JavaScript cheat sheet

Trivial expressions

`blah`

A plain word refers to a variable in the current environment.

`"blah"`

A quoted word is a string, a value containing piece of text.

`12`

A number value.

`-12 1.5 3.4e10`

Other ways to write numbers (negative, fractional, with exponent).

`true`

Boolean (yes/no) value. `true` for yes, `false` for no.

Operator expressions

`a + b`

Binary operator applied to two values. `+` to add, `-` to subtract, `*` to multiply, `/` to divide.

`(a + b) * c`

Parenthesis for explicit grouping.

`a < b`

Comparison operators `==`, `!=` (not equal), `<`, `>`, `<=` (less or equal), `>=`.

`a = b`

Assignment, set variable `a` to value `b`. Not to be confused with `==` comparison. `a += b` is a shorthand for `a = a + b`, also for `-=` etc.

`a && b`

Logical operators — `&&` for AND, `||` for OR.

`-a`

Unary (one-operand) operator. `-` to negate, `!` for boolean negation.

Composite expressions

`a[b]`

Subscript, fetch the field named by `b` from value `a`.

`a.x`

Shorthand for `a["x"]`.

`a(b)`

Function call. Call the function value `a` with `b` as argument. Zero or more argument expressions can be given, separated by spaces. `a(1, 2, 3, 4)`

`a.x(b)`

Method call. Call the function found in field `x` of value `a`, **and** pass `a` as the `this` argument.

`[1, 2, 3, 4]`

Array value with zero or more elements.

```
{a: 1, b: 2}
```
Object value with zero or more `name: value` field definitions.

```
function(arg1, arg2) { /* ... body ... */ }
```
Function value. Zero or more argument names. Any statements may appear in body.

Statements

```
a;
```
Any expression, followed by a semicolon, is a statement.

```
{a; b; c;}
```
A series of statements, wrapped in braces, form a composite statement.

```
var a = b;
```
Variable definition. The variable with name `a` is defined and given value `b`. Value is optional. `var a;` sets `a` to undefined.

```
function foo(arg1, arg2) { /* ... body ... */ }
```
Function definition. Defines variable `foo` to have a function value. Zero or more arguments, any statements may appear in body.

```
if (a) { /* ... */ } else { /* ... */ }
```
Conditional statement. If value `a` is true, the first statement, otherwise the else statement executes. Else part may be left off. Can be chained as in `if (a) {} else if (b) {} else {}`.

```
while (a) { /* ... */ }
```
A loop. The loop body statement will be executed as long as `a` produces a true value.

```
for (var a = 0; a < 10; a = a + 1) { /* ... */ }
```
Example `for`-loop statement. `var a = 1` initializes the loop, `a < 10` checks whether it has ended yet, and `a = a + 1` moves to the next step.

```
return a;
```
Only valid inside a function. Returns value `a` as the result of the function call.

Useful functions

```
Number(v)
```
Converts `v` to a number. `Number("5")` gives `5`.

```
String(v)
```
Converts `v` to a string.

```
alert("hello")
```
Show a dialog window saying 'hello'.

```
confirm("are you sure?")
```
Show a yes/no dialog. Returns a `true`/`false` value indicating whether the user clicked yes.

```
prompt("what is your name?", "")
```
Show a dialog asking for input. First argument is the message, second argument is the initial value of the input.

Useful string properties

`"foo".length`

The length (number of characters) of the string.

`"foo".charAt(n)`

Get the character at position `n`. (Zero is the first character.)

`"foo".slice(from, to)`

Get a piece of the string. `"012345".slice(1, 4)` gives `"123"`.

`"a b c".split(" ")`

Split the string on a character, producing an array of strings (`["a", "b", "c"]`).

Useful array properties

`a[i]`

If `i` is an integer, this will access the element at that position.

`a.length`

The number of elements in the array.

`a.push(b)`

Add value `b` to the end of the array.

`a.pop()`

Remove the last element of the array, and return it.

`a.slice(from, to)`

Get a piece of the array, similar to the `slice` method on strings.

Useful math properties

`Math.random()`

Produce a random number between 0 and 1.

`Math.round(x)`

Round `x` to an integer.

`Math.abs(x)`

Returns the absolute (positive) value of `x`.

`Math.max(a, b, c, ...) Math.min(a, b, c, ...)`

Given any number of values, returns the greatest (`max`) or smallest (`min`) one.

`Math.PI`

The pi (π) constant.

`Math.cos(x) Math.sin(x) Math.tan(x)`

Trigonometric functions.

`Math.acos(x) Math.asin(x) Math.atan(x)`

Inverse trigonometric functions.