Unit 6 - JavaScript Report

Task 1 - Discribing Application and Limitation

JavaScript is a programming language that can be compiled by moder browsers. This makes JavaScript popular for coding webpages. Because each browser compiles JavaScipt differently, there are often problems with cross browser comapatability. This means that some features that are available in one browser, do not work in another.

Task 2 - Explaining a Sequence Statement

A sequence statement is anything that happens in a sequence. Mostly executed inside other statements.

var variable = 10; document.getElementById.innerHTML = variable; var x = variable + 10;

As you can see, These statements are all on the same line, and are simple. They only execute once.

Task 3 - Explaining a Select Statement

A select statement is a statement that selects a value for comparison. Examples of select statements include: if, else if, else, and while.

Here is an example of some code that checks is a number is validy by the system's terms:

```
function isNumber(n) {
    return !isNaN(parseFloat(n)) && isFinite(n);
}
function validateNum(num){ //This defines a function for validating a number.
    if (num == "") {// Checks that the number has been entered (1)
        return false;
    }
    else if (!isNumber(num)) { // Checks if the number is 6 characters long( 2)
        return false;
    }
    else if (!num >= 0){
        return true;
    }
    else { // ( 3)
        return  true;// if everything checks out, the function returns true
    }
}
function myFunction() {// this function is called by a button in HTML
    var message;
    var num = prompt("Please enter a number","");// Asks the user for input
    if (validateNum(num)){
        message = "Hello. The number "+ num +"is a valid number." ;
    }
    else{
        message = "Please click again and enter a valid number";
    }
    $("#greeting").html(message);// Publishes the message to an HTML element
}
```

If you look find the line with '(1)' at the end, you will see the first if statement. `if (num == "") {/ Checks that the number has been entered (1) return false; }` The stuff inside the brackets (num == ""), is a logical test. If you're familliar with maths, it's essentially asking if what is inside the brackets makes mathematical sense. If not, it's asking if "" is in the variable 'name'.

If this test is valid (ie. num is in fact equal to "" (Nothing)) then the second line is executed, forcing the function to stop and returning false.

If the statement does not make sense, then the second line is ignored.

Another example of a Select statement is 'if else'. Look at the statement at (2):

```
else if (name.length <= 1) { // Checks if the name is longer that one character (2)
    return false;
}
```

This statement is exactly the same as an if statement, with the exception that it will only execute if the if statement in the line above fails the test.

Task 4 - Explaining a Case Statement

A case statement is used to execute code if various satements are true. It is just short hand of an if else statement.

```
var hello = 0;
switch(hello) {
    case == 0 {
        alert(hello);
    }
    case != 0 {
        alert('WRONG');
    }
}
```

This case statement will check if hello is 0, and alert them with 0. If not, it will alert them with 'WRONG'.

Task 5 - Explaining an Itteration Statement

A iteration statement is a statement that cycles through data or executes mutiple times. The best example of a itteration statement is a for loop through an array.

```
var names = ['Bob','Bill','John','James'];// defining a variable
for (var i = 0; i < names.length; i++) {// Creating to loop where i is any number                //between 0
and the length of names
    alert('Hello ' + names[i]);// This will alert the user with every name
}
```

As you can see, the for loop cyles through items in the array 'names', and alerts for each name.

Task 6 - Explaining Arrays

An array is a type of variable that contains multiple values. These are useful to iterate through. An example of an array is given above. Arrays can contain multiple data types, including other arrays. There are also different types of arrays. There are multidimensional, and assosiative arrays.

Multidimentional arrays are arrays that contain other arrays. These are useful for containing essentailly records like a database, but are flexible to do almost anything.

Assosiative arrays are arrays that are index by both numbers and strings. This is useful for arrays that contains a largely variant amout of data, as the data can be acacessed easily.