

# «Ветвление, логические операции»



JavaScript  
Courses

[www.courses.dp.ua](http://www.courses.dp.ua)

# JavaScript как язык программирования

*его*

*концепции*

**Переменные / Типы /**

**Операции**

**Ветвления (условные**

**операторы)**

**Циклы / Массивы (структуры**

**данных)**

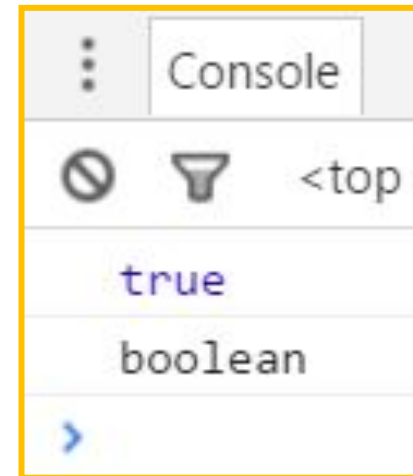
**Функци**

**Объект**

**ы**

# Тип Boolean

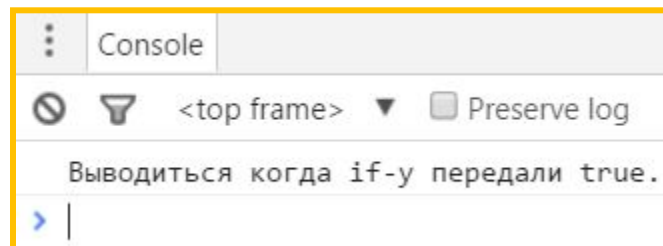
```
1 <script>  
2  
3   var a = true;  
4  
5   var b = false;  
6  
7   console.log(a);  
8  
9   console.log(typeof(a));  
10  
11 </script>
```



*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

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

*Блок else – является необязательным.*

# Откуда берётся boolean?

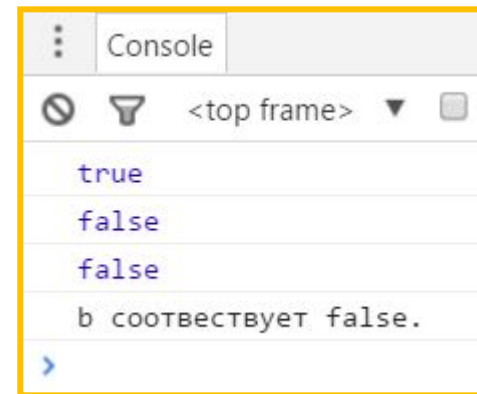
*undefined, NaN => false;*

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

*String: "" => false; все остальные =>*

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

```
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 соответствует true.");
13  }else{
14      console.log("b соответствует false.");
15  }
16
17 </script>
```

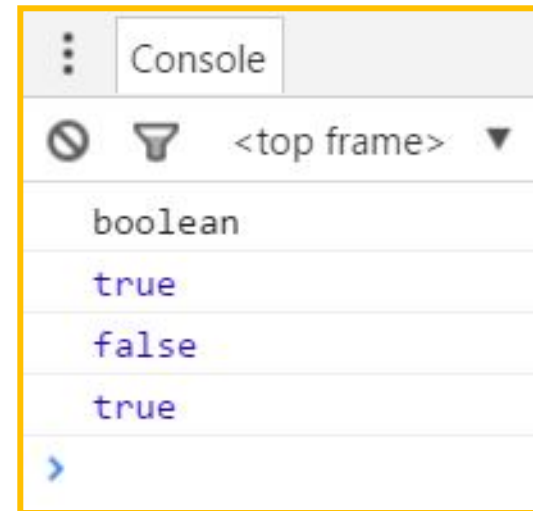


*Из преобразование типов.*

# 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>
```





# Откуда берётся boolean?



```
1 <script>
2
3   var x = 6;
4
5   var y = 8;
6
7   var bool_var = x > y;
8
9   if(bool_var){
10      //..
11   }else{
12      //...
13   }
14
15 </script>
```



```
1 <script>
2
3   var x = 6;
4
5   var y = 8;
6
7   if(x > y){
8      //..
9   }else{
10      //...
11   }
12
13 </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>
```

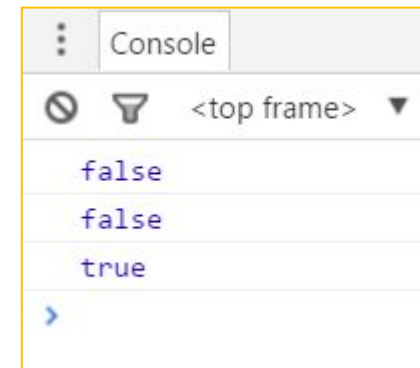


# Откуда берётся boolean?



*Сравнение строк осуществляется посимвольно.  
Сравниваются 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>
```

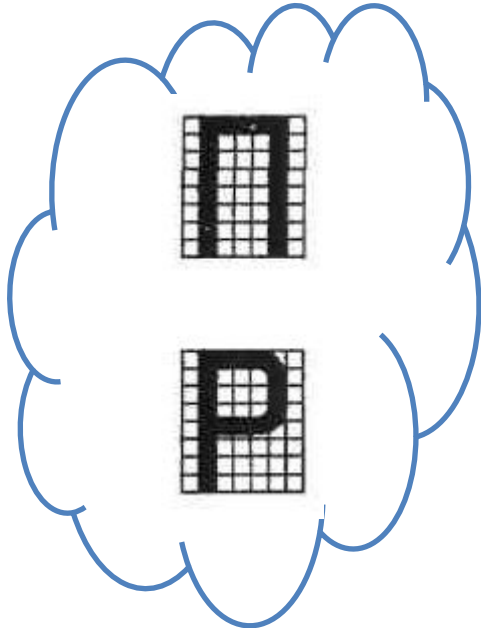


# Кодировка

	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
0		0	@	P	`	р	Ъ	ђ	NBSP	°	А	Р	а	р
1	!	1	A	Q	a	q	Ґ	‘	Ў	±	Б	С	б	с
2	"	2	B	R	b	r	,	’	ў	І	В	Т	в	т
3	#	3	C	S	c	s	ѓ	“	Ј	і	Г	У	г	у
4	\$	4	D	T	d	t	„	”	Ѡ	г	Д	Ф	д	ф
5	%	5	E	U	e	u	…	•	Г	µ	Е	Х	е	х
6	&	6	F	V	f	v	†	-	і	¶	Ж	Ц	ж	ц
7	'	7	G	W	g	w	‡	-	§	·	З	Ч	з	ч
8	(	8	H	X	h	x	¡	¡	Ё	ё	И	Ш	и	ш
9	)	9	I	Y	i	y	%	™	©	№	Й	Щ	й	щ
A	*	:	J	Z	j	z	Љ	љ	Є	е	К	Ъ	к	ъ
B	+	;	K	[	k	{	<	>	«	»	Л	Ы	л	ы
C	,	<	L	\	l		НЬ	нь	-	і	М	Ь	м	ь
D	-	=	M	]	m	}	Ќ	ќ	Š	Š	Н	Э	н	э
E	.	>	N	^	n	~	Ђ	ђ	®	š	О	Ю	о	ю
F	/	?	O	_	o		Џ	џ	Ї	ї	П	Я	п	я

```

Файл  П_р_а_в_к_а  В_и_д  К_о_д_и_р_о_в_к_а  С_п_р_а_в_к_а
00000000: CF F0 E8 E2 E5 F2 21 21 21
    
```



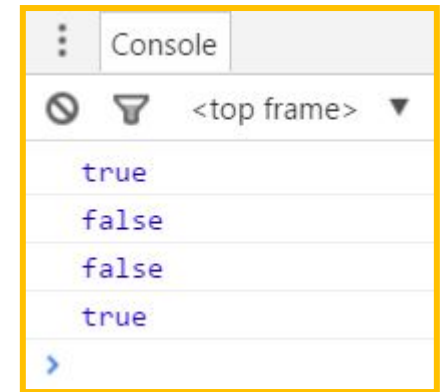
12

П 207	р 240	и 232	в 226	е 229	т 242	! 33	! 33	! 33
----------	----------	----------	----------	----------	----------	---------	---------	---------

# Откуда берётся boolean?



```
1 <script>
2
3   var a = "5";
4   var b = 5;
5
6   console.log(a == b);
7
8   console.log(a === b);
9
10  console.log((typeof(a) == typeof(b)) && (a == b));
11
12  console.log(typeof(a) == "string");
13
14 </script>
```



*Сравнение с учетом типа*

# Откуда берётся boolean?

```
1 <script>
2
3   var user_email = prompt("Enter email: ");
4
5   var is_email_correct = user_email.match(/\\S+@\\S+\\.\\S+/);
6
7   if(is_email_correct){
8
9       alert("email корректный.");
10
11   }else{
12
13       alert("email не корректный!");
14   }
15
16 </script>
```

*Проверка данных*

# Логические операторы

&&

||

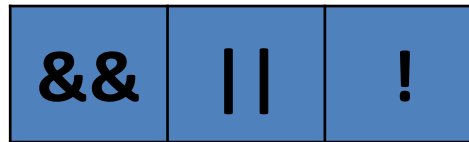
!

```
1 <script>
2 //...
3
4 if( (user_name == "ivan") && (user_password == "12345")){
5     //...
6 }
7
8 //...
9
10 if( (user_height > 145) || (user_age > 12) ){
11     //...
12 }
13 //..
14
15 var is_blocked = true;
16
17 if( !is_blocked ){
18     //...
19 }
20 </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
False	True	False
True	False	True

## Таблицы

Операторы логического И (**&&**) и логического 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



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

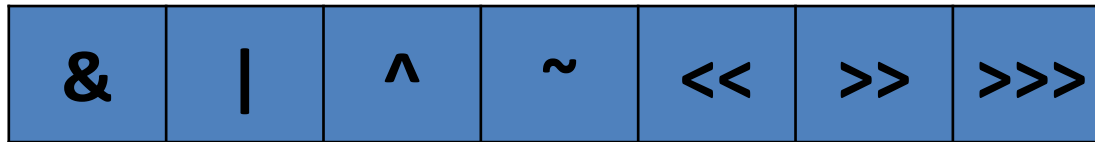
&&   ||   !

```
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

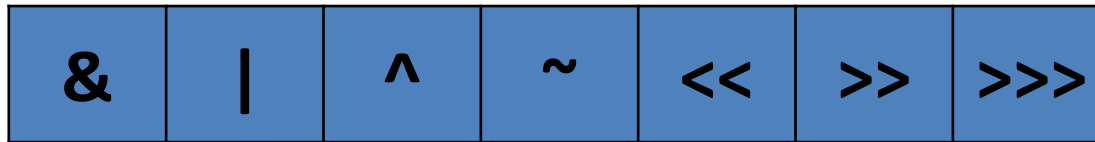


```
1 <script>
2
3   var a = 21;
4   var b = 10;
5
6   var logic    = a && b;
7   var bits     = a & b;
8
9   console.log( logic, bits );
10
11 </script>
```



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

# Побитовые операторы

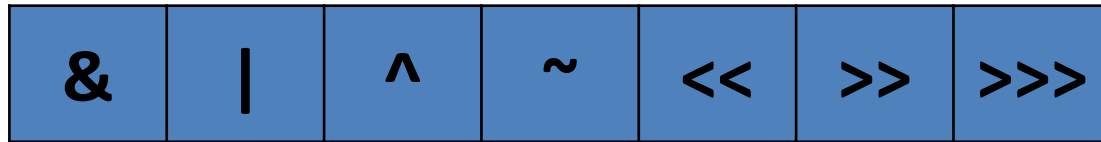


```
1 <script>
2
3   var a = 21;
4   var b = 10;
5
6   var logic    = a && b;
7   var bits     = a & b;
8
9   console.log( logic, bits );
10
11 </script>
```



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

# Побитовые операторы



**&**

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

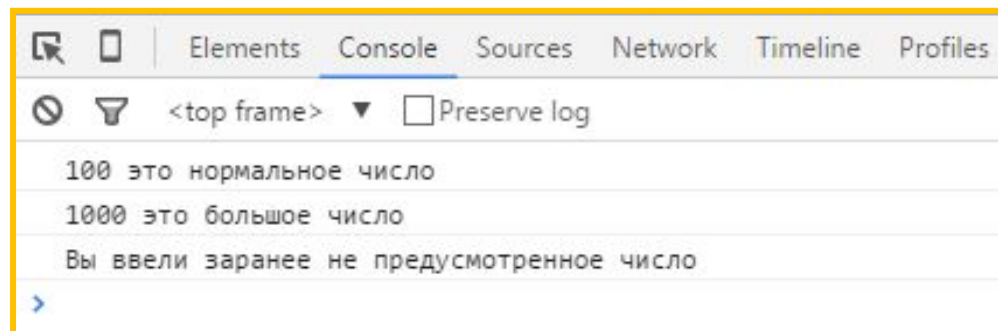
```
1 <script>
2
3   var a = Number(prompt("Введите число"));
4
5   switch (a) {
6       case 1:
7           console.log("1 это маленькое число");
8           break;
9       case 100:
10            console.log("100 это нормальное число");
11            break;
12       case 1000:
13            console.log("1000 это большое число");
14            break;
15       default:
16            console.log("Вы ввели заранее не предусмотренное число");
17            break;
18   }
19
20 </script>
```

*Позволяет выбрать из множества вариантов, работает только сравнением*



# Оператор выбора switch

```
1 <script>
2
3   var a = Number(prompt("Введите число"));
4
5   switch (a) {
6     case 1:
7       console.log("1 это маленькое число");
8     case 100:
9       console.log("100 это нормальное число");
10    case 1000:
11      console.log("1000 это большое число");
12    default:
13      console.log("Вы ввели заранее не предусмотренное число");
14  }
15
16 </script>
```



*И тут есть нюансы...*

# Многовариантный выбор

```
1 <script>
2
3   var a = Number(prompt("Введите число"));
4
5   if(a == 1){
6       console.log("1 это маленькое число");
7   }else if(a == 100){
8       console.log("100 это нормальное число");
9   }else if(a == 1000){
10      console.log("1000 это большое число");
11   }else{
12      console.log("Вы ввели заранее не предусмотренное число");
13   }
14 }
15 </script>
```

*Можно обойтись и без switch*

# Выбор в диапазоне

```
1 <script>
2
3   var a = Number(prompt("Введите температуру воздуха"));
4
5   if(a < 15){
6       console.log("Холодно...");
7   }else if(a < 30){
8       console.log("Комфортно...");
9   }else{
10      console.log("Жарко...");
11  }
12
13 </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	<b>Нет</b>
Нет	Нет	Да	<b>Нет</b>
Нет	Да	Нет	<b>Нет</b>
Нет	Да	Да	<b>Нет</b>
Да	Нет	Нет	<b>Да</b>
Да	Нет	Да	<b>Нет</b>
Да	Да	Нет	<b>Нет</b>
Да	Да	Да	<b>Да</b>

*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%

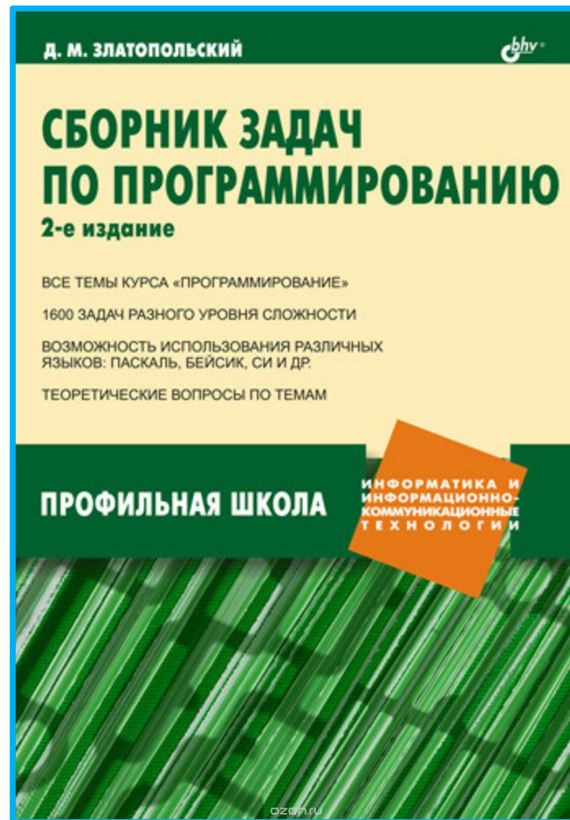
**Задание:** написать скрипт, который будет запрашивать сумму годового дохода человека, и рассчитывать сумму федерального налога (для США), которую человек должен оплатить (за основу взять ставки для лиц не состоящих в браке – *single*).

# Где брать задачи 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 научиться программированию – это лучшее что может предложить интернет.*

[http://courses.prometheus.org.ua/courses/Prometheus/CS50/2016\\_T1/about](http://courses.prometheus.org.ua/courses/Prometheus/CS50/2016_T1/about)