

Values		
Operators	Strings	Weird Stuff
+, -, *, /, %	"stringing"	infinity
+=, -=, *=, /=	'string'	-infinity
++, --	`string`	NaN
<, <=, >, >=	.length	null, undefined
&&, , !	\n	
typeof()	+ (concat)	
state ? 1 : 2	\t (tab)	
==, != (convert)	\"	
===, !==	\\	
-Backtick-quoted (template literals) ables: `fourteen is also \${10+4}` // fourteen is also 14		
-Empty strings convert to false when parse to boolean: "" = false		
-let ,const & var scope with funtions		
-var doesnt scope with loops		
-let & const scope with loops		

Program Structure	
Conditional	Useful Things
if(){else if(){else{}}	break
while(){}	continue
do{ while{}	//, /* */
for(let i=0;i<x;i++){}	
switch (x) { case 1: break; case "cloudy": console.log("Bring umbrella."); break; default: break; }	

Function
// Define a function const makeNoise = function() { console.log(" Pli - ng! "); }; // Declaration notation function makeNoise { console.log(" Pli - ng! "); }; // Arrow function const makeNoise = () => { console.log(" Pli - ng! "); }

-Default value of parameters is undefined.

-Default value of parameters can be change:

```
function power(base, exponent = 2) {
  ...
}
```

Variables Control	
Conversion	Binding
Number()	let, var, const
String()	prompt()
Boolean()	
-A single bind allows multiple values: let x = 0, y = 'dog';	
-Bind names can't begin with numbers: let 4number = 4; WRONG!	
- Bind names may contain: letters, numbers, (\$), (_)	

Create Variables
let string = "x"
let number = 2
let boolean = true
let array = [0, 1]
let object = { happy : true, age : 20 }

Number Object
Number Methods
Number.isNaN()



By AlvaroRockster

Not published yet.

Last updated 5th April, 2018.

Page 1 of 1.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>