<!------ script to generate question --------->

<!------ it must include at least one 1 as the example uses 0 AND 0--------->

<script>

var bit1 = Math.floor(Math.random()\*2);

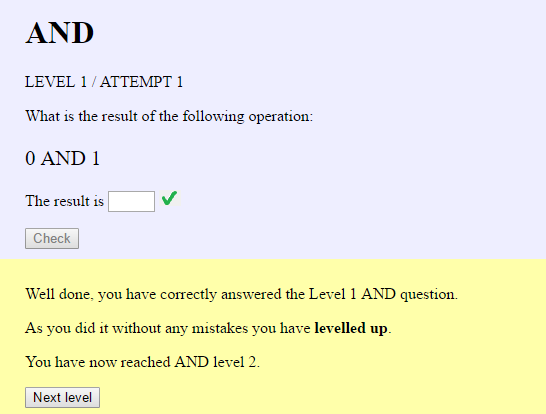
var bit2 = (bit1 == 0) ? 1 : Math.floor(Math.random()\*2);

document.getElementById("bit1").innerHTML = bit1;

document.getElementById("bit2").innerHTML = bit2;

window.answer = bit1 & bit2;

</script>



The problem is that Javascript does implicit type conversion, so when it compares an empty string (the empty text box) with the integer zero it converts the empty string to an integer, and that gives zero. So the values match.

The solution was to use the toString() method.

<script>

var bit1 = Math.floor(Math.random()\*2);

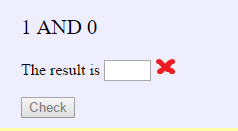
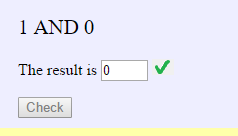
var bit2 = (bit1 == 0) ? 1 : Math.floor(Math.random()\*2);

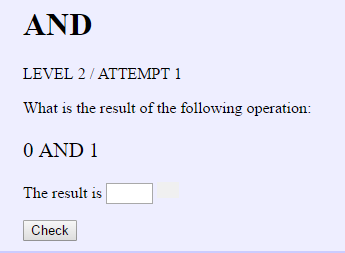
document.getElementById("bit1").innerHTML = bit1;

document.getElementById("bit2").innerHTML = bit2;

window.answer = (bit1 & bit2).toString();

</script>



This should show 00 and 01

<script>

var bits1 = Math.floor(Math.random()\*4);

var bits2 = Math.floor(Math.random()\*4);

document.getElementById("bits1").innerHTML = bits1.toString(2);

document.getElementById("bits2").innerHTML = bits2.toString(2);

var answer = (bit1 & bit2).toString(2);

</script>

Clearly the conversion strips leading zeros, so I need to add them back in.

<script>

var bits1 = Math.floor(Math.random()\*4);

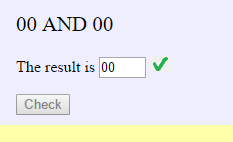
var bits2 = Math.floor(Math.random()\*4);

document.getElementById("bits1").innerHTML = (bits1 < 2) ? "0" + bits1 : bits1.toString(2);

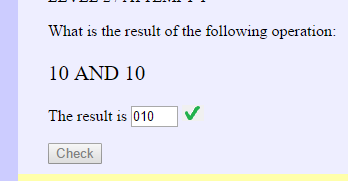
document.getElementById("bits2").innerHTML = (bits2 < 2) ? "0" + bits2 : bits2.toString(2);

var answer = "0" + (bits1 & bits2).toString(2).slice(-2);

</script>



The display is sorted, but the answer check is not working properly:



<script>

var bits1 = Math.floor(Math.random()\*4);

var bits2 = Math.floor(Math.random()\*4);

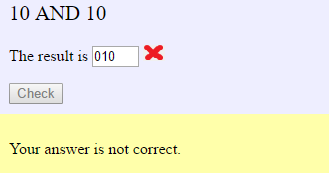
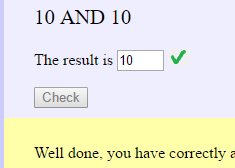
document.getElementById("bits1").innerHTML = (bits1 < 2) ? "0" + bits1 : bits1.toString(2);

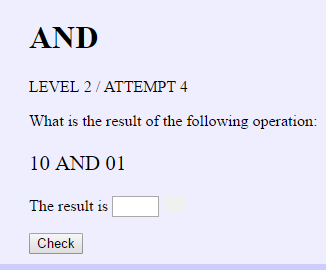
document.getElementById("bits2").innerHTML = (bits2 < 2) ? "0" + bits2 : bits2.toString(2);

var answer = ("0" + (bits1 & bits2).toString(2)).slice(-2);

</script>

I realised that I had forgotten to put brackets around the answer string operations, so the ‘slice’ happened before the ‘0’ was prepended.



When the player has had all three goes they can try a new question. This should be ‘Attempt 1’ again, but it displays as ‘Attempt 4’

function retryClicked() {

document.getElementById("studentAnswer").value = "";

document.getElementById("markSymbol").src = "images/blank.png";

document.getElementById("submitAnswer").disabled = false;

document.getElementById("wrong" + errorCount).style.display = "none";

document.getElementById("attemptNo").innerHTML = errorCount + 1;

}

function newClicked() {

retryClicked();

errorCount=0;

}

The problem is that the errorCount is reset *after* the display is changed. However we cannot reset it before the retryClicked() function is performed as this function uses the errorCount to reference the current response that is being shown, in order to hide it.

To solve the problem I have simply added an extra line which resets the display correctly.

function newClicked() {

errorCount=0;

retryClicked();

document.getElementById("attemptNo").innerHTML = errorCount + 1;

}