JS

javascript

```
> typeof NaN
"number"
> 9999999999999999
10000000000000000
> 0.5+0.1==0.6
true
> 0.1+0.2==0.3
false
> Math.max()
-Infinity
> Math.min()
Infinity
> []+[]
""
> []+{}
"[object Object]"
> {}+[]
0
> true+true+true===3
true
> true-true
0
```

```
> true==1
true
> true===1
false
> (!+[]+[]+![]).length
9
> 9+"1"
"91"
> 91-"1"
90
> []==0
true
```

Java vs JavaScript

объект-ориентированный язык	мультипарадигменный язык
запускается на сервере	браузеры и серверы
строго типизирован	не типизирован

JavaScript === ECMAScript

JavaScript

Javascript на одной картинке

Переменные

Переменные

var	устаревшее	
let	можно изменять	EcmaScript 2015 года (ES6)
const	нельзя изменять	EcmaScript 2015 года (ES6)

Примитивные типы

Числа

number $-(2^{53} - 1)$ до $(2^{53} - 1)$, Infinity, -Infinity, NaN

-9007199254740991 — +9007199254740991

дроби -> .toFixed()

infinity -> 1e999

NaN -> .isNan()

BigInt

```
const int = 4
```

Строки

```
const double    = ""
const single    = ''
const literals  =`текст ${переменная}`
```

Boolean

True / false

Бинарные (0) (1)

undefined / null

undefined	неизвестное или неопределённое значение
null	отсутствие (намеренно) значения → задается программистом

Symbol

Symbol создания скрытых свойств объектов

Примитивы

Boolean
Null
Undefined
Number
BigInt
String
Symbol

Всё остальное объекты

Преобразование типов

Явные

```
String(42)
Number("42")
Boolean(42)
```

Неявные

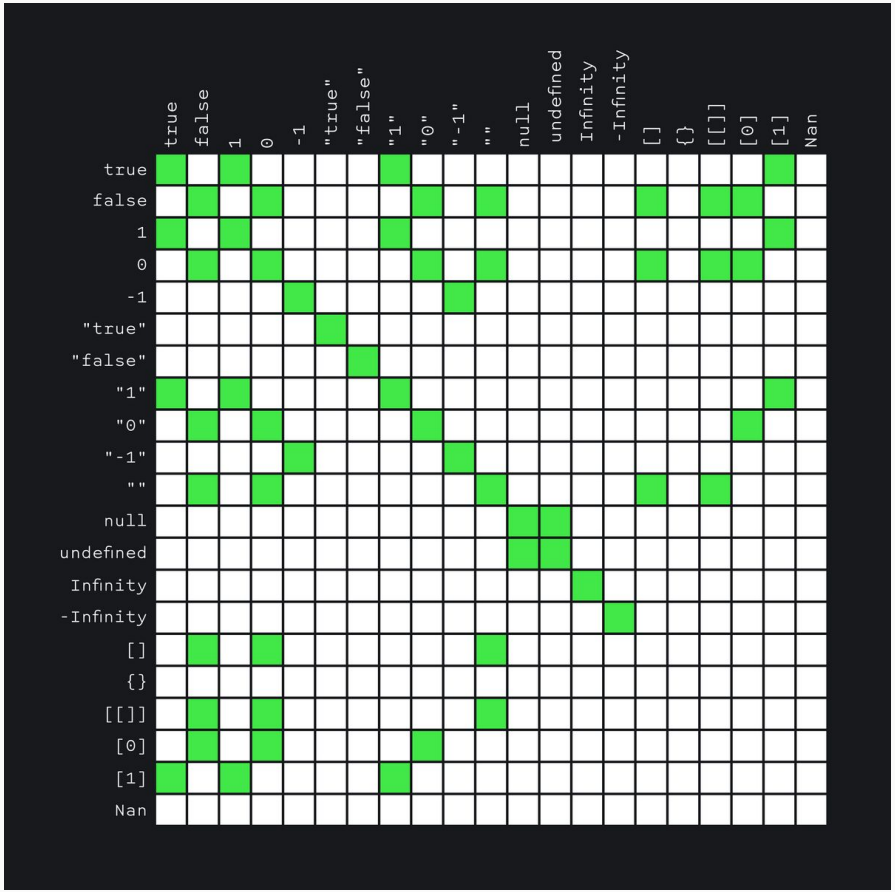
логические –	&& и ||
строка –	+ и один операндов строка
число –	<, <=, >, >=
	/ * - + (+ если не противоречит правилу строк)

4+5+6+”7”

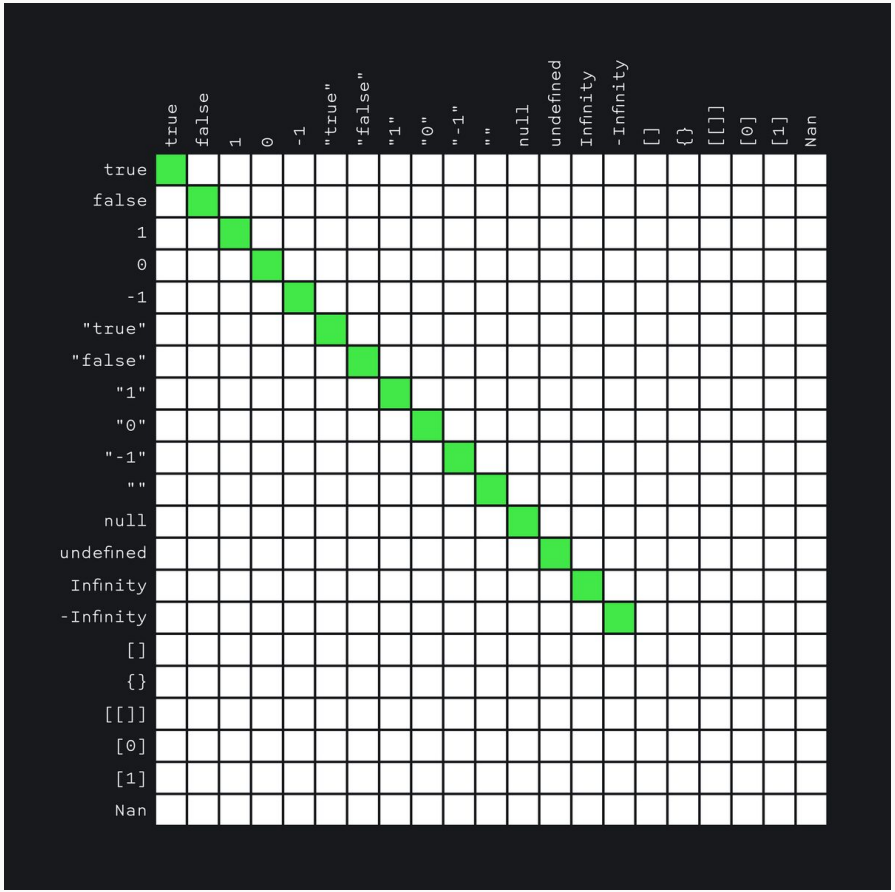
Строгое и нестрогое равенство

(==)	Пытается привести типы к одному
(===)	Нет изменений в типах

Нестрогое (==)


true
false
1
0
-1
"true"
"false"
"1"
"0"
"-1"
""
null
undefined
Infinity
-Infinity
[]
{}
[[]]
[0]
[1]
Nan

Строгое (===)


true
false
1
0
-1
"true"
"false"
"1"
"0"
"-1"
""
null
undefined
Infinity
-Infinity
[]
{}
[[]]
[0]
[1]
Nan

Массив

массив коллекция элементов

const array = []

const zeka = [‘zeka’, 30, ‘developer’]

push —> для добавления в конец массива

unshift —> для добавления в начало массива

indexOf —> индексом элемента.

includes —> элемент есть в массиве:

Объекты

набор свойств

```
const cat = {}

const zeka = {
        name: 'zeka',
        age: 30,
        work: 'developer',
        company: {
                name: 'Intexsys',
                title:   'Fe developer',
        },
}

zeka.keyName = 2
delete zeka.name
```

Функции

```
function (параметры) {
        тело функции
}
```

```
const myFunction = function() {
  console.log('In')
}

myFunction()
```

Логические операторы

Логические операторы

&& -	логический оператор И
||	логический оператор ИЛИ
!	логический оператор НЕ

Циклы

ЦИКЛЫ

if else
switch
while
for
try catch

If … else

```
if (условие) {

} else {

}
```

switch

```
switch (имя_переменной_значение_которой_сравниваем) {
  case значение:
    // код
    break
}
```

While

```
while(условие) {
    // код
}
```

For

```
for (инициализация; условие; завершающая операция ) {
  // тело цикла
}
```