«Ветвление, логические операции»

JavaScript Courses

www.courses.dp.ua

JavaScript как язык программирования

его

концепции

Переменные / Типы / Операции

Ветвления (условные операторы)

Циклы / Массивы (структуры данных)

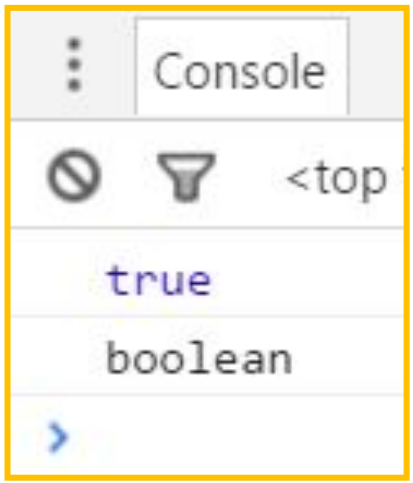
Функции

И

Объекты

Тип Boolean

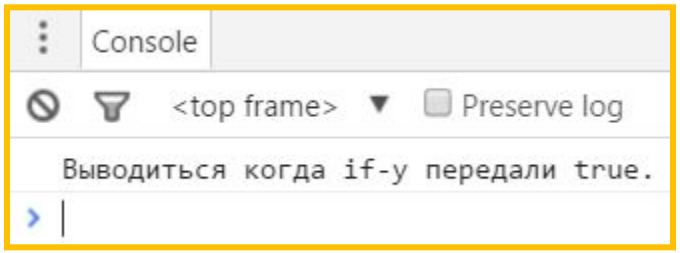
```
1  <script>
2
3      var a = true;
4
5      var b = false;
6
7      console.log(a);
8
9      console.log(typeof(a));
10
11 </script>
```



*Переменная типа **boolean** содержит один из всего 2 вариантов значения: истина (**true**) или ложь (**false**).*

Оператор if-else

```
1  <script>
2
3      var bool_type_variable = true; //или false
4
5      if( bool_type_variable ) {
6
7          console.log("Выводиться когда if-у передали true.");
8
9      }else{
10
11         console.log("Выводиться когда if-у передали false.");
12     }
13 </script>
```



*Оператор **if-else** в зависимости от переданного (**true** или **false**) значения выполняет один из двух блоков кода (**первый** или **второй**, соответственно), другой блок при этом не выполняется.*

Операторные скобки { } и блок кода

```
1  <script>
2
3      var a = true;
4
5      if(a){
6          alert("#1");
7          alert("#2");
8      };
9  </script>
```

?!?

```
1  <script>
2
3      var a = true;
4
5      if(a)
6          alert("#1");
7          alert("#2");
8
9  </script>
```

Фигурные скобки формируют блок из нескольких (от одной и больше) строк кода, который для условных операторов (и циклов) выглядит как единая команда.

*Переменные объявленные с использованием **let** внутри блока по завершению блока уничтожаются.*

Оператор if-else

```
<script>

    var bool_type_variable = true; //или false

    if( bool_type_variable ) {

        console.log("Выводиться только когда if-у передан true.");

    };
</script>
```

*Блок **else** – является необязательным.*

Откуда берётся boolean?

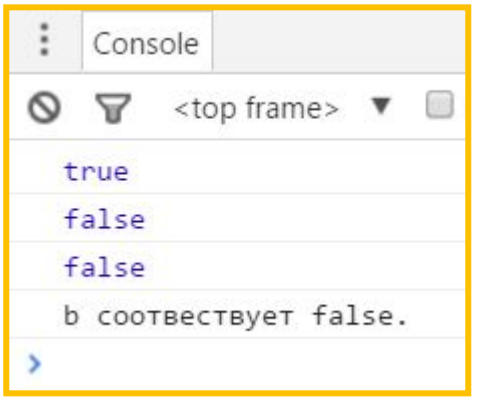
*undefined, NaN => **false**;*

***Number:** 0 => **false**; все остальные => true;*

***String:** “” => **false**; все остальные => true;*

Object: null => **false**; существующий объект => true;

```
1  <script>
2
3      var a = 23;
4      var b = "";
5      var c = null;
6
7      console.log(Boolean(a));
8      console.log(Boolean(b));
9      console.log(Boolean(c));
10
11     if(b){
12         console.log("b соотвествует true.");
13     }else{
14         console.log("b соотвествует false.");
15     }
16
17 </script>
```

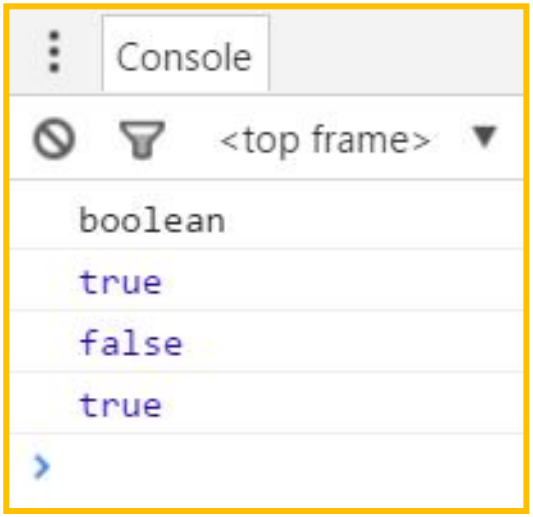


Из преобразование типов.

Откуда берётся boolean?

>	<	>=	<=	==	!=	!	===	!==

```
1  <script>
2
3      var a = 3 < 4;
4      var b = 56 == 77;
5      var c = "UA" == "UA";
6
7      console.log(typeof(a));
8      console.log(a);
9      console.log(b);
10     console.log(c);
11
12 </script>
```



Операторы сравнения

Откуда берётся boolean?

>	<	>=	<=	==	!=	!	===	!==

```
<script>

    var x = 6;

    var y = 8;

    var bool_var = x > y;

    if(bool_var){
        //..
    }else{
        //...
    }

</script>
```

=

```
<script>

    var x = 6;

    var y = 8;

    if(x > y){
        //..
    }else{
        //...
    }

</script>
```

«Каноническая» запись

Откуда берётся boolean?

>	<	>=	<=	==	!=	!	===	!==

```
1  <script>
2
3      var a = 6;
4      var b = 51;
5      var c = "51";
6      var d = "6";
7
8      console.log(a > b);
9      console.log(a > c);
10     console.log(d > c);
11
12 </script>
```

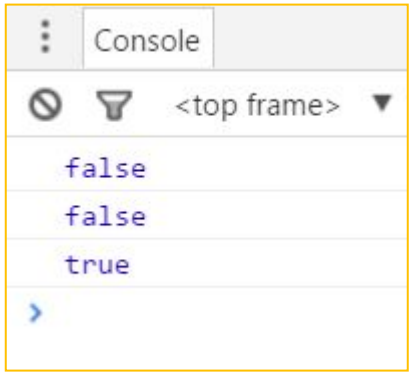
?!?

Откуда берётся boolean?

>	<	>=	<=	==	!=	!	===	!==

Сравнение строк осуществляется посимвольно. Сравниваться коды символов в таблице кодировки.

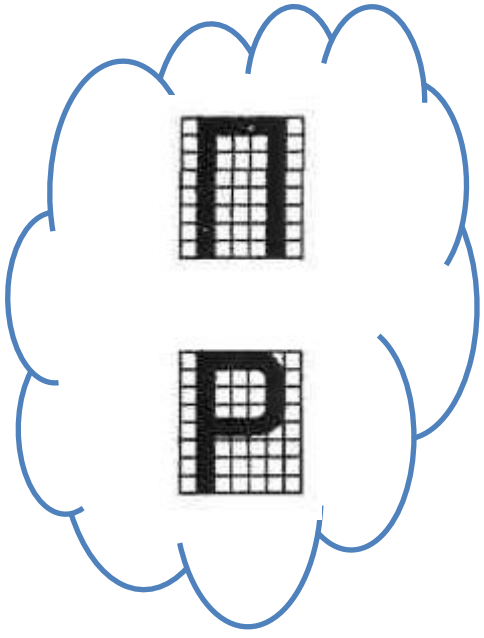
```
1  <script>
2
3      var a = "Ivan";
4      var b = "Iven";
5
6      console.log(a == b);
7      console.log(a > b);
8      console.log(a < b);
9
10 </script>
```

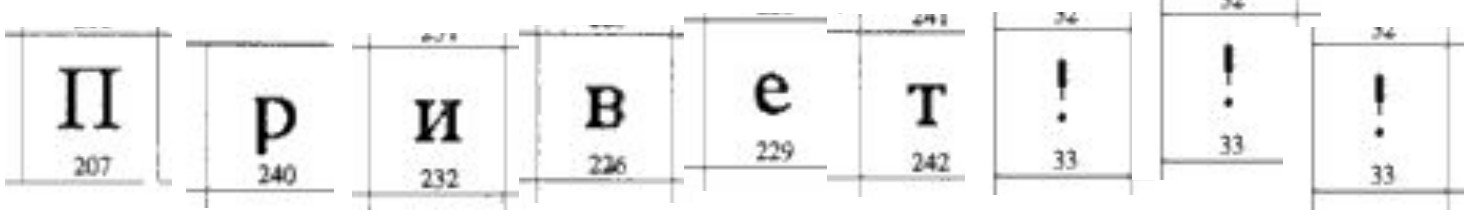


Кодировка

	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
0	32	0 48	@ 64	P 80	` 96	p 112	Ъ 128	ђ 144	NBSP 160	° 176	А 192	Р 208	а 224	р 240
1	! 33	1 49	A 65	Q 81	a 97	q 113	Ѓ 129	‘ 145	Ў 161	± 177	Б 193	С 209	б 225	с 241
2	" 34	2 50	B 66	R 82	b 98	r 114	, 130	’ 146	ў 162	І 178	В 194	Т 210	в 226	т 242
3	# 35	3 51	C 67	S 83	c 99	s 115	ѓ 131	“ 147	Ј 163	і 179	Г 195	У 211	г 227	у 243
4	$ 36	4 52	D 68	T 84	d 100	t 116	„ 132	” 148	¤ 164	ґ 180	Д 196	Ф 212	д 228	ф 244
5	% 37	5 53	E 69	U 85	e 101	u 117	… 133	• 149	Ґ 165	µ 181	Е 197	Х 213	е 229	х 245
6	& 38	6 54	F 70	V 86	f 102	v 118	† 134	- 150	¦ 166	¶ 182	Ж 198	Ц 214	ж 230	ц 246
7	' 39	7 55	G 71	W 87	g 103	w 119	‡ 135	– 151	§ 167	· 183	З 199	Ч 215	з 231	ч 247
8	(40	8 56	H 72	X 88	h 104	x 120	NOT USED 136	NOT USED 152	Ё 168	ё 184	И 200	Ш 216	и 232	ш 248
9	) 41	9 57	I 73	Y 89	i 105	y 121	‰ 137	™ 153	© 169	№ 185	Й 201	Щ 217	й 233	щ 249
A	* 42	: 58	J 74	Z 90	j 106	z 122	Љ 138	љ 154	Є 170	є 186	К 202	Ъ 218	к 234	ъ 250
B	+ 43	; 59	K 75	[91	k 107	{ 123	‹ 139	› 155	« 171	» 187	Л 203	Ы 219	л 235	ы 251
C	, 44	< 60	L 76	\ 92	l 108	| 124	Њ 140	њ 156	¬ 172	ј 188	М 204	Ь 220	м 236	ь 252
D	- 45	= 61	M 77	] 93	m 109	} 125	Ќ 141	ќ 157	SHY 173	S 189	Н 205	Э 221	н 237	э 253
E	. 46	> 62	N 78	^ 94	n 110	~ 126	Ћ 142	ћ 158	® 174	s 190	О 206	Ю 222	о 238	ю 254
F	/ 47	? 63	O 79	_ 95	o 111	127	Џ 143	џ 159	Ї 175	ї 191	П 207	Я 223	п 239	я 255

Файл Правка Вид Кодировка Справка
00000000: CF F0 E8 E2 E5 F2 21 21 21

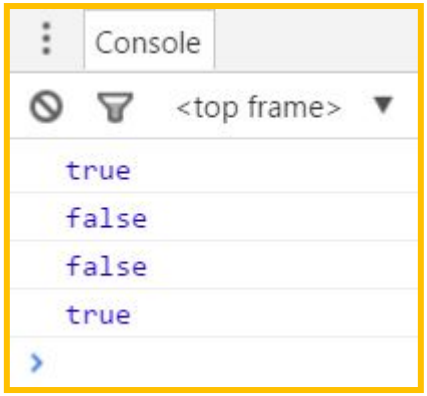




Откуда берётся boolean?

>	<	>=	<=	==	!=	!	===	!==

```
<script>

    var a = "5";
    var b = 5;

    console.log(a == b);

    console.log(a === b);

    console.log((typeof(a) == typeof(b)) && (a == b));

    console.log(typeof(a) == "string");

</script>
```



Сравнение с учетом типа

Откуда берётся boolean?

```
<script>

    var user_email = prompt("Enter email: ");

    var is_email_correct = user_email.match(/\S+@\S+\.\S+/);

    if(is_email_correct){

        alert("email корректный.");

    }else{

        alert("email не корректный!");
    }

</script>
```

Проверка данных

Логические операторы

&&	||	!

```
<script>
    //...

    if( (user_name == "ivan") && (user_password == "12345")){
        //...
    }

    //...

    if( (user_height > 145) || (user_age > 12) ){
        //...
    }
    //..

    var is_blocked = true;

    if( !is_blocked ){
        //...
    }
</script>
```

Когда нужны «сложные» условия

Логические операторы

&&	||	!

&&	False	True
False	False	False
True	False	True

||	False	True
False	False	True
True	True	True

!	False	True
	True	False

Таблицы истинности

Операторы логического И (&&) и логическое ИЛИ (||) работают по такой схеме:

1) *Приводят левый операнд к boolean;*
2) *Если по нему можно сделать выводы, то возвращают левый операнд (в том типе в котором он и был);*
3) *Если нет, то возвращают правый операнд (в том типе в котором он и был).*

Логические операторы

&&	||	!

```
<script>

    var a = 5;
    var b = 5;

    if( (a++ > 10) && (b++ > 10) ){
        //...
    }

    console.log(a, b);

</script>
```

?!?

Есть нюансы

Логические операторы

&&	||	!

```
1  <script>
2
3      var a = 5;
4      var b = 5;
5
6      if( (a++ > 0) || (b++ > 0) ){
7          //...
8      }
9
10     console.log(a, b);
11
12 </script>
```

?!?

Есть нюансы

Логические операторы

&&	||	!

Логические операторы && и || могут не проверять правый операнд, если значение левого операнда уже достаточно для итогового результата выражения.

Есть нюансы

Логические операторы

&&	||	!

```
1  <script>
2
3      var a = 5 && 7;
4      var b = 0 && 4;
5      var c = 8 || 1;
6
7      console.log(a, b, c);
8
9  </script>
```

?!?

Есть нюансы

Побитовые операторы

&	|	^	~	<<	>>	>>>

```
1  <script>
2
3      var a = 21;
4      var b = 10;
5
6      var logic   = a && b;
7      var bits    = a & b;
8
9      console.log( logic, bits );
10
11 </script>
```

?!?

Не путайте логически и побитовые операторы, их результат далеко не всегда совпадает

Побитовые операторы

&	|	^	~	<<	>>	>>>

```
<script>

    var a = 21;
    var b = 10;

    var logic   = a && b;
    var bits    = a & b;

    console.log( logic, bits );

</script>
```

?!?

Не путайте логически и побитовые операторы, их результат далеко не всегда совпадает

Побитовые операторы

&	|	^	~	<<	>>	>>>

&

21	**=>**	0	0	0	1	0	1	0	1
10	**=>**	0	0	0	0	1	0	1	0
0	**<=**	0	0	0	0	0	0	0	0

|

21	**=>**	0	0	0	1	0	1	0	1
10	**=>**	0	0	0	0	1	0	1	0
31	**<=**	0	0	0	1	1	1	1	1

Побитовые операторы осуществляют действия с числами в двоичном виде

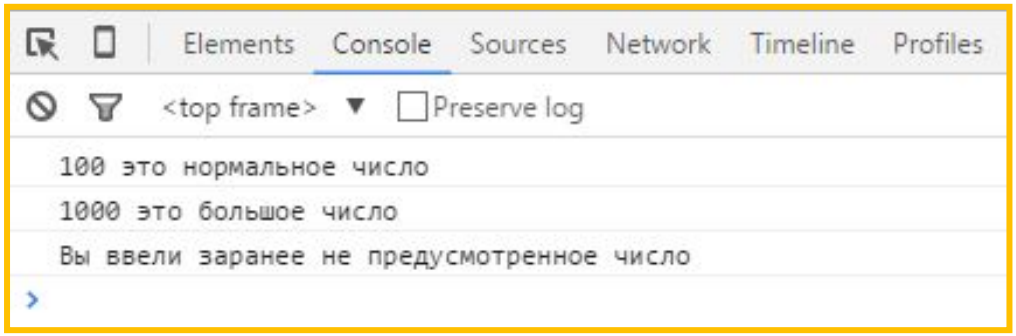
Оператор выбора switch

```
<script>

    var a = Number(prompt("Введите число"));

    switch (a) {
        case 1:
            console.log("1 это маленькое число");
            break;
        case 100:
            console.log("100 это нормальное число");
            break;
        case 1000:
            console.log("1000 это большое число");
            break;
        default:
            console.log("Вы ввели заранее не предусмотренное число");
            break;
}

</script>
```

Позволяет выбрать из множества вариантов, работает только сравнением

Оператор выбора switch

```
<script>

    var a = Number(prompt("Введите число"));

    switch (a) {
        case 1:
            console.log("1 это маленькое число");
        case 100:
            console.log("100 это нормальное число");
        case 1000:
            console.log("1000 это большое число");
        default:
            console.log("Вы ввели заранее не предусмотренное число");
}

</script>
```



И тут есть нюансы…

Многовариантный выбор

```
<script>

    var a = Number(prompt("Введите число"));

    if(a == 1){
        console.log("1 это маленькое число");
    }else if(a == 100){
        console.log("100 это нормальное число");
    }else if(a == 1000){
        console.log("1000 это большое число");
    }else{
        console.log("Вы ввели заранее не предусмотренное число");
    }
}
</script>
```

Можно обойтись и без switch

Выбор в диапазоне

```
<script>

    var a = Number(prompt("Введите температуру воздуха"));

    if(a < 15){
        console.log("Холодно...");
    }else if(a < 30){
        console.log("Комфортно...");
    }else{
        console.log("Жарко...");
    }

</script>
```

Пошаговое отсекание вариантов

Немного практики

***Задача**: Посетитель вводит год, и необходимо определить сколько дней в году, и сообщить ответ посетителю.*

Задача о високосных годах…

Крат. 4	Крат. 100	Крат. 400	Результат
Нет	Нет	Нет	**Нет**
Нет	Нет	Да	**Нет**
Нет	Да	Нет	**Нет**
Нет	Да	Да	**Нет**
Да	Нет	Нет	**Да**
Да	Нет	Да	**Нет**
Да	Да	Нет	**Нет**
Да	Да	Да	**Да**

Таблица истинности для 3-х логических переменных.

Немного практики

***Задача**: Сегодня вторник, пользователь задаёт количество дней. Необходимо узнать какой день недели будет по прошествии количества дней которые задал пользователь.*

Немного практики

***Задача**: Мы знаем день, месяц и год рождения человека. Мы также знаем сегодняшний день месяц и год, необходимо узнать сколько человеку полных лет.*

Домашнее задание

Домашнее задание

1. Познакомиться с тернарным оператором (он же оператор ***?:*** *).*

2. Разработать скрипт, который на основе роста и веса пользователя выводит информацию о том, нормальный ли вес у пользователя или выше нормы или ниже нормы. За норму, для расчётов, взять градации «индекса массы тела» (по википедии).

1. *Недостаток информации;*
2. *Избыточная информация;*
3. *Косвенно влияющие аспекты.*

Домашнее задание

О налогах в США

http://www.tax-rates.org/federalincometax

Single	Married (Jointly)	Married (Separately)	Head of Household	Marginal Tax Rate
$0+	$0+	$0+	$0+	**10.00%**
$9,325+	$18,650+	$9,325+	$13,350+	**15.00%**
$37,950+	$75,900+	$37,950+	$50,800+	**25.00%**
$91,900+	$153,100+	$76,550+	$131,200+	**28.00%**
$191,650+	$233,350+	$116,675+	$212,500+	**33.00%**
$416,700+	$416,700+	$208,350+	$416,700+	**35.00%**
$418,400+	$470,700+	$235,350+	$444,500+	**39.60%**

Задание: *написать скрипт, который будет запрашивать сумму годового дохода человека, и рассчитывать сумму федерального налога (для США), которую человек должен оплатить (за основу взять ставки для лиц не состоящих в браке – single).*

Где брать задачки для тренировки?

Чтобы научиться программировать – нужно тренироваться…
Чтобы тренироваться нужны задачи…
Чтобы были задачи нужно уметь программировать…

Д. М. Златопольский
Сборник задач по программированию

Prometheus CS50

Если вы хотите научится программированию – это лучшее что может предложить интернет.

http://courses.prometheus.org.ua/courses/Prometheus/CS50/2016_T1/about