JavaScript

github.com/JKramerDD
Jeremy Kramer

Basic Syntax:

var x	Treated as though declared at top of function or global
let x	let declarations exist only in block
const x	Declares immutable value, block level
for(var x...	the value x will be accessible after the loop since ‘var’

Variable Declarations:

```
var stringx = “hello” | ‘hello’;
var charx = ‘h’;
var intx = [0-9]+;
var boolx = true | false;
var arrayx = [“x”, “y”, “z”];
var arrayx = new Array(“x”, “y”, “z”);
var objectx = {key1:”val1”, key2:”val2”, ... };
            objectx.key1 => val1
function myfunction(in1, in2){
            ...
}
var x = function(a,b){return a*b};
            x(4,3) => 12
```

Logic:

```
while ( condition ) { ... }
if  ( condition ) { ... }
else if  ( condition ) { ... }
else { ... }
switch ( expression ) {
            case label_x: ...
                        [break;]
            ...
            default:
                    [break;]
}
for( x = y; x < z; x++) { ... }
try { ... }
catch (e) { ... }
finally { ... }
```

Reg-Ex:

^	Start of string	a+	One or more of a
$	End of string	a+?	One or more, ungreedy
.	Any single character	a{3}	Exactly 3 of a
(a|b)	a or b	a{3,}	3 or more of a
(...)	Group section	a{,6}	Up to 6 of a
[abc]	In range (a, b or c)	a{3,6}	3 to 6 of a
[^abc]	Not in range	a{3,6}?	3 to 6 of a, ungreedy
\s	White space	\	Escape character
a?	Zero or one of a	[:punct:]	Any punctuation
a*	Zero or more of a	[:space:]	Any space character
a*?	Zero or more, ungreedy	[:blank:]	Space or tab

Array Functions:

arr1.concat(arr2);	(array)	arr.pop();	(var)
arr.filter(func);	(array)	arr.push(var);	()
arr.find(func);	(int)	arr.reduce(func);	(var)
arr.findIndex(func);	(int)	arr.reverse();	()
arr.forEach(func);	()	arr.slice(int, int);	(array)
arr.join();	(string)	arr.splice(int, int, var...);	()
arr.isArray();	(bool)	arr.toString();	()
arr.lastIndexOf(func);	(int)		

String Functions:

str.charAt(int);	(char)
str.concat(str2);	(str)
str.endsWith(expr);	(bool)
str.includes(expr);	(bool)
str.indexOf(str);	(int)
str.lastIndexOf(str);	(int)
str.match(regex);	(array)
str.repeat();	(str)
str.replace(str/regex, str);	(str)
str.search(str/regex);	(int)
str.substr(int, int);	(str)
str.toLowerCase();	(str)
str.toUpperCase();	(str)
var.toString();	(str)
str.trim();	(str)

Math Functions:

num.abs()
num.acos()
num.asin()
num.atan()
num.ceil()
num.floor()
num.cos()
num.sin()
num.tan()
num.exp()
num.max()
num.min()
num.pow()

num.random()
num.round()
num.sqrt()
num.toExponential()
num.toFixed()
num.toPrecision()
num.toString()
num.MAX_VALUE()
num.MIN_VALUE()
num.NEGATIVE_INFINITY()
num.POSITIVE_INFINITY()
num.isNaN()

Reg-Ex Modifiers:

g	Global match	x *	Allow comments and whitespace in pattern
i *	Case-insensitive		
m *	Multiple lines	e *	Evaluate replacement
s *	Treat as single line	U *	Ungreedy pattern