**Javascript basics**

**FUNDAMENTALS PART 1**

1. How do you declare a variable?
2. What are three different ways to declare a variable?
3. Which one should you use when?
4. What are the rules for naming variables?
5. What are operators, operands, and operations?
6. What is concatenation and what happens when you add numbers and strings together?
7. What are the different type of operators in JavaScript?
8. What is the difference between == and ===?
9. What are operator precedence values?
10. What are the increment/decrement operators?
11. What is the difference between prefixing and post-fixing them?
12. What are assignment operators?z
13. What is the “Unary +” Operator?
14. **JavaScript Arithmetic Operators**

|  |  |  |  |
| --- | --- | --- | --- |
| **DIE MD MAS** | | | |
| **D** | Decrement | -- | Right To Left |
| **I** | Increment | ++ | Right To Left |
| **E** | Exponentiation | \*\* | Right To Left |
| **M** | Modulus | % | Left To Right |
| **D** | Division | / | Left To Right |
| **M** | Mutiplication | \* | Left To Right |
| **A** | Addition | + | Left To Right |
| **S** | Subtraction | - | Left To Right |

Operand are the number in an arithmetics operation. (**2 + 2**)

Operator are the arithmetics sign that perform the action between the operand (**2 + 2**)

**JavaScript Operator Precedence Values**

New ECMAScript 2015 (ES6) operator

\*\*

in

instanceof

yield

**Method**

var x = 5;

var z = Math.pow(x,2); // result is 25

Math.pow(x,2)

is the same as var z = x\*\*2; // result is 25

1. **JavaScript Numbers**

JS has only one type of number. With and without decimal

JavaScript Numbers are Always 64-bit Floating Point

JavaScript numbers are always stored as double precision floating point numbers, following the international IEEE 754 standard.

This format stores numbers in 64 bits, where the number (the fraction) is stored in bits 0 to 51, the exponent in bits 52 to 62, and the sign in bit 63:

The maximum number of decimals is 17, but floating point arithmetic is not always 100% accurate:

var x = 0.2 + 0.1;

Adding Numbers and Strings

WARNING !!

JavaScript uses the + operator for both addition and concatenation.

Numbers are added. Strings are concatenated.

**NaN - Not a Number**

NaN is a JavaScript reserved word indicating that a number is not a legal number.

use global JavaScript function isNaN() to find out if a value is a number

var x = 100 / "Apple";

isNaN(x); // returns true because x is Not a Number

NaN is a number: typeof NaN returns number

typeof NaN; // returns "number"

**Infinity**

Infinity (or -Infinity) is the value JavaScript will return if you calculate a number outside the largest possible number.

Division by 0 (zero) also generates Infinity:

var x = 2 / 0; // x will be Infinity

var y = -2 / 0; // y will be -Infinity

Infinity is a number: typeof Infinity returns number.

typeof Infinity; // returns "number

**Numbers Can be Objects**

Normally JavaScript numbers are primitive values created from literals:

var x = 123;

But numbers can also be defined as objects with the keyword new:

var y = new Number(123);

== only check the value

=== operator check the value and type

JS object can not be compare.

Comparing JS object will always return false