

Comp I25 - Visual Information Processing

Spring Semester 2018 - week 2 - friday

Dr Nick Hayward

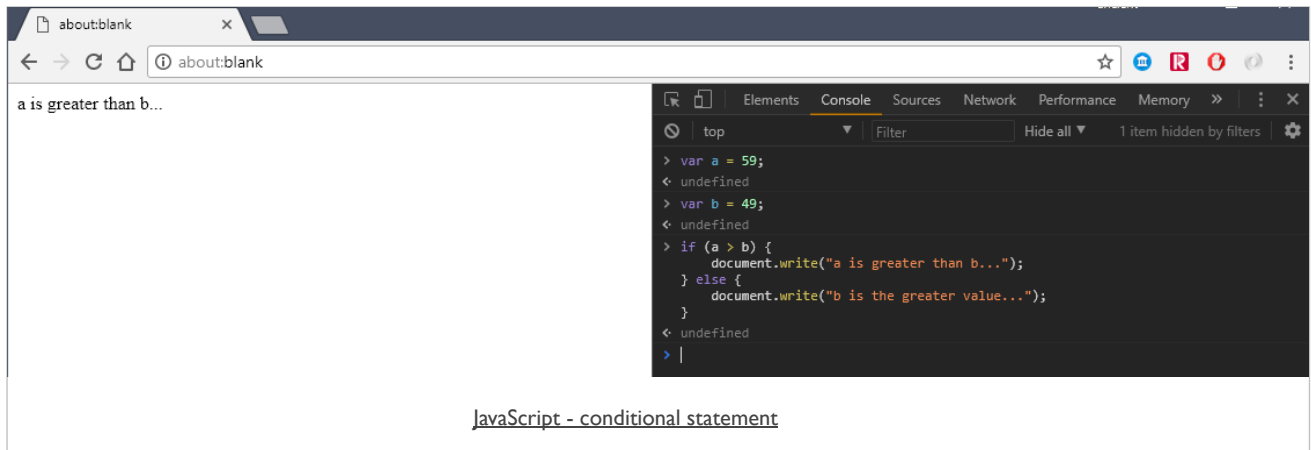
JS Basics - logic - blocks & conditionals - part 2

- additional option if this expression returns false
 - using an **else** clause

```
if (a > b) {  
  console.log("a is greater than b...");  
} else {  
  console.log("no, b is greater...");  
}
```

- for an `if` statement, JS expects a boolean
- JS defines a list of values that it considers *false*
 - e.g. `0`...
- any value **not** on this list of *false* values will be considered true
 - coerced to true when defined as a boolean
- conditionals in JS also exist in another form
 - the *switch* statement
 - more to come...

JS Basics - logic - conditional statement



The screenshot shows a web browser window with a single tab titled 'about:blank'. The address bar also shows 'about:blank'. The main content area of the browser displays the text 'a is greater than b...'. The browser's developer tools are open, showing the 'Console' tab. The console contains the following JavaScript code and its output:

```
> var a = 59;  
← undefined  
> var b = 49;  
← undefined  
> if (a > b) {  
    document.write("a is greater than b...");  
} else {  
    document.write("b is the greater value...");  
}  
← undefined  
> |
```

Below the browser window, the text JavaScript - conditional statement is displayed.

JS Basics - logic - loops

- loops allow repetition of sets of actions until a condition fails
- repetition continues whilst the requested condition holds
- loops take many different forms and follow this basic behaviour
- a loop includes the *test condition* as well as a *block*
 - *normally within curly braces*
 - *block executes, an iteration of the loop has occurred*
 - *four kinds of loop by default in JS,*
 - `for`
 - `for/in`
 - `while`
 - `do/while`

JS Basics - logic - loops - for

- for loop has three clauses, including
 - *initialisation clause*
 - *conditional test clause*
 - *update clause*

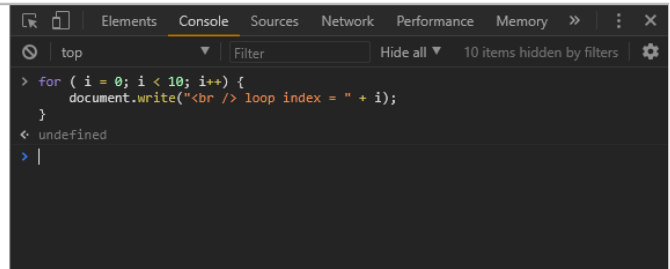
```
for (statement1; statement2; statement3) {  
  ...code block...  
}
```

- statement1 = executes before loop starts
 - statement2 = condition for running the loop
 - statement3 = executes after each iteration of the loop

JS Basics - logic - loops - for

Loop through a defined index from 0...

```
loop index = 0  
loop index = 1  
loop index = 2  
loop index = 3  
loop index = 4  
loop index = 5  
loop index = 6  
loop index = 7  
loop index = 8  
loop index = 9
```



The screenshot shows a web browser's developer console with the 'Console' tab selected. The console displays the output of a JavaScript for loop: 'loop index = 0' through 'loop index = 9', each on a new line. The code in the console is:

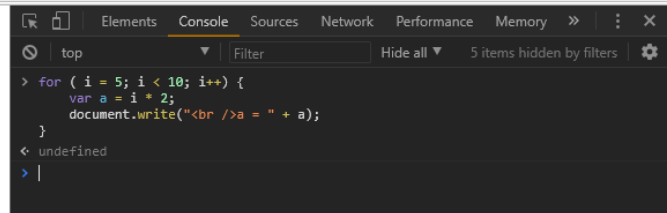
```
> for ( i = 0; i < 10; i++) {  
    document.write("<br /> loop index = " + i);  
}  
← undefined  
> |
```

JavaScript - for loop

JS Basics - logic - loops - for

Create a custom index and multiply per loop iteration...

```
a = 10  
a = 12  
a = 14  
a = 16  
a = 18
```



```
Elements Console Sources Network Performance Memory »  
top Filter Hide all 5 items hidden by filters  
> for ( i = 5; i < 10; i++) {  
    var a = i * 2;  
    document.write("<br />a = " + a);  
}  
← undefined  
> |
```

[JavaScript - for loop with multiplication](#)

JS Basics - logic - loops - while & do/while

- while and do...while loops
- basic difference between these loops, while and do...while
 - *conditional tested is before the first iteration (while loop)*
 - *after the first iteration (do...while) loop*
- if the condition is initially false
 - *a while loop will never run*
 - *a do...while will run through for the first time*
 - *other specialised forms of loop in JavaScript*
 - e.g. for/in...

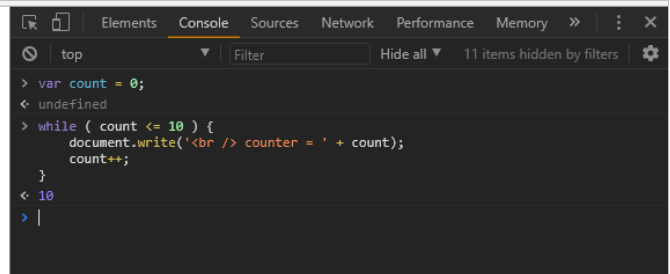
n.b. programming languages, and CS in general, start counting at 0. i.e. an index of values...

JS Basics - logic - loops - while

while loop continues to execute whilst condition remains true...

```
while (condition is true) {  
  ...code block...  
}
```

```
counter = 0  
counter = 1  
counter = 2  
counter = 3  
counter = 4  
counter = 5  
counter = 6  
counter = 7  
counter = 8  
counter = 9  
counter = 10
```

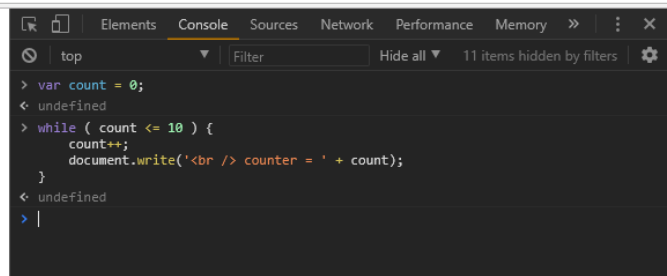


JavaScript - while loop

JS Basics - logic - loops - while

while loop with counter increment before output...

```
counter = 1  
counter = 2  
counter = 3  
counter = 4  
counter = 5  
counter = 6  
counter = 7  
counter = 8  
counter = 9  
counter = 10  
counter = 11
```



The screenshot shows a web browser's developer console with the 'Console' tab selected. The code being executed is as follows:

```
> var count = 0;  
< undefined  
> while ( count <= 10 ) {  
  count++;  
  document.write('<br /> counter = ' + count);  
}  
< undefined  
> |
```

The console output shows the counter incrementing from 1 to 11, with each value on a new line. The text '11 items hidden by filters' is visible at the top right of the console panel.

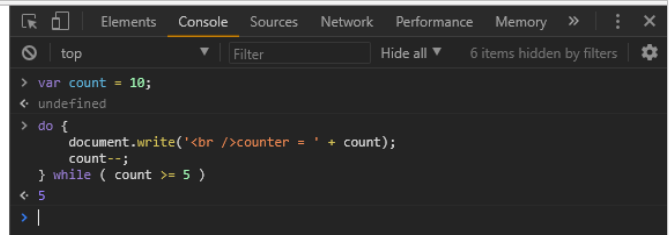
JavaScript - while loop

JS Basics - logic - loops - do/while

do/while loop executes do first, and then checks while condition...

```
do {  
    ...code block...  
} while (condition is true)
```

```
counter = 10  
counter = 9  
counter = 8  
counter = 7  
counter = 6  
counter = 5
```



The screenshot shows a web browser's developer console with the 'Console' tab selected. The console displays the following JavaScript code and its output:

```
> var count = 10;  
← undefined  
> do {  
    document.write('<br />counter = ' + count);  
    count--;  
} while ( count >= 5 )  
← 5  
> |
```

The output in the console shows the counter decreasing from 10 to 5, with each iteration printing the current value of the counter followed by a line break.

JavaScript - do/while loop