

Comp 125 - Visual Information Processing - Notes

Spring Semester - 22nd January 2018

Dr Nick Hayward

JS Basics - operators

- operators allow us to perform
 - *mathematical calculations*
 - *assign one thing to another*
 - *compare and contrast*
 - ...
- we can perform multiplication
 - `*`
 - e.g.

```
2 * 4
```

- we can also add, subtract, and divide numbers
 - `+`, `-`, `/`
 - e.g.

```
2 + 4
```

```
4 - 2
```

```
4 / 2
```

JS Basics - some common operators - part I

Assign values to variables, perform mathematical calculations, and compare values...

Assignment

- =
- e.g.

```
a = 4;  
b = 4 + 6;
```

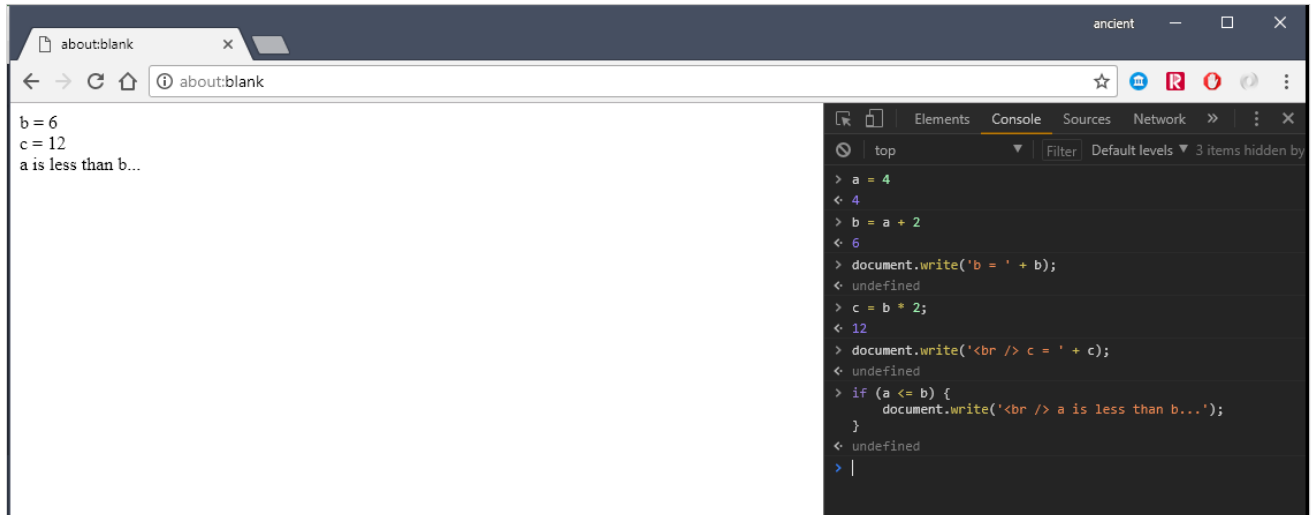
Comparison

- <, > <=, >=
- e.g.

```
a <= b
```

JS Basics - example operator usage

Assign values to variables, perform mathematical calculations, and compare values...



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 following text:

```
b = 6  
c = 12  
a is less than b...
```

On the right side of the browser window, the 'Console' tab is open, showing the following JavaScript code and its output:

```
> a = 4  
< 4  
> b = a + 2  
< 6  
> document.write('b = ' + b);  
< undefined  
> c = b * 2;  
< 12  
> document.write('<br /> c = ' + c);  
< undefined  
> if (a <= b) {  
    document.write('<br /> a is less than b...');  
}  
< undefined  
> |
```

JavaScript - assignment and comparison

JS Basics - some common operators - part 2

Compound assignment

- `+=`, `-=`, `*=`, `/=`
- compound operators are used to combine a mathematical operation with assignment
- same as `result = result + expression`
 - e.g.

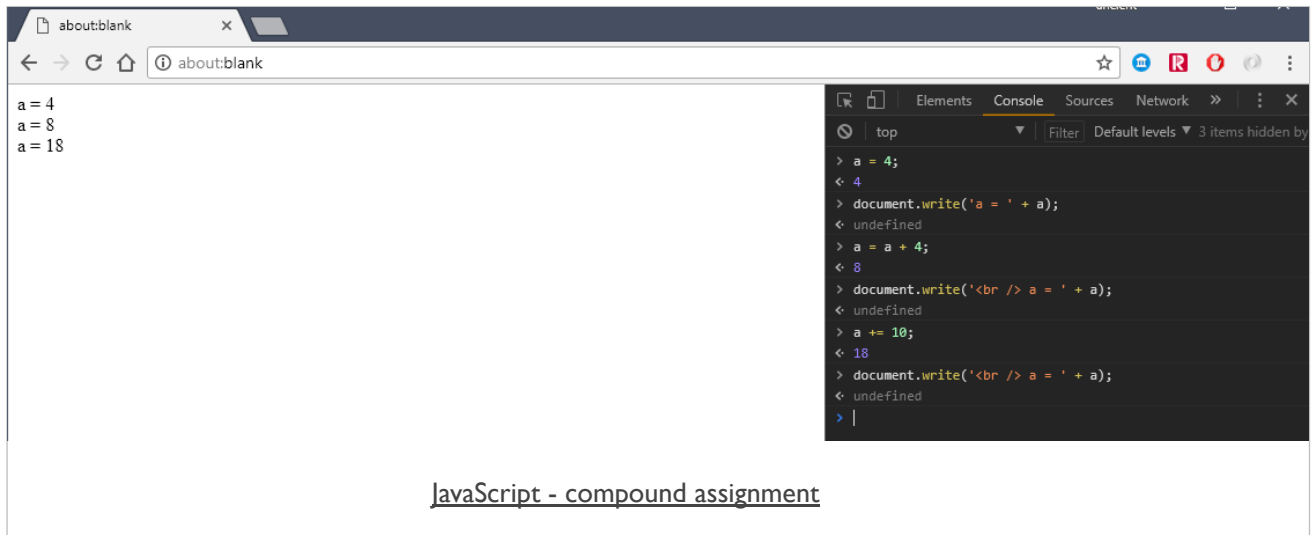
```
a = 4;  
a += 4;
```

is the same as

```
a = 4;  
a = a + 4;
```

JS Basics - example operator usage

Combine a mathematical operation with assignment of value to variable.



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 following text:

```
a = 4  
a = 8  
a = 18
```

On the right side of the browser window, the developer console is open, showing the following code and its output:

```
> a = 4;  
< 4  
> document.write('a = ' + a);  
< undefined  
> a = a + 4;  
< 8  
> document.write('<br /> a = ' + a);  
< undefined  
> a += 10;  
< 18  
> document.write('<br /> a = ' + a);  
< undefined  
> |
```

Below the browser window, the text JavaScript - compound assignment is displayed.

JS Basics - some common operators - part 3

Equality

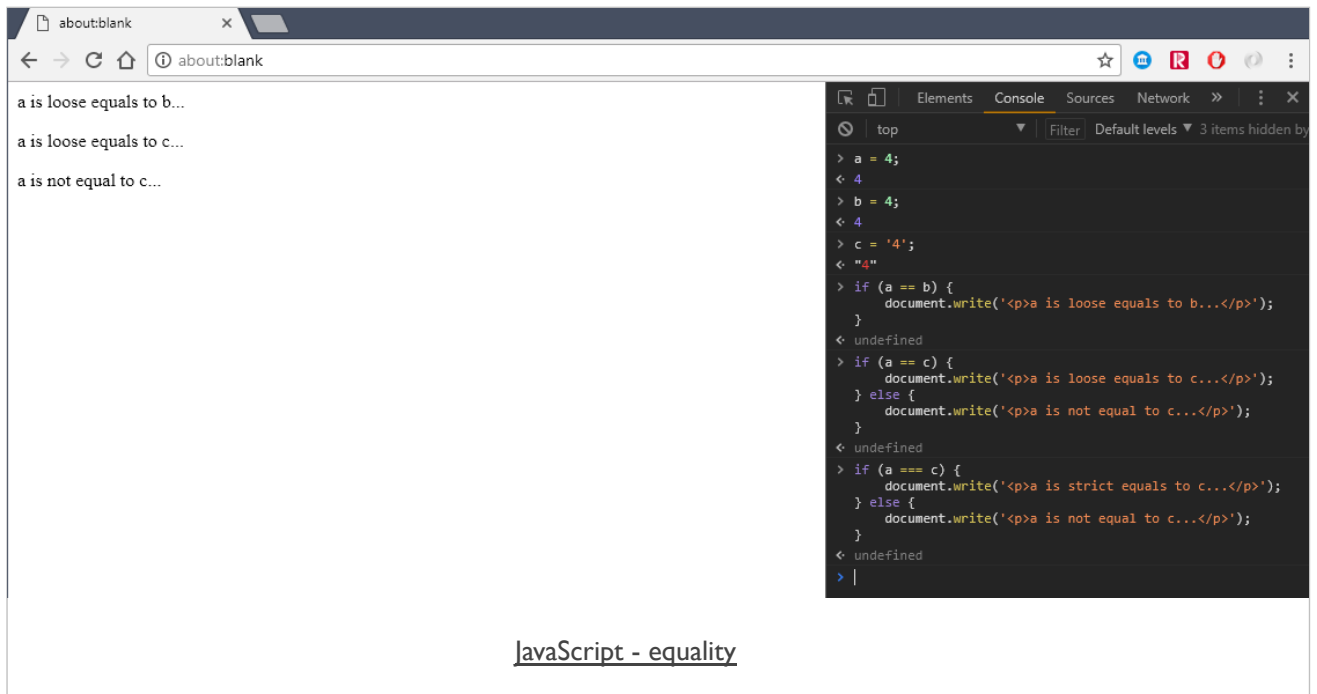
operator	description
==	loose equals
===	strict equals
!=	loose not equals
!==	strict not equals

■ e.g.

```
a == b  
a === b
```

JS Basics - example operator usage

Equality options for value, and value and type.



The screenshot shows a web browser window with an 'about:blank' page. The page content displays three lines of text: 'a is loose equals to b...', 'a is loose equals to c...', and 'a is not equal to c...'. The browser's developer console is open, showing the following JavaScript code and its execution results:

```
> a = 4;  
< 4  
> b = 4;  
< 4  
> c = '4';  
< "4"  
> if (a == b) {  
  document.write('<p>a is loose equals to b...</p>');  
}  
< undefined  
> if (a == c) {  
  document.write('<p>a is loose equals to c...</p>');  
} else {  
  document.write('<p>a is not equal to c...</p>');  
}  
< undefined  
> if (a === c) {  
  document.write('<p>a is strict equals to c...</p>');  
} else {  
  document.write('<p>a is not equal to c...</p>');  
}  
< undefined  
> |
```

Below the browser window, the text JavaScript - equality is displayed.

JS Basics - some common operators - part 4

Logical

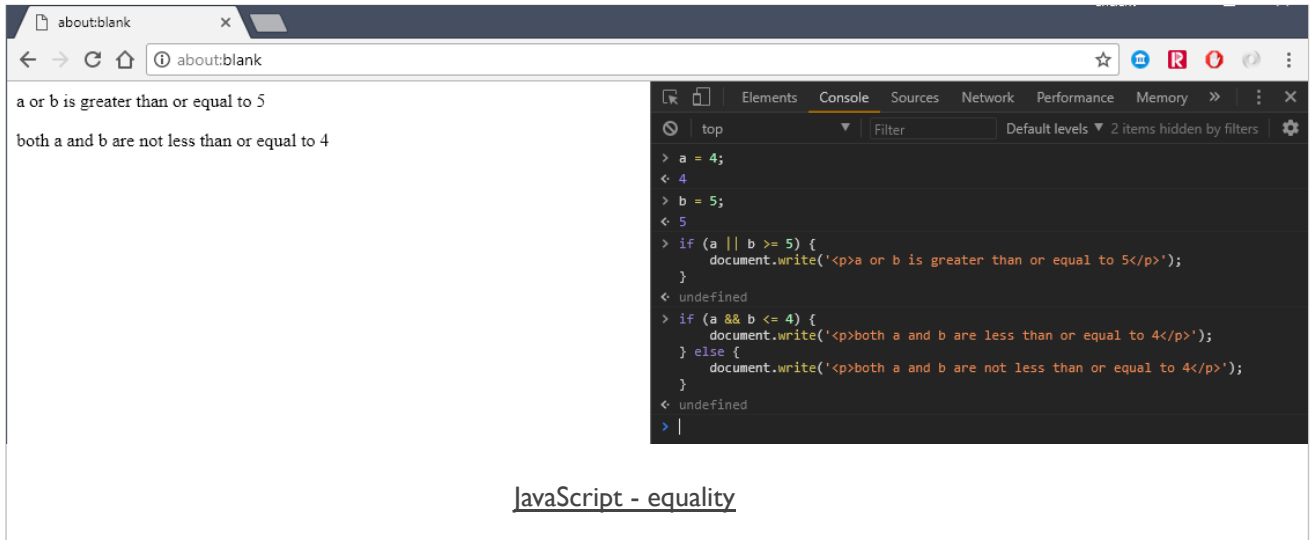
- used to express compound conditionals - **AND, OR**
 - `&&`, `||`
 - e.g. ***a*** or ***b***

```
a || b
```

JS Basics - example operator usage

Check compound conditionals, e.g.

- a **AND** b
- a **OR** b



The screenshot shows a web browser window with a single tab titled 'about:blank'. The address bar shows 'about:blank'. The main content area displays two lines of text: 'a or b is greater than or equal to 5' and 'both a and b are not less than or equal to 4'. The browser's developer tools are open, showing the 'Console' tab. The console contains the following code and output:

```
> a = 4;  
< 4  
> b = 5;  
< 5  
> if (a || b >= 5) {  
    document.write('<p>a or b is greater than or equal to 5</p>');  
}  
< undefined  
> if (a && b <= 4) {  
    document.write('<p>both a and b are less than or equal to 4</p>');  
} else {  
    document.write('<p>both a and b are not less than or equal to 4</p>');  
}  
< undefined  
> |
```

Below the browser window, the text JavaScript - equality is displayed.

JS Basics - values and types

- able to express different representations of values
 - *often based upon need or intention*
 - known as **types**
- JS has built-in types
 - allow us to represent **primitive** values
 - e.g. **numbers, strings, booleans**
 - **n.b.** boolean is either **TRUE** or **FALSE**
- such values in the source code are simply known as **literals**
- **literals** can be represented as follows,
 - *string literals use double or single quotes*
 - e.g. *"some text" or 'some more text'*
 - *numbers and booleans are represented without being escaped*
 - e.g. *49, TRUE*
- also consider arrays, objects, functions...