CC Lab: Week 02

Functions, loops, and animating in the browser with Canvas

Instructor: Paweena Prachanronarong

Installing packages in Sublime Text 2

1. **control + ~** (opens the console in ST2)
2. **https://sublime.wbond.net/installation#st2** (paste code into console)
3. **Restart ST2** (finishes installation)
4. **⌘ + shift + p**
5. **Type “install package”**
6. **“SublimeLinter”** (reads source code and looks for common mistakes)
7. **Repeat steps 4 – 5, select “ColorPicker”** (⌘ + shift + p to open color picker)
8. **Repeat steps 4 – 5, select “Tag”** (collection of packages about tags)
8. **Repeat steps 4 – 5, select “JSFormat”** (JavaScript formatting plugin)

Functions in JavaScript:

Declared with `var`

```
<!DOCTYPE html>
<html>
    <head>
    </head>
    <body>
        <div id="myDiv"> here</div>

        <script type="text/javascript">
        var myFunction = function (){
            console.log("hit my function!!");
            var variable1 = 5;
            var variable2 = 8;
            return variable1 + variable2;
        }

        var result = myFunction ();
        console.log( result );
        </script>
    </body>
</html>
```

Functions in JavaScript:

Can accept **parameters**! (Just like Processing)

```
var addUp = function( arg1 , arg2 ){
    console.log("hit addUp!!");
    console.log("adding: "+ arg1 +" and "+ arg2);
    return arg1 + arg2;
}

var result = addUp(8, 19);
console.log( "result: "+ result );
```

HOMEWORK EXAMPLE

DOM: Document Object Model

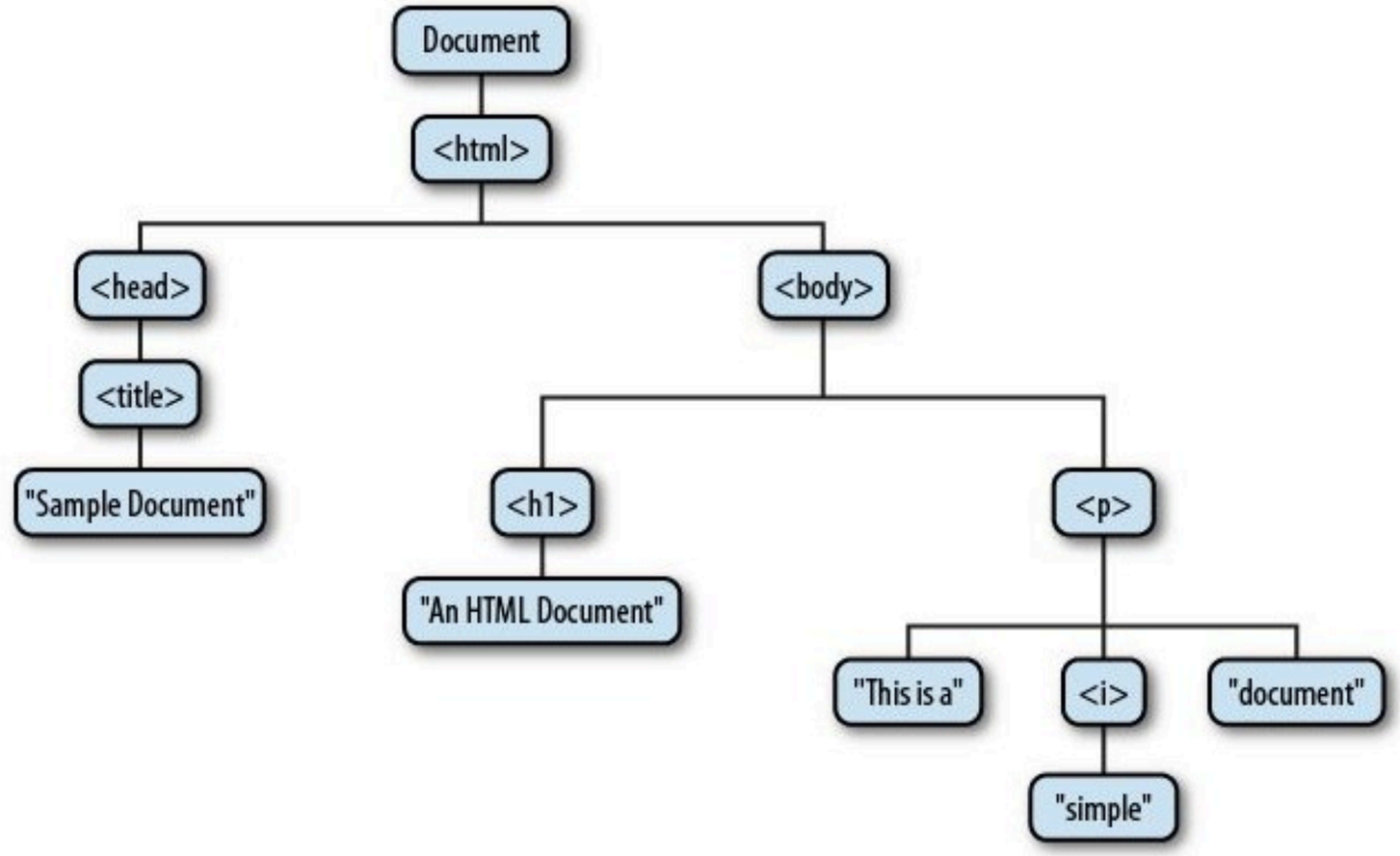
- Cross-platform, language-independent convention for **representing and interacting with objects in HTML**
- Browsers render HTML using a **DOM tree structure**, top-most being the "Document object"
- When a page is rendered, the **browser downloads the HTML** into local memory, inventories all elements in the DOM
- **The DOM is also how JavaScript transmits the status of the browser** in HTML pages

DOM: Document Object Model

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Sample Document</title>
5      </head>
6      <body>
7          <h1>An HTML Document</h1>
8          <p>This is a <i> simple </i> document.</p>
9      </body>
10 </html>
```

Flanagan, David (2011-04-18). JavaScript: The Definitive Guide (Definitive Guides) (p. 362). O'Reilly Media. Kindle Edition.

DOM: Document Object Model



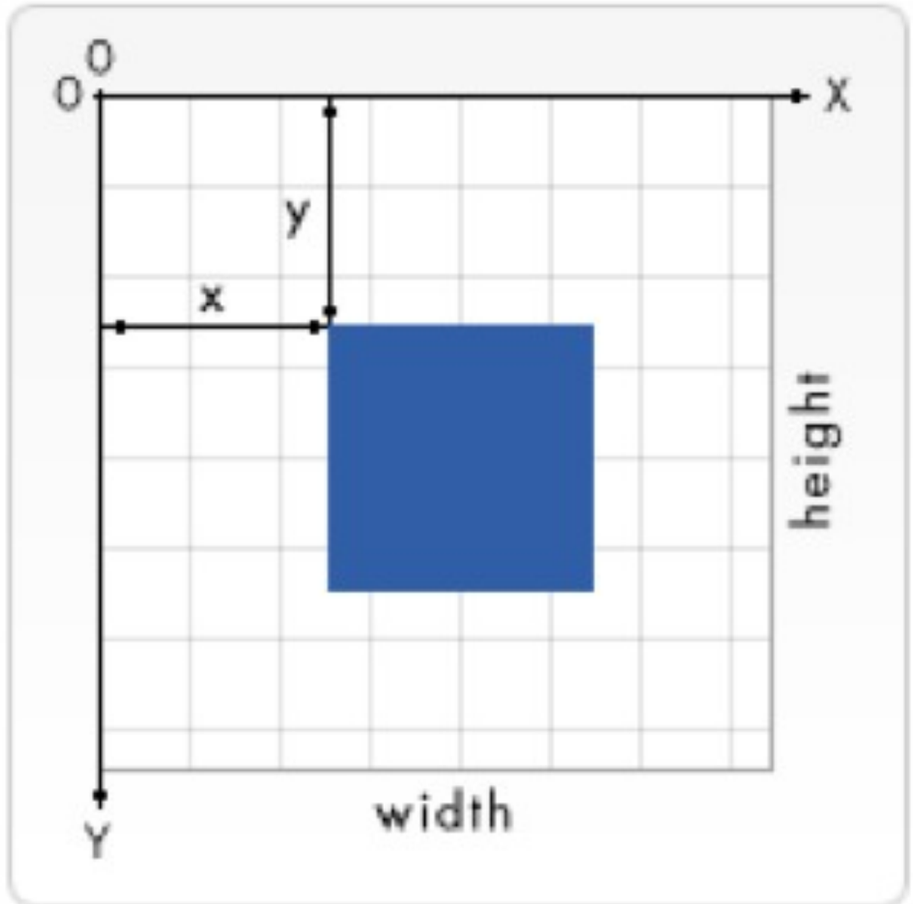
Flanagan, David (2011-04-18). JavaScript: The Definitive Guide (Definitive Guides) (p. 363). O'Reilly Media. Kindle Edition.

Canvas.js:

- `<canvas>` is an empty **HTML element**
- Used to draw **graphics** using JavaScript
- Can **draw** graphs, photo compositions, or simple and complex **animations**
- Supported by Safari, Firefox, Chrome and IE9+
- Standardized by **HTML5** but has been around longer than that

The grid:

- Just like in Processing, when drawing in the `<canvas>`, we refer to a **global coordinate space**



Barebones canvas example:

```
<html>
  <head> <!-- html up until here, notice the comment syntax -->
    <script type="application/javascript"> //-- we're now javascripting

      function draw() {
      //this animation function, to be called later on
        //locate the "myCanvas" element from the <body>
        var canvas = document.getElementById("awesomeCanvas");
        if (canvas.getContext) { // checks if this browser supports canvas!

          /*** meat and potaters start here ***/
          var ctx = canvas.getContext("2d");
          //ctx means context, needed to activate canvas
          //2d means works in 2d mode

          ctx.fillStyle="rgb(200,0,0)"; //set fill color (r,g,b)
          ctx.fillRect(10, 10, 300, 300); //draw a rectangle (x,y,w,h)
        }
      }
    </script>

  </head>
  <!-- onLoad means the browser will wait until all code is loaded
  before rendering *anything* -->
  <body onload="draw();">
    <!-- only two attributes for a <canvas> object are id, width, and height -->
    <canvas id="awesomeCanvas" width="700" height="700"></canvas>
  </body>
</html>
```

Drawing rectangles:

- Rectangles are the only primitive shape supported by canvas
- All other shapes are created by combining one or more paths

```
fillRect(x, y, width, height);   //Draws a filled rectangle.
strokeRect(x, y, width, height); //Draws a rectangular outline.
clearRect(x, y, width, height);  //Clears the specified rect area,
                                 //making it transparent.
```

Drawing color:

```
// all of these create an orange fill
ctx.fillStyle = "#FFA500";              // hex
ctx.fillStyle = "rgb(255,165,0)";       // (r, g, b)
ctx.fillStyle = "rgba(255,165,0,200)"; // (r, g, b, a)
```

```
// all of these create an orange stroke
ctx.strokeStyle = "#FFA500";              // hex
ctx.strokeStyle = "rgb(255,165,0)";       // (r, g, b)
ctx.strokeStyle = "rgba(255,165,0,200)"; // (r, g, b, a)
```

Look familiar?

JavaScript...

```
ctx.fillStyle = "rgba(0, 0, 200, 0.5)"; //color
ctx.fillRect(130, 130, 300, 300);       //(x,y,w,h)
```

Processing...

```
fill(100, 200, 50, 128);
rect(0, 0, 150, 50);
```

Fonts and text:

Again, super similar...

```
ctx.fillStyle="rgb(0,0,0)"; //black color
ctx.font="30px Arial";      //font attributes
ctx.fillText("Hello World",50,350); //("poetry", x, y)
```

...notice how the attributes are in fact **CSS**!

Homework :-)

1. Add a profile photo to the class blog.
2. Order Arduino kits!
3. Create a simple animation using the <canvas> object. Experiment with shapes other than rectangles such as paths, circles, semi-circles, etc.

   FOCUS ON CREATIVITY!!

   BONUS: Use a minimum of one mouse interaction (click to start, move to draw, drag to manipulate, etc.)
4. Post code and references to the blog (**can include people too!**).
5. Write about one thing you learned on your own on the blog.

Arduino Kits:

RECOMMENDED:

https://www.sparkfun.com/products/11930

http://www.adafruit.com/products/68

GET ONE OF THESE IF YOU PLAN ON DOING LOTS IN ARDUINO:

https://www.sparkfun.com/products/11576

http://www.adafruit.com/products/1078

References:

- http://www.html5canvastutorials.com/tutorials/html5-canvas-tutorials-introduction/

- http://www.w3schools.com/html/html5_canvas.asp

- http://www.html5canvastutorials.com/advanced/html5-canvas-advanced-tutorials-introduction/

- https://developer.mozilla.org/en-US/docs/Web/API/EventTarget.addEventListener