HTML5 & Javascript Programming – Tutorial Sheet 2

Functions

In each case, write the requested function and then write a typical statement that calls the function. To test your functions and calls, add the code to the HTML listed below, load this into a browser and open a browser console (Ctrl+Shitft+J) to view the result (note the console.log() call at the end of the <script> tag).

1. Write a function that returns the circumference of a circle, given that the radius is passed into the function as a parameter. Repeat this for the area of a circle.
2. Write a function that will calculate and add VAT to a price. The function should take two parameters, these being the price and the VAT rate (as a percentage).
3. Write a function that converts degrees Farenheit into degrees Celcius (see last week's tutorial). Repeat for converting Celcius into Farenheit.

```
<!DOCTYPE html>
<html>
<head lang="en">
    <meta charset="UTF-8">
    <title></title>
</head>
<body>
<script>
    // An example function...
    function getYear() {
        var d = new Date();
        return d.getFullYear();
    }
    console.log(getYear());
</script>
</body>
</html>
```

4. Write a function that takes two parameters, these being a person's first and last names, and returns the person's full name (including the space between first and last).
5. Write a function that takes a person's full name (e.g. "Fred Bloggs") and returns the surname part. You will need to use two standard JS functions – `substr(4)` returns part of the string from the fifth character on, and `indexof(' ')` returns the location of the first space in a string.
6. Write a function that returns the person's first name from a full name, given that the function call `substr(0, 4)` will return the first four characters of a string.
7. Write a function that takes a persons full name and returns an object with two properties – first and second, these being the first name and surname.
8. Write a function that takes a number as a parameter and returns true if the number is the current year number. You can get a Date object containing today's date using ***`new Date()`***, and the Date type has a function `getFullYear()` that returns this number.
9. The Date type has functions to return the day, month and year of a date, but these are fairly inconsistent in how they work. Given a date object, `d`, `d.getDate()` returns the day number, `d.getMonth()` returns the month number **counting from 0**, and `d.getFullYear()` returns the year number in 4 digits. Write a new

function, `getDateParts()` that takes a date as a parameter and returns an object with properties day, month, year, which contain the normal parts of a given date (e.g. day=25, month=12 and year=2015 being Christmas day in this year).

10. The `Date` type also includes time information – e.g. `d.getHours()`, `d.getMinutes()` and `d.getSeconds()` return the appropriate information. Write a function that creates a time object, which has hours, minutes and seconds properties. Note: `t = new Date("01-01-01 12:30:00")` will return a time set to half past midday on January 1st 2001.

Methods (i.e .Functions attached to objects)

Javascript contains a number of built-in objects that can be used in application code (Date is one of these types). The Math object contains a number of common maths functions, such as square root. We could use that in a program to create a right-angled triangle object:

```
var tri = {
    x: 5,
    y: 12,
    hyp: function() {
        return Math.sqrt(this.x*this.x + this.y*this.y);
    }
}
```

11. Write code to create a rectangle object, which has width and height properties. Add two functions to the object – perimeter and area, to calculate the distance around and the area of a given rectangle. Your code should work as shown:

```
var rect = { // your object code in here }
console.log(rect.perimeter());
console.log(rect.area());
```

12. Repeat the above exercise, but define a Circle object – it should have a radius property and `circumference()` and `area()` methods.

13. Create an object which has a `fullName` proprerty, and `firstName()` and `lastName()` methods .

14. Create an Arithmetic object which has properties `number1` & `number2`, and methods `sum()`, `difference()`, `product()` and `quotient()` (i.e. division).

Constructors and Prototypes

15. Questions 11-15 were about creating Objects in code. A more sustainable way of creating objects is to define a Constructor function, and add methods to the constructor's prototype object (see the notes and slides from tis weeks' lecture. Repeat questions 11-15 to create new ***types*** of object, with each having a constructor function and one or more prototype methods. Write code to test your object type definitions.