Лекция №2

Основы JS

Приведение типов

Приведение типов

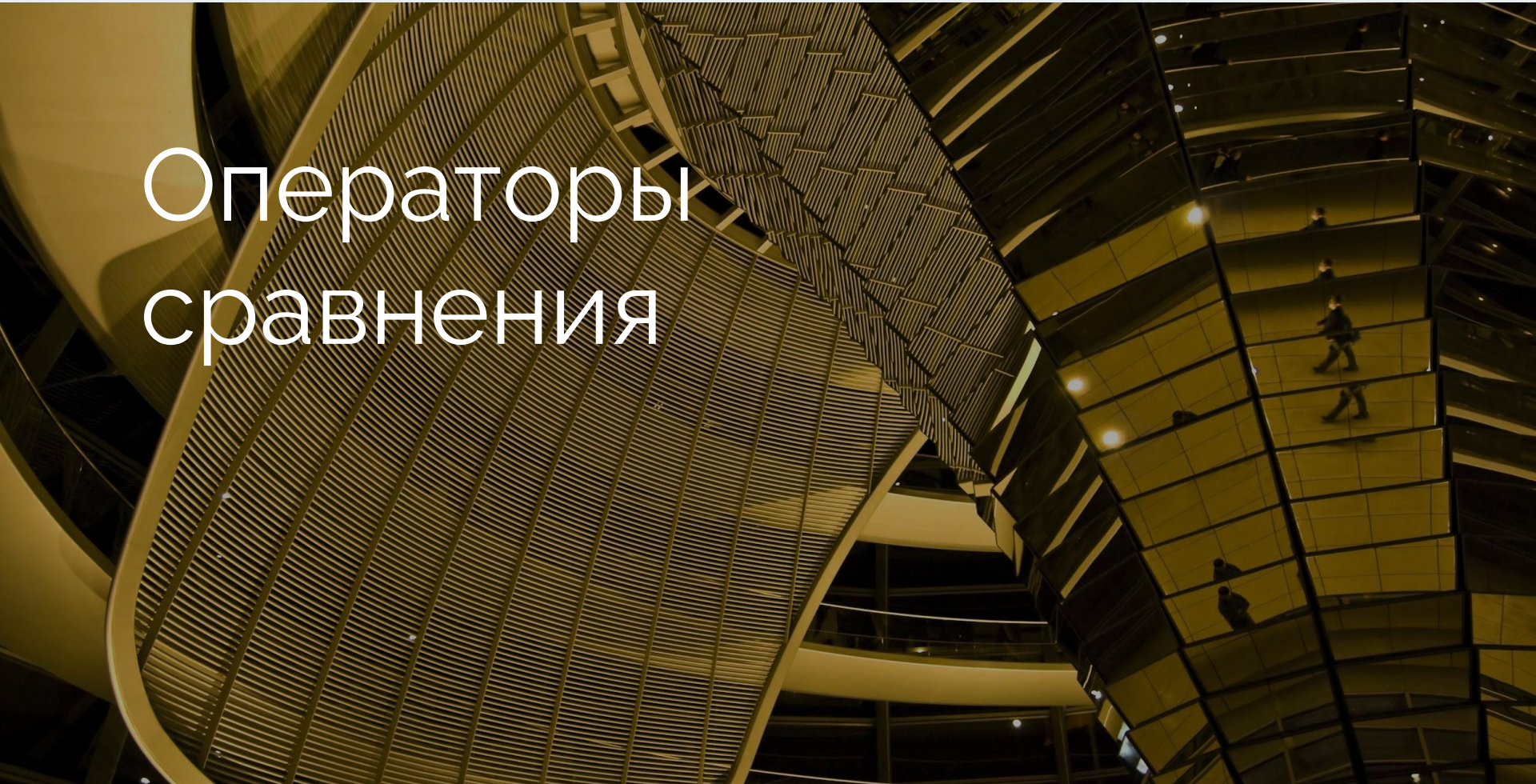
Приведение типов (type coercion) — это автоматическое или неявное преобразование значений из одного типа данных в другой (например, строки в число). *Type conversion* похоже на *приведение типа*, потому что они оба преобразуют значения из одного типа данных в другой с одним ключевым различием — *приведение типа* выполняется неявно, тогда как преобразование типа может быть неявным *или* явным.


Булевые значения

Описание

Значение, переданное первым параметром, при необходимости преобразуется в логическое значение. Если значение опущено или равно `0`, `-0`, `null`, `false`, `NaN`, `undefined` или пустой строке (`""`), объект имеет начальное значение, равное `false`. Все остальные значения, включая любые объекты или строку `"false"`, создают объект с начальным значением, равным `true`.

Не путайте примитивные значения `true` и `false` логического типа со значениями `true` и `false` объекта `Boolean`.



Операторы сравнения

JavaScript предоставляет три оператора сравнения величин:

- равенство ("двойное равно") использует `==`,
- строгое равенство (или "тройное равно" или "идентично") использует `===`,
- и `Object.is` (новшество из ECMAScript 6).

Выбор оператора зависит от типа сравнения, которое необходимо произвести.

В общих чертах, двойное равно перед сравнением величин производит приведение типов; тройное равно сравнивает величины без приведения (если величины разных типов, вернёт `false`, даже не сравнивая); ну и `Object.is` ведёт себя так же, как и тройное равно, но со специальной обработкой для `NaN`, `-0` и `+0`, возвращая `false` при сравнении `-0` и `+0`, и `true` для операции `Object.is(NaN, NaN)`. (В то время как двойное или тройное равенство вернут `false` согласно стандарту IEEE 754.) Следует отметить, что все эти различия в сравнениях применимы лишь для примитивов. Для любых не примитивных объектов `x` и `y`, которые имеют одинаковые структуры, но представляют собой два отдельных объекта (переменные `x` и `y` не ссылаются на один и тот же объект), все операторы сравнения вернут `false`.

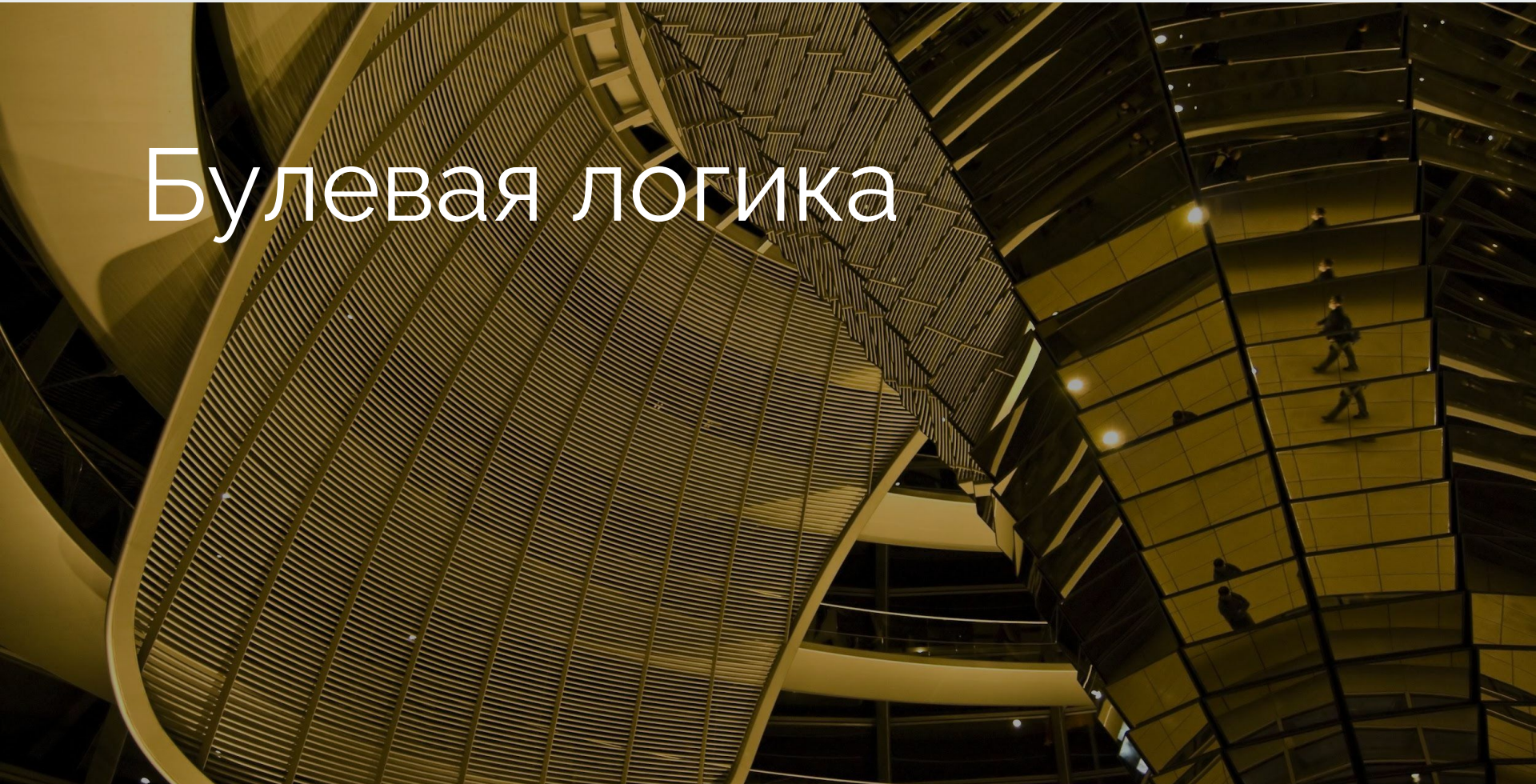
Сравнение с использованием `==`

Перед сравнением оператор равенства *приводит* обе величины к общему типу. После приведений (одного или обоих операндов), конечное сравнение выполняется также как и для `===`. Операция сравнения *симметрична*: `A == B` возвращает то же значение, что и `B == A` для любых значений `A` и `B`.

Строгое равенство с использованием `===`

Строгое равно проверяет на равенство две величины, при этом тип каждой из величин перед сравнением не изменяется (не приводится). Если значения имеют различающиеся типы, то они не могут быть равными. С другой стороны все не числовые переменные, принадлежащие одному типу, считаются равными между собой, если содержат одинаковые величины. Ну и, наконец, числовые переменные считаются равными, если они имеют одинаковые значения, либо одна из них `+0`, а вторая `-0`. В то же время, если хотя бы одна из числовых переменных содержит значение `NaN`, выражение вернёт `false`.

Булевая логика



Логическое И (&&)

Логический оператор И (`&&`) (конъюнкция) для набора операндов со значением типа `Boolean` будет `true` только в случае, если все операнды содержат значение `true`. В противном случае это будет `false`.

В целом, оператор вернёт значение первого ложноподобного операнда при вычислении, либо значение последнего операнда, если все операнды оказались истиноподобными.

Описание

Логическое И (`&&`) вычисляет операнды слева направо, возвращая сразу значение первого попавшего ложноподобного операнда; если все значения истиноподобны, возвращается значение последнего операнда.

Если значение может быть преобразовано в `true`, то оно рассматривается как истиноподобное (truthy). Если же значение может быть преобразовано в `false`, то оно называется ложноподобным (falsy).

Логическое ИЛИ (||)

Логический оператор ИЛИ (`||`) (дизъюнкция) для набора операндов истинен будет `true` только в случае, если один или несколько его операндов имеют значение `true`.

Обычно используется с булевыми (логическими) значениями. Тогда возвращается булевое значение. Однако фактически оператор `||` возвращает значение одного из операндов, поэтому если этот оператор используется с небулевыми значениями, он вернет небулевое значение.

Описание

Если `expr1` может быть преобразовано в `true`, то вернётся `expr1`; в противном случае возвращается `expr2`.

Если значение может быть преобразовано в `true`, то оно рассматривается как истиноподобное (truthy). Если же значение может быть преобразовано в `false`, то оно называется ложноподобным (falsy).

Приоритет операторов

Оператор И имеет более высокий приоритет, чем оператор ИЛИ, поэтому оператор `&&` выполнится раньше оператора `||` (см. приоритет операторов).

```
false || true && true             // вернёт true
true && (false || false)          // вернёт false
(2 == 3) || (4 < 0) && (1 == 1)   // вернёт false
```

Полезные ссылки

https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Operators/Logical_OR
https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Operators/Logical_AND

Спасибо!