## Tables

```
<table>
```

Graphics
2:40-4pm

```
</table>
```

**table row**

**table data**

**This type of code usually goes after the head.**

# Intro to JavaScript

# Intro to JavaScript

```
<!DOCTYPE html>  
<html>  
<head>  
<title>Canvas Graphics</title>
```

JavaScript goes here  
(usually within the head)

```
</head>  
<body onload="init()">  
    <canvas id="theCanvas" width="640" height="480"></canvas>  
</body>  
</html>
```

# Intro to JavaScript

```
<!DOCTYPE html>
<html>
<head>
<title>Canvas Graphics</title>
<script>
    var canvas;    // DOM object corresponding to the canvas
    var graphics;  // 2D graphics context for drawing on the canvas

    function draw() {
        // draw on the canvas, using the graphics context
        graphics.fillText("Hello World", 10, 20);
    }

    function init() {
        canvas = document.getElementById("theCanvas");
        graphics = canvas.getContext("2d");
        draw(); // draw something on the canvas
    }
</script>
</head>
<body onload="init()">
    <canvas id="theCanvas" width="640" height="480"></canvas>
</body>
</html>
```

JavaScript  
tags



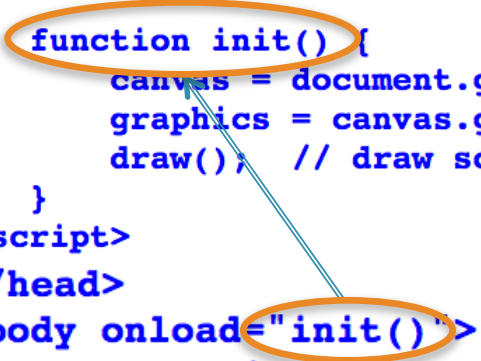


# Intro to JavaScript

```
<script>
    var canvas;    // DOM object corresponding to the canvas
    var graphics;  // 2D graphics context for drawing on the canvas

    function draw() {
        // draw on the canvas, using the graphics context
        graphics.fillText("Hello World", 10, 20);
    }

    function init() {
        canvas = document.getElementById("theCanvas");
        graphics = canvas.getContext("2d");
        draw(); // draw something on the canvas
    }
</script>
</head>
<body onload="init()">
    <canvas id="theCanvas" width="640" height="480"></canvas>
</body>
</html>
```

A diagram consisting of two orange ovals. The first oval is around the 'function init()' line in the JavaScript code block. The second oval is around the 'init()' part of the 'onload="init()"' attribute in the HTML body tag. A thin blue line connects the two ovals, pointing from the function definition to its invocation.

# Intro to JavaScript

Python:

def init():

JavaScript:

function init() {  
}

```
<script>
  var canvas;    // DOM object corresponding to the canvas
  var graphics; // 2D graphics context for drawing on the canvas

  function draw() {
    // draw on the canvas, using the graphics context
    graphics.fillText("Hello World", 10, 20);
  }

  function init() {
    canvas = document.getElementById("theCanvas");
    graphics = canvas.getContext("2d");
    draw(); // draw something on the canvas
  }
</script>
</head>
<body onload="init()">
  <canvas id="theCanvas" width="640" height="480"></canvas>
</body>
</html>
```

# Intro to JavaScript

```
<script>
    var canvas;    // DOM object corresponding to the canvas
    var graphics;  // 2D graphics context for drawing on the canvas

    function draw() {
        // draw on the canvas, using the graphics context
        graphics.fillText("Hello World", 10, 20);
    }

    function init() {
        canvas = document.getElementById("theCanvas");
        graphics = canvas.getContext("2d");
        draw(); // draw something on the canvas
    }
</script>
</head>
<body onload="init()">
    <canvas id="theCanvas" width="640" height="480"></canvas>
</body>
</html>
```

# Intro to JavaScript

**Declare the variable canvas**  
**Initialize canvas**  
**Refer to canvas attributes**  
**Example: canvas.width**

```
<script>
  var canvas;    // DOM object corresponding to the canvas
  var graphics;  // 2D graphics context for drawing on the canvas

  function draw() {
    // draw on the canvas, using the graphics context
    graphics.fillText("Hello World", 10, 20);
  }

  function init() {
    canvas = document.getElementById("theCanvas");
    graphics = canvas.getContext("2d");
    draw(); // draw something on the canvas
  }
</script>
</head>
<body onload="init()">
  <canvas id="theCanvas" width="640" height="480"></canvas>
</body>
</html>
```

# Intro to JavaScript

Python: `i = 1`

JavaScript: `var i = 1;`

```
<script>
  var canvas;    // DOM object corresponding to the canvas
  var graphics;  // 2D graphics context for drawing on the canvas

  function draw() {
    // draw on the canvas, using the graphics context
    graphics.fillText("Hello World", 10, 20);
  }

  function init() {
    canvas = document.getElementById("theCanvas");
    graphics = canvas.getContext("2d");
    draw(); // draw something on the canvas
  }
</script>
</head>
<body onload="init()">
  <canvas id="theCanvas" width="640" height="480"></canvas>
</body>
</html>
```

# Intro to JavaScript

Note: neither “var” nor “;” are strictly necessary, but good practice!

Python: `i = 1`

JavaScript: `var i = 1;`

```
<script>
  var canvas;    // DOM object corresponding to the canvas
  var graphics;  // 2D graphics context for drawing on the canvas

  function draw() {
    // draw on the canvas, using the graphics context
    graphics.fillText("Hello World", 10, 20);
  }

  function init() {
    canvas = document.getElementById("theCanvas");
    graphics = canvas.getContext("2d");
    draw(); // draw something on the canvas
  }
</script>
</head>
<body onload="init()">
  <canvas id="theCanvas" width="640" height="480"></canvas>
</body>
</html>
```

# Intro to JavaScript

```
<script>
  var canvas;    // DOM object corresponding to the canvas
  var graphics;  // 2D graphics context for drawing on the canvas

  function draw() {
    // draw on the canvas, using the graphics context
    graphics.fillText("Hello World", 10, 20);
  }

  function init() {
    canvas = document.getElementById("theCanvas");
    graphics = canvas.getContext("2d");
    draw(); // draw something on the canvas
  }
</script>
```

# Graphics functions

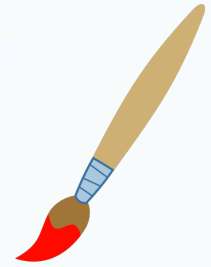
- 1) `graphics.fillStyle = "red";`
- 2) `graphics.fillRect(x, y, w, h);`



# Graphics functions

1) `graphics.fillStyle = "red";`

*pick up paintbrush*

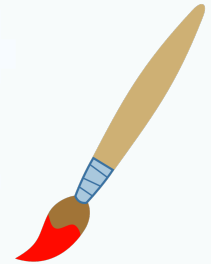


2) `graphics.fillRect(x, y, w, h);`

# Graphics functions

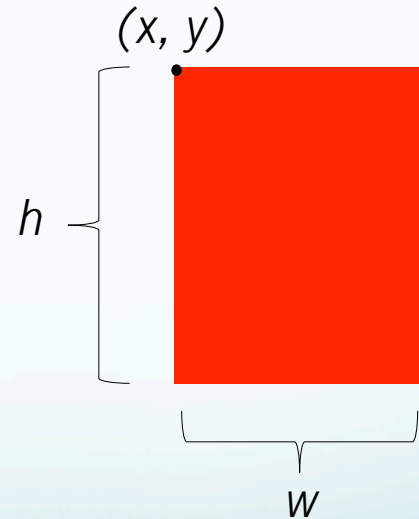
1) `graphics.fillStyle = "red";`

*pick up paintbrush*



2) `graphics.fillRect(x, y, w, h);`

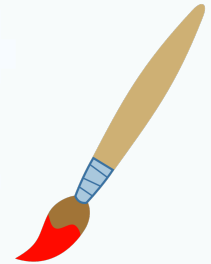
*paint*



# Graphics functions

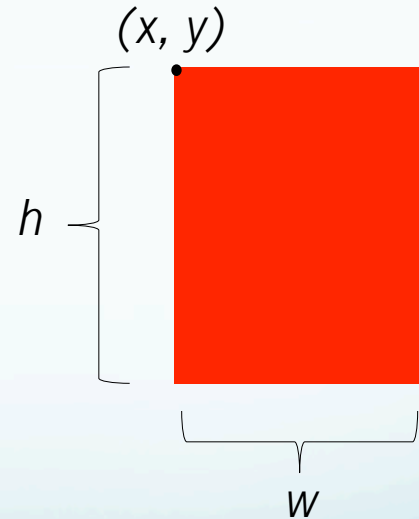
1) `graphics.fillStyle = "red";`

*pick up paintbrush*



2) `graphics.fillRect(x, y, w, h);`

*paint*



For now: `graphics.fillRect(x, y, 1, 1);`

*pixel coloring*

# Syntax

# For Loops

## Python:

```
for i in range(20, 40, 2):  
    print(i)
```

## JavaScript:

```
for (var i = 20; i < 40; i += 2) {  
    window.alert(i);  
}
```

# Conditionals

## Python:

```
if x > 0:
    print("positive")
elif x < 0:
    print("negative")
else:
    print("zero")
```

## JavaScript:

```
if (x > 0) {
    window.alert("positive");
} else if (x < 0) {
    window.alert("negative");
} else {
    window.alert("zero");
}
```

# Other Syntax

- Mod: %
- And: &&
- Or: ||
- Comment: //
- Always double check:
  - curly braces { }
  - indent based on { }
  - parenthesis ( )

# Debugging Demo



# Poll Everywhere + Lab 1

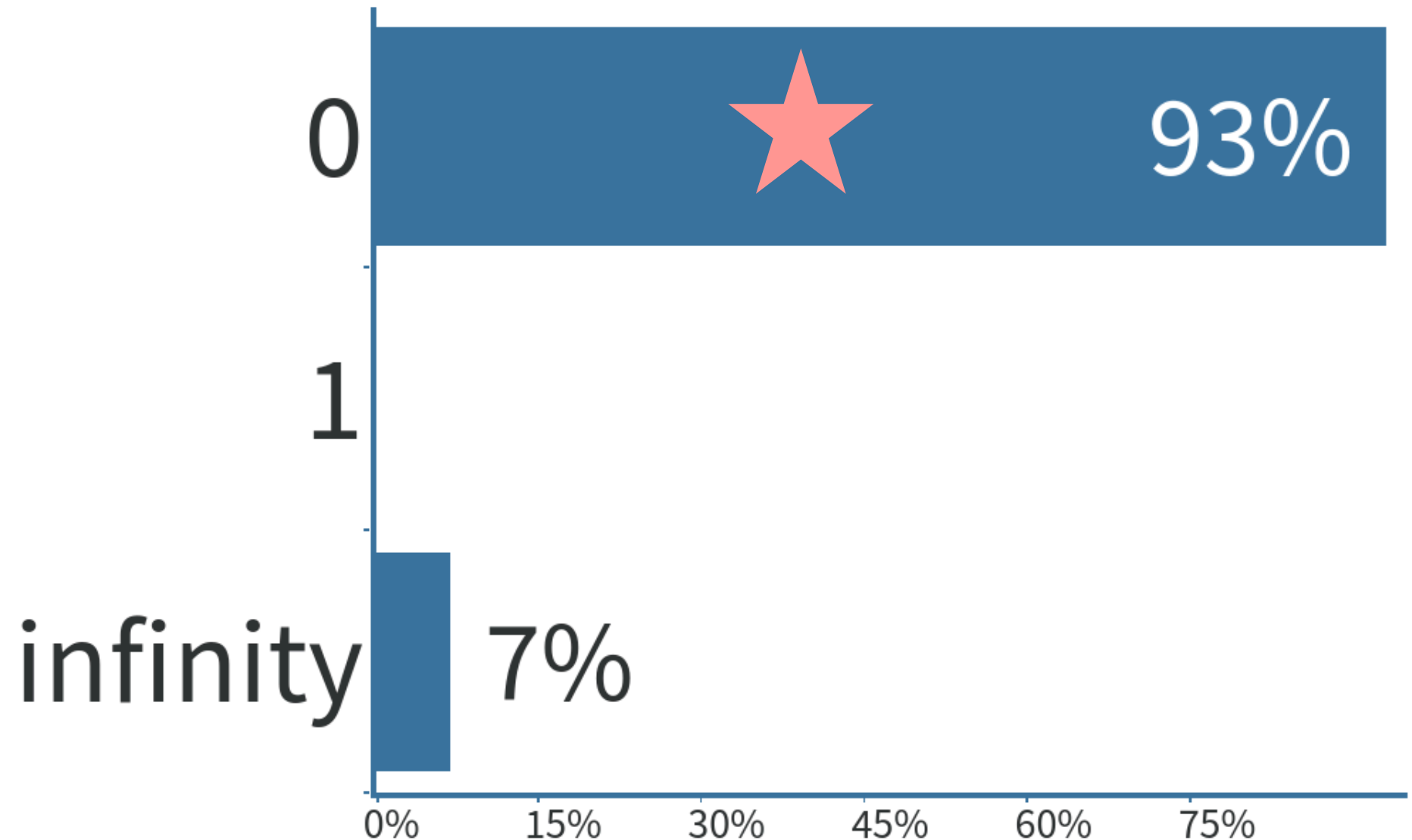
# What is the slope of a line with endpoints (50,30) and (70,30)?



When poll is active, respond at **PollEv.com/saramathieso692**



Text **SARAMATHIESO692** to **22333** once to join



# What is the slope of a line with endpoints (50,30) and (50,70)?



When poll is active, respond at **PollEv.com/saramathieso692**



Text **SARAMATHIESO692** to **22333** once to join

