Graphical output

# TODO

# Lesson Notes

## No output yet

* Reminder of CPU/Memory/Input/Output
* No output so far!
* All debugger to verify
* Learning to draw on a web page
* Still use debugger to verify
* How does a screen work, anyways?

## HTML Elements are objects

* Reminder of what elements and attributes are
* Every element on a web page is an object
* We can use these objects from our javascript
* Each attribute becomes a property (sometimes with a different name)
* The browser provides a global variable called “document” object
* The document object has functions that us look up the elements
* But there are many elements – we need a way of saying which one
* The browser has a special attribute, the “id” attributes
* We can find an element using the “id”

## The canvas element

* Canvas is a rectangle on a web page of a fixed width/height
* Like an image, only your functions do the drawing
* Two modes: 2d and 3d, we will use 2d
* Explain the 2d coordinate system
* Different than math (upside down)
* (Link to Mozilla)
* Zero based
* Showing the edges of the canvas (0,0,Width-1,Height-1)
* Units are pixels, what is a pixel
* Examples of monitor sizes

## Adding a canvas to your page

* Add canvas element
* Getelementbyid
* Getting the context
  + Two kinds (2d and 3d)

## Drawing a rectangle

* fillRect example

## Drawing text

* fillText example
* “hello world”, and the story behind it

## Understanding paths

* A "path" is a series of geometries that are grouped together
  + lines, arc, curves, rectangles, etc. can all be within one path
* We can “color in” the path using fill()
  + context.beginPath();
  + // path goes here
  + context.closePath();
  + context.fill();
* We can “outline” the path using stroke()
  + context.beginPath();
  + // path goes here
  + context.closePath();
  + context.stroke();
* Or both:
  + context.beginPath();
  + // path goes here
  + context.closePath();
  + context.stroke();
  + context.fill();

## Drawing a line

* Line is a geometry that can appear in a path:
  + moveTo – pick up the pen and move to a location on the canvas
  + lineTo – move to new location and draw line from old to new location
* So:
  + function drawLine(context, x1, y1, x2, y2) {
  + context.beginPath();
  + context.moveTo(x1, y1);
  + context.lineTo(x2, y2);
  + context.endPath();
  + context.stroke();
  + }

## Drawing arcs and circles

* Arc is a geometry that can appear in a path
  + Arc – draw part or all of a circle
* <http://www.html5canvastutorials.com/tutorials/html5-canvas-arcs/>
  + function drawCircle(context, centerX, centerY, radius) {
  + context.beginPath();
  + context.arc(centerX, centerY, radius, 0, Math.PI \* 2, false);
  + context.closePath();
  + context.stroke();
  + }

## Exercise

* Draw each type of shape within the canvas

## Adding colors and fonts

* Add more syntax to try

## Random numbers

* What is a random number
  + The idea of “pseudorandom” numbers
  + Random numbers are used in cryptography
* Math.random() will return a value from 0 to 1
  + An interval: [0, 1)
  + From math, bracket is inclusive, and parenthesis is exclusive
* Scaling random numbers:
  + Scale: [0, scale)
    - Math.random() \* scale
  + Shift and scale: [-scale/2, scale/2)
    - (Math.random() - 0.5) \* scale
  + Integer scale: [0, scale - 1]
    - Math.floor(Math.random() \* scale)
  + Integer scale: [1, scale]
    - Math.floor(Math.random() \* scale) + 1
* Random decisions:
  + Math.random() < 0.2 // ~20% chance

## Exercise

* Draw a rectangle, circle, line, text at random locations within the canvas
* Vary the colors randomly
* Vary the sizes randomly
* Press [F5] to see the different results

## Two types of graphics here

* The canvas “forgets” what you did (immediate mode)
* HTML elements “remember” the structure (retained mode)
* General choice in computer science:
  + Remember the result
  + Remember how to build the result
  + Advantages to both
  + Can mix the two

## Exercise

* Draw a bar for each grade of the school for number of students
* Show example
* Height of bar is in pixels
* Given y-axis value, canvas size, and school data
* Extra: draw text underneath the bar
* Extra: each bar a different color