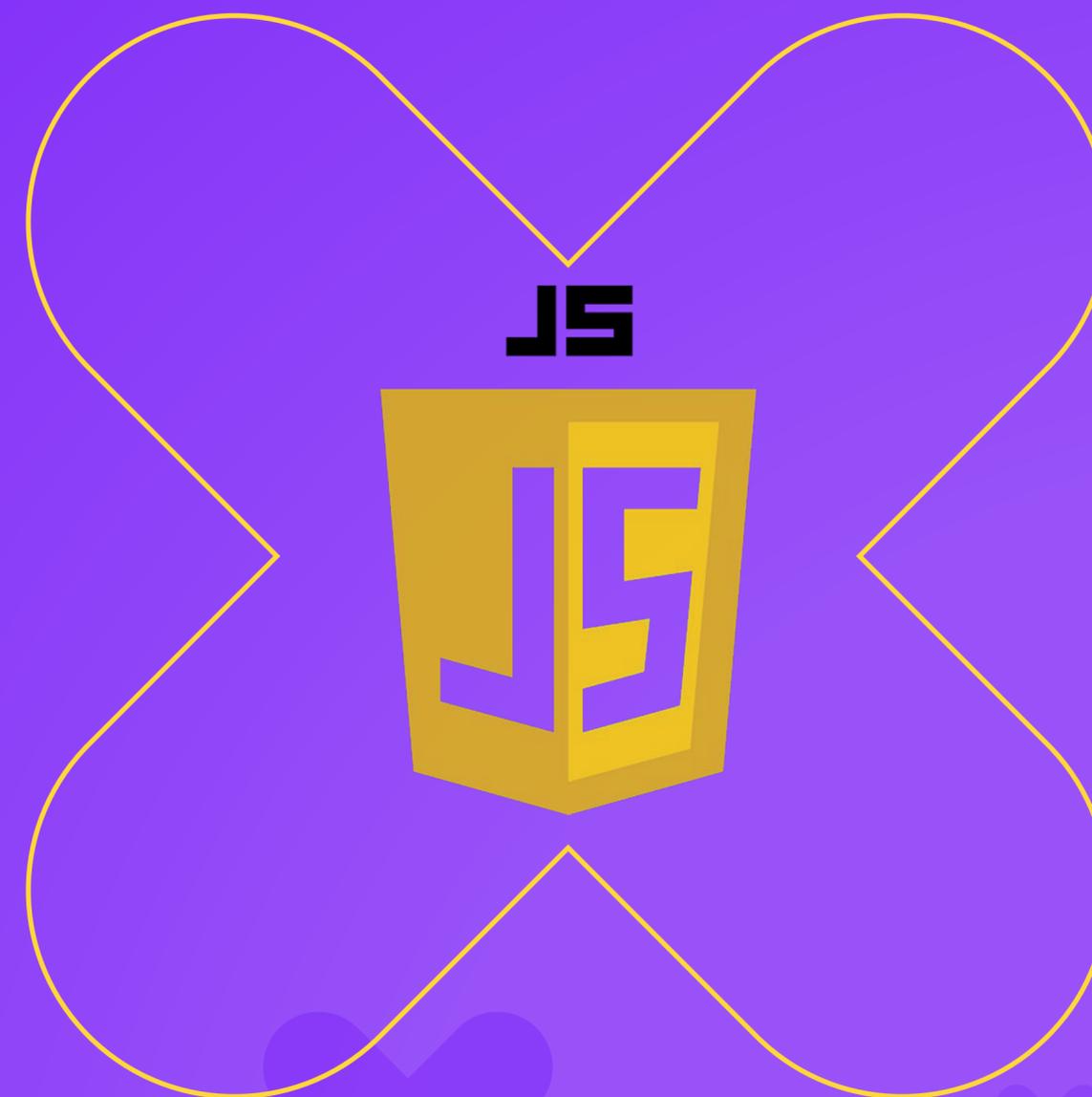




ЗАНЯТИЕ №3-4

# Условные операторы

Логические операции



# Логические операции

# Булевские переменные

---

Булевские переменные имеют только 2 доступных значения - true или false (истина или ложь).

Название берется из понятия булева алгебра (Дж. Буль - математик, один из основателей математической логики)

# Операции над булевыми значениями

---

! - логическое отрицание, инверсия, НЕ

&& - логическое умножение, конъюнкция, И

|| - логическое сложение, дизъюнкция, ИЛИ

^ - исключающее ИЛИ

# ! - логическое отрицание, инверсия, НЕ

---

A

true

false

!A

false

true

# && - логическое умножение, конъюнкция, И

A

B

A && B

true

false

false

false

true

false

true

true

true

false

false

false

# || - логическое сложение, дизъюнкция, ИЛИ

---

A	B	A    B
true	false	true
false	true	true
true	true	true
false	false	false

## ^ - исключающее ИЛИ

A	B	$A \wedge B$
true	false	true
false	true	true
true	true	false
false	false	false

# Операции, дающие булево значение как результат

>	больше
<	меньше
>=	больше или равно
<=	меньше или равно
==	проверка равенства без проверки типов
===	проверка равенства с проверкой типов
!=	проверка неравенства без проверки типов
!==	проверка неравенства с проверкой типов

# Условный оператор

---

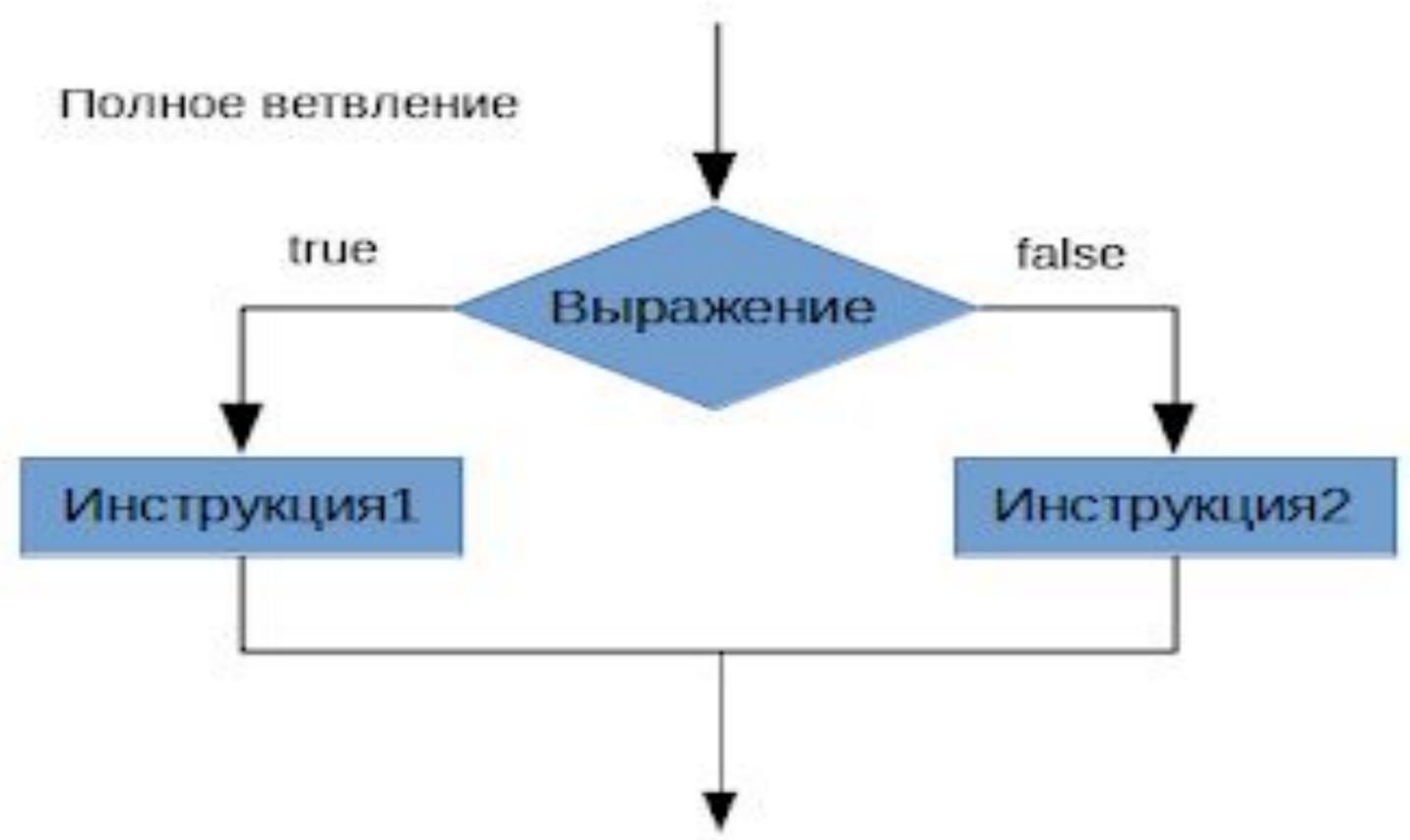


Условный оператор используется для проверки условия. Если условие выполняется, то выполняется одно действие, если не выполняется - то другое.

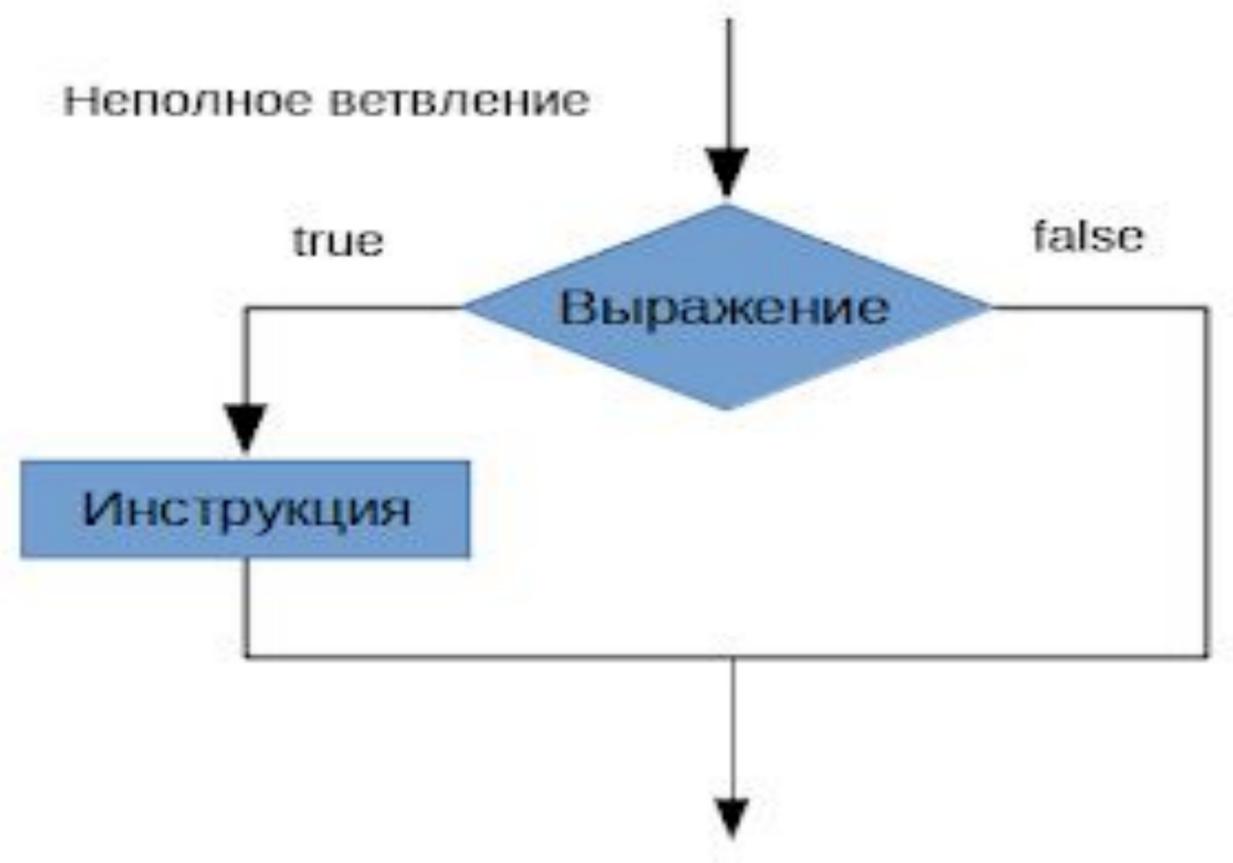


Условный оператор может реализовать неполное ветвление, то есть если условие не выполняется, то не выполняется никаких действий.

Полное ветвление



Неполное ветвление



# Условный оператор (неполное ветвление)

```
if (statement) {  
    //операторы  
}
```

**statement** - это выражение, результатом которого будет булевское значение.

Так как Javascript имеет динамическую типизацию, то **statement** может быть любого типа, однако в операторе **if** произойдет преобразование в **true** или **false**

# Программные скобки

---

Фигурные скобки `{}` могут использоваться для группировки нескольких действий. По умолчанию после `if ()` должно быть записано только одно выражение. Однако, с фигурными скобками при верном условии может быть несколько действий.

# Условный оператор (полное ветвление)

```
if (statement) {  
    //действие 1  
}  
else {  
    //действие 2  
}
```

# else

---

Блок кода после `else` выполняется в случае, если выражение в `if` имеет значение `false`, то есть если условие не выполняется.

Конец

