

Добро пожаловать
на 2 занятие
по JS!

Лирическое отступление про ДЗ





Числа в JS

- Все числа в JavaScript, как **целые** так и **дробные**, имеют тип Number и хранятся в 64-битном формате IEEE-754, также известном как «double precision».
- $0.7+0.1= 0.7999999999999999$
- $0.2+0.1= 0.30000000000000004$
- $0xFF = 255$ (в шеснадчатеричной системе)
- $3e5 = 300000$ (в научном формате)

Работа с числами

- `parseInt(string[, radix=10])` – преобразует строку в число нужной системы счисления
- `string` - строковое представление числа
- `radix` - основание системы счисления

`parseInt("17", 8) = 15`

- `IntValue.toString(Основание)` – преобразует число в строку в нужной системе счисления

`t=5;`

`t.toString(2)="101" // Можно 5.toString(2)`

Инкремент и декремент

- **Postfix** `a - - / a + +` (без пробела)

```
var a=5;
```

```
alert(a++); //5 Сначала выводит, потом ++
```

```
alert(a) //6
```

- **Prefix** `-- a / ++ a` (без пробела)

```
var b=5;
```

```
y=--b; //4 Сначала --, потом присвоение
```

```
alert('b='+b+',y='+y) //4
```

Строки в JS

- Создание/присвоение

```
var text = "моя строка"; //
```

- `text.length` // 10
- `text.charAt(2)` // 'я' – 3-ий символ
- `text[2]` - ? //
- `text.toUpperCase()` //МОЯ СТРОКА
- `text.substring(2)` //'я строка' – с 3-его до конца
- `text.substring(2,5)` //'я с'– с третьего до 5-го(не вкл.)

Спец Символы

- Специальные символы

```
alert( 'Привет\nМир' ); // выведет "Мир" на новой строке
```

Символ	Описание
<code>\b</code>	Backspace
<code>\f</code>	Form feed
<code>\n</code>	New line
<code>\r</code>	Carriage return
<code>\t</code>	Tab

- Экранирование специальных символов

```
var str = '\\m a JavaScript "programmer";  
var str = "I'm a JavaScript \"programmer\" ";  
alert(str); // I'm a JavaScript "programmer"
```


Дополнительные возможности

- Вывод

`alert(x)` – вывод сообщения

`console.log(x)` – вывод в консоль браузера

`document.write("x="+x)` – вывод в текущую страницу с заменой содержимого

`document.getElementById('myedit').innerHTML=`

`'Пишу, что хочу!'` – вывод внутрь тега с id `'myedit'`

If – Игра угадай число!

```
max=10;
min=0
mynumber=Math.floor(Math.random() *
(max - min + 1)) + min;
var answer = prompt('Я загадал число от '+min+' до
'+max+"\n Какое число я загадал?");
if (answer < mynumber) {
  alert('Больше...');
} else if (answer > mynumber) {
  alert('Меньше');
} else {
  alert('Именно!');
}
console.log('Загадано число '+mynumber)
```

prompt и его особенности

- Ввод значений

```
x = prompt("Введите число\n");
```

!! Prompt не производит приведение типов и переменная *x* **ВСЕГДА** будет **СТРОКОЙ**

Как бороться?

```
x="5";
```

```
x=parseInt(x) – Вернет число // x=5;
```

Победа!

Тернарный оператор

- Проверка авторизации (упрощенная)

```
console.log(loged = (typeof(login) !== 'undefined')  
? 'Auth ok' : 'Need auth' );
```

SWITCH – вместо тысячи if

```
var a=prompt('Введите целое число от 1 до 5')
a=parseInt(a);
switch(a){
case 1: console.log(a+'<=1')
case 2: console.log(a+'<=2')
case 3: console.log(a+'<=3')
case 4: console.log(a+'<=4')
case 5: console.log(a+'<=5')
break;
default: console.log('Нужно ввести целое число от 1 до
5!\n Вы ввели '+a)
}
```


SWITCH – вместо тысячи if

```
var a=prompt('Введите целое число от 1 до 5')
a=parseInt(a);
switch(a){
case 1: console.log(a+'<=1')
case 2: console.log(a+'<=2')
case 3: console.log(a+'<=3')
case 4: console.log(a+'<=4')
case 5: console.log(a+'<=5')
break;
//вариант default, если ни один другой не подошел
default: console.log('Нужно ввести целое число от 1 до
5!\n Вы ввели '+a)
}
```

SWITCH – важная мелочь!

```
var a=prompt('Введите целое число от 1 до 5')
a=parseInt(a);
switch(a){
case 1: console.log('Вы ввели число 1'); break;
case 2: console.log('Вы ввели число 2'); break;
case 3: console.log('Вы ввели число 3'); break;
case 4: console.log('Вы ввели число 4'); break;
case 5: console.log('Вы ввели число 5'); break;
break;
default: console.log('Нужно ввести целое число от 1 до
5!\n Вы ввели '+a)
}
```

Мой первый цикл!

- Конечно же это цикл **while**

```
while(true) {  
[Do something]  
}
```

Пример использования:

```
var a=0;  
while (a!=new Date().getFullYear()){  
a=prompt('Введите номер текущего года')  
}  
alert('верно');
```

Управление Break/Continue

```
b=0;
while(true)
{
  b++;
  if (b%2==0) {
    console.log("Четное!");
    continue; //
  }
  console.log(b);
  if(b>10) { //Как изменится цикл при if(b>=10) ?
    break;}
}
```



Break multipleLoops (*)

```
b=0;
a=1;
breakthis: while (a==1)
{
  while(true)
  {
    b++;
    console.log(b);
    if(b>=5) {
      a=2;
      break breakthis;}
  }
}
```


Самое время вернуться к истокам

И дописать игру угадай число!

- Угадывание должно идти до правильного ответа.
- Число между попытками не меняется
- Отобразить с какой попытки угадано число

Задачи 1

- Используя цикл `while` и введенную с клавиатуры строку отобразить её наоборот.
(Привет =>тевирП)
- ** Перевести число, введенное с клавиатуры в систему счисления по основанию 3, отобразить цифры наоборот и перевести обратно в десятичную систему из троичной.
'29' => 1002 => 2001 => '55' //Проверка
- Посчитать сумму чисел от 1 до «введенного с клавиатуры неотрицательного числа».

Задачи 2

- Написать Switch, который принимает на вход значение от 1 до 7 и отображает день недели. Если выбраны выходные (if) написать "Ура!"
1 – "Понедельник"
6 – "Ура! Суббота"
- Написать меню игры, в котором принимаются команды: 'start' , 'options' и 'exit'. После ввода команды выполняется действие команды и ожидается следующая (кроме exit).

При 'start' выведите "начало", при 'options' => "опции", при 'exit' => "выход". При вводе других команд напишите – "команда не распознана"

Задачи 3

- Используя форматирование вывода вывести числа от [начального] (ввод с клавиатуры) до [конечного] в виде [n] колонок.

Пример для 1 15 5

```
1  2  3  4  5
6  7  8  9 10
11 12 13 14 15
```

Пример для 1 15 3

```
1  2  3
4  5  6
7  8  9
10 11 12
13 14 15
```

** Учесь длину числа, используемую при выводе

Д/З

- ДЗ в идеале сдается до 23:59 даты следующего занятия.
- Самый поздний срок сдачи первой версии – через 8 дней после занятия.
- Собираете скрипты задания в папку с номером занятия
- Готовое задание заливаете на GitHub и пишете на почту или vk.com (Со ссылкой на репозиторий)
- Почта hiddenfun@mail.ru vk.com/hiddenfun

ДЗ зачем сдавать вовремя?

- Легче понимать материал;
- Сданные задания дают бонусы при защите (освобождают от части практических задач);
- Вы пришли сюда программировать, а программирование – это *практический* навык
- Бонусы:
- Сданы все задания:
- На итоговой защите решаете *одно* задание
- Сданы половина-большинство заданий - решаете *одно или два* задания (по покрытиям тем)
- Сдано меньше половины заданий *2 или 3* задания + вопросы по их выполнению

Сдача JS

- Сдача работы (приложения/игры) на js с html страницей;

Командная работа поощряется

- Индивидуальное решение набора задач по материалам курса (влияет бонус от ДЗ).

Следующее занятие
суббота

11:00