IAT 355 : Lab 01

Web Basics

Overview

- HTML
- CSS
- Javascript
- HTML & Graphics

HTML - the language for the content of a webpage

```
<!DOCTYPE html>
<html>
  <head>
    <title>A Web Page</title>
    <style> … put css rules here … </style>
    <script> … javascript code goes here … </script>
  </head>
  <body>
        … content of the html page …
  </body>
</html>
```

example

HTML (contd)

There are several HTML elements, each for a different purpose. For example

<a> can be used to create links

<p> for paragraphs

** ** for grouping other elements in a block

Each element can have multiple attributes, which have predefined function

<div **id**=“block1” **class**=“block”> …

// example: 01.html

CSS - for styling the content of a webpage

CSS consists of rules. Each rule contains a selector — for selecting elements on a webpage — and corresponding rules.

Here is an example that adds a margin of 100px to all div elements on the page

```
div {
    margin-left: 100px;
}
```

Here is another that adds a background to all elements with class ‘block’

```
.block{
    background: #cdcdcd;
}
// example: 02.html
```

Javascript - for run-time processing in a webpage

Javascript is a programming language that allows the programmer to manipulate the content of a webpage. Here is what it looks like

```
<script>

    console.log(“Hello World”); //similar to println(“Hello World”) in processing

</script>
```

Externalizing CSS and JS

You can store the css and js code in separate files than the html files. To do that, you write files with extensions **.css** and **.js** for CSS and JS code respectively. Use the **<head>** section to include external files.

To include css:

```
<link rel="stylesheet" href="04.css" type="text/css">
```

To include js:

```
<script type="text/javascript" src="04.js">
</script>
```

example: 04.html

Javascript: Crash Course

Variable Declaration and Type

var x = 100; //var is optional but can affect scope

x = “Ryan”; //this is NOT an error unlike Java/Processing

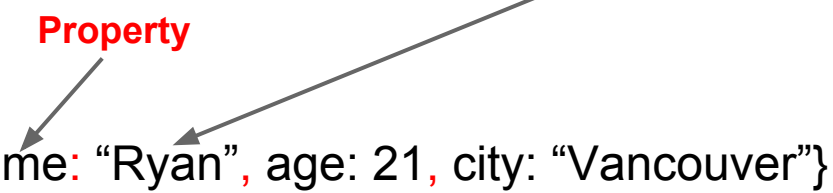
//in Processing you do **int x = 100;** OR **String x = “Hello”**;

Comments

Use ample of comments using

- // (single line comment)
- /* ... */ (multi-line comment)

Objects



var myObj = {name: “Ryan”, age: 21, city: “Vancouver”}

Concatenation vs Addition

- 5 + 5 //10
- 5 + “5” // “55”

Functions

```
function myFunction(arg1, arg2) {
    console.log(arg1);
    console.log(arg2);
}
myFunction() //undefined, undefined
myFunction(“a”) //a, undefined
myFunction(“a”,“b”) //a, b
```

Variable Scope & use of var

```
function f(){
        var myVar = 10; //myVar is local
        myVar = 100; //myVar is STILL local
}
```

```
function f(){
        myVar = 10; //myVar is global
}
```

Creating Objects using Prototypes

```
//Javascript
function Person(name, age){
        this.name = name;
        this.age = age;
}
var myPerson = new Person("Ryan", 21);
myPerson.name //Ryan
```

```
//Processing
class Person{
        private String name;
        private int age;
         public Person(String name, int age){
                this.name = name;
                this.age = age;
        }
}
```

More Javascript objects

Properties aka fields can be added to objects are runtime, unlike Processing.

var myObj = new Person(“Ryan”, 21); // {name: “Ryan”, age: 21}

myObj.location = “Vancouver”; // {name: “Ryan”, age: 21, location: “Vancouver”}

You can even add new functions in the object

myObj.printName = function(){console.log(this.name;)}

HTML & SVG

SVG & HTML

You can create SVG on a webpage, just like any other HTML element. To create a SVG, you use the svg tag.

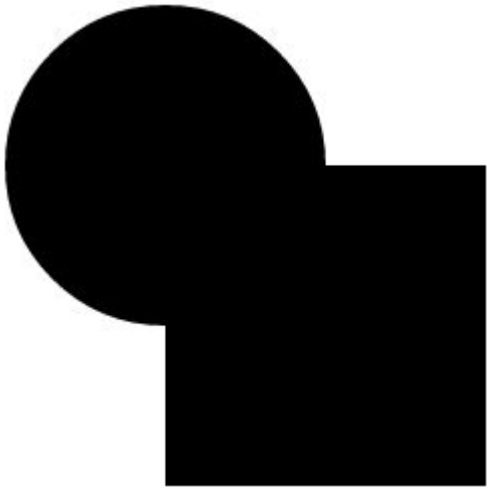
```
<svg>
    … content of the svg
</svg>
```

SVG Syntax

The <svg> tag:

```
<body>
<svg>
    <!-- Include SVG elements here -->
</svg>
<body>
```

An SVG Example



```
<svg width="500" height="500">
    <circle cx="100" cy="100" r="80" />
    <rect x="100" y="100" width="160" height="160" />
</svg>
```

SVG Elements

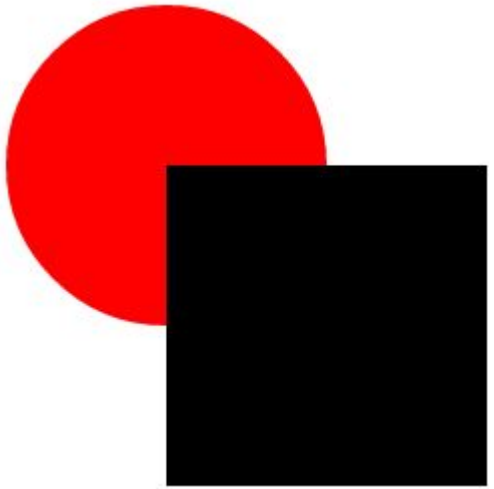
Some of the common graphics elements include:

- `<circle>`
- `<ellipse>`
- `<image>`
- `<line>`
- `<polygon>`
- `<polyline>`
- `<rect>`
- `<text>`

A More Complete Reference

https://developer.mozilla.org/en-US/docs/Web/SVG/Element

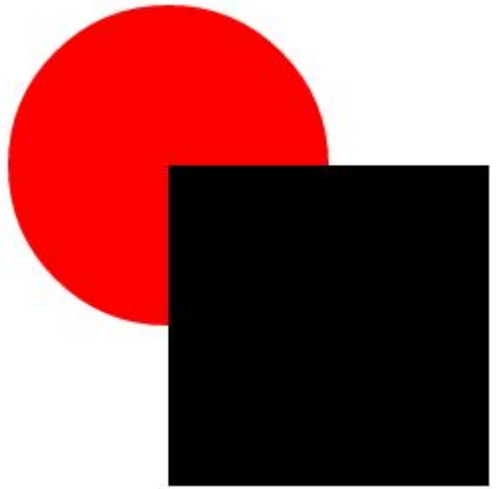
An SVG Example with some color



```
<svg width="500" height="500">
    <circle cx="100" cy="100" r="80" fill="red"/>
    <rect x="100" y="100" width="160" height="160" />
</svg>
```

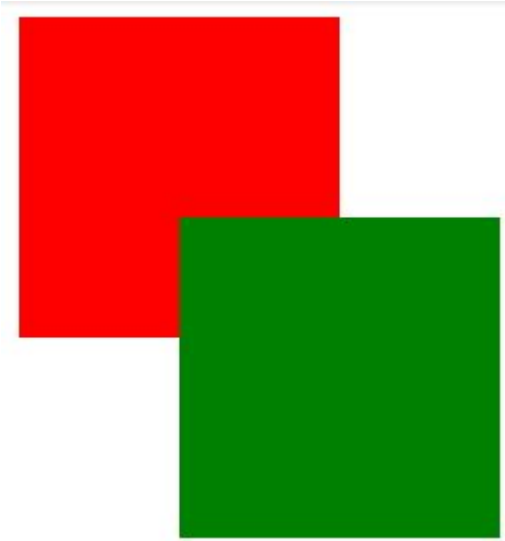
Styling with some CSS

```
//css
circle { fill: red;}
```



```
//html
<svg width="500" height="500">
    <circle cx="100" cy="100" r="80"/>
    <rect x="100" y="100" width="160" height="160" />
</svg>
```

You can event add ID & classes



```
//css
#mySquare{ fill: green }
.red { fill:red }

//html
<svg width="500" height="500">
    <rect x="20" y="20" width="160" height="160" class="red"/>
    <rect x="100" y="100" width="160" height="160" id="mySquare"/>
</svg>
```

example: 05.html

Creating SVG using d3.js

d3.js is a Javascript library — collection of helper classes and functions — to manipulate webpages, based on data. Hence the name, d3 (data-driven documents). You can even use it to manipulate the page without any data.

We can use d3 to create visualizations !

Using D3

1. Download the core library by clicking this link: https://github.com/d3/d3/releases/download/v4.4.1/d3.zip
2. Extract the zip file
3. Copy d3.js in the same directory as the html file.
4. Repeat steps 1-3 for any micro-libraries (for d3 v4.x only)
5. In the head of the html, include the file

```
<script type="text/javascript" src="d3.js">
</script>
```

Recreate the squares using d3

First create the svg container element

```
var svg = d3.select("body")
            .append("svg")
            .attr("width", 500)
            .attr("height", 500);
```

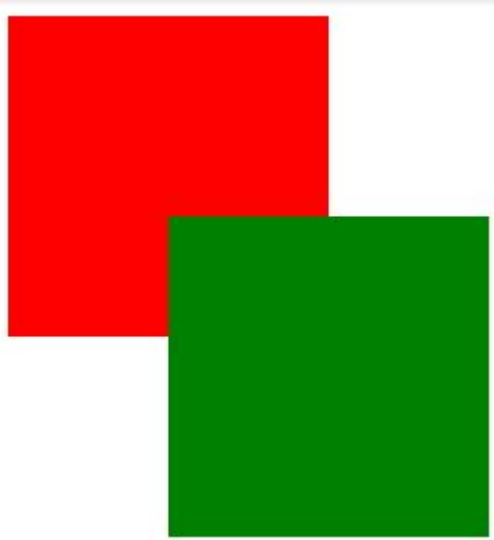
Adding the red rectangle

```
svg.append("rect").attrs({
    x: 20,
    y: 20,
    width: 160,
    height: 160,
    class: "red"
    })
```

Recreate the squares using d3 (contd.)

Add the green rectangle

```
svg.append("rect").attrs({
    x: 100,
    y: 100,
    width: 160,
    height: 160,
    id: "mySquare"
})
```



example: 06.html