**Приведення в boolean - використовуються функція Boolean()**

|  |  |
| --- | --- |
| **Приведется к true** | **Приведется к false** |
| **не пуста строка,**  **число яке не дорівнює 0**  **будь-який object** | **Пуста строка ("" '')**  **Число яке дорівнює 0**  **NaN null undefinded** |

**Приведення в number – функції Number(),parseInt(),parseFloat()**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Number()** | **parseInt()** | **parseFloat()** |
| **Boolean** | **true -> 1, false-> 0** | **NaN** | **NaN** |
| **null** | **0** | **NaN** | **NaN** |
| **undefined** | **NaN** | **NaN** | **NaN** |
| **String** | | | |
| **"10"** | **10** | **10** | **10** |
| **"000011"** | **11** | **11** | **11** |
| **"-1.23"** | **-1.23** | **-1** | **-1.23** |
| **"0xf"** | **15** | **15** | **0** |
| **"", " "** | **0** | **NaN** | **NaN** |
| **"123px"** | **NaN** | **123** | **123** |
| **"abc123"** | **NaN** | **NaN** | **NaN** |
| **"Hello world"** | **NaN** | **NaN** | **NaN** |

**Приведення в строку – функція String() або метод toString()**

|  |  |  |
| --- | --- | --- |
|  | **toString()** | **String()** |
| **true или false** | **"true" или "false"** | **"true" или "false"** |
| **null** | **error** | **"null"** |
| **undefined** | **error** | **"undefined"** |
| **123** | **"123"** | **"123"** |
| **-12.3** | **"-12.3"** | **"-12.3"** |

**При арифметичних операціях проводиться неявнє приведення типів**

|  |
| --- |
| **Для операції \* / % -**  *Якщо будь-який з операндів не є типом* ***number****, то інтерпретатор робить*  *його неявне приведення до числа за правилами функції* ***Number()***  **Для операції +**  **-** якщо серед операндів не має операнда з типом **string** то вони приводяться  до типу **number** за правилом **Number();**  **-** якщо обидва операнди мають тип **string** – проводиться їх конкатенація;  **-** якщо один из операндов має тип  **string,** то другий приводиться до  строкового типу використовуючи правила функції **String**(),  а вже потім проводиться їх конкатенація. |

**Логичні оператори**

|  |  |  |  |
| --- | --- | --- | --- |
| **p** | **q** | **p && q** | **p || q** |
| **true** | **true** | true | true |
| **true** | **false** | false | true |
| **false** | **true** | false | true |
| **false** | **false** | false | false |

**Приведення типів при логичниї операціях**

**!!! Логічні операції | та && приводять значення операндів до логічного типу, але повертають вихідні операнди**

|  |  |
| --- | --- |
| **expr1 && expr2** | **if**(**expr1 === true ) return expr2**  **else return expr1** |
| **expr1 || expr2** | **if**(**expr1 === true ) return expr1**  **else return expr2** |

**Оператор NOT (!)**

|  |  |
| --- | --- |
| **!expr** | Поверне **false** якщо **expr** може бути перетворений на **true**, інакше поверне **true**. |

**Приведення типів при операціях порівняння < <= > >=**

При порівнянні різних типів інтерпретатор намагається перетворити операнди на числа за правилами функції **Number()**

**Виняток:** коли обидва операнди — рядки, ***тоді вони не перетворюються***.

|  |  |
| --- | --- |
| **Что сравниваем** | **Правила сравнения** |
| **строка - строка** | Посимвольно порівнюються коди чисел |
| **число – число** | Числа порівнюються арифметично |
| **bool - bool** | Порівнюються як числа **true = 1 false = 0** |
|  | |
| **строка – число** | Строка приводится до типу **number** за правилами **Number()** |
| **bool – число** | Boolean тип приводится до типу **number** за правилами **Number()** |
| **cтрока - bool** | Обидва приводяться до типу **number** за правилами **Number()** |
| **один з операндів NaN** | Результат порівняння **false** |

**--------------------------------------------------------------------------**

**Правила порівняння при операторі ==** *(equality)*и **!=** *(not equality)*

|  |  |
| --- | --- |
| **boolean - любий тип** | Приводяться до числа за правилами **Number()** |
| **string - number** | Строка приводится до **number** за правилами **Number()** |
|  | |
| Значення **null** та **undefined** рівні один одному, але не дорівнюють чомусь іншому.  **null == undefined** // true  **null === undefined** // false | |
| Значення **null** та **undefined** не можуть бути перетворені на інші типи  під час операцій **equality.**  null == 0 **// false**  "" == 0 **// true** | |
| Якщо операнд **NaN ,** то операція **==** поверне **false**  а операція **!=** поверне  **true** | |
| Якщо обидва операнда **object,** то порівнюється - це один і той же **object ?** | |