Основы JS(4)

Объекты JS

- Структура в виде ассоциативного массива произвольной сложности

- Задается одним из способов:
- var train= new Object();//Конструктор
- var train = { } // С помощью квадратных скобок

Установка свойств

Аналогична работе с ассоциативными массивами:

- //Стиль свойства

```
train.speed=60;
```

- //Стиль массива

```
train['coal']=20;
```

Присвоение свойств(значения)

- //Стиль свойства

```
var speed=train.speed;
```

//Стиль массива

```
var coalleft=train['coal'];
```

- Пример:

```
speed=train.speed // 60
train.speed=20; // 20
speed //60
```

Начальная инициализация (присвоение) свойств

```
var train = {
speed : 60 ,
 coal : 20
}
```

Усложненная структура

```
var cat = {
    name : 'Барсик',
    details: {
        age: 3,
        color: 'brown',
        owners : ['Даша','Вася']
        }
}
```

Обращение к свойствам

```
console.log(cat.name) //Имя
console.log(cat.details.age) //Возраст из details
console.log(cat.details.owners[1]) //Второй хозяин
```

- У объекта отсутствует длина, поэтому обход свойств как для ассоциативных массивов:

```
for (key in cat)
  {console.log('arr['+key+']='+cat[key]);}
```

Функции!

Объявление функции

```
function sayhello(){
alert('Привет из функции');//Блок из команд
}
```

- Вызов

```
sayhello();// Первый вызов вернет сообщение
sayhello();// Повторный вызов
```

Переменные внутри функции

```
function sayhello(){
var name='Иван'; // переменная локальна
alert('Привет, '+ name);
}
sayhello(); //Вернет "Привет, Иван"
alert(name);// ?
```

Переменные внутри функции

```
function sayhello(){
var name='Иван'; // переменная локальна
alert('Привет, '+ name);
}
sayhello(); //Вернет "Привет, Иван"
alert(name);// ?
//ReferenceError: name is not defined
```

Функции и внешние переменные

```
var name='Иван'
function sayhello(){
alert('Привет, '+ name);
name='Александр';
}
sayhello();
alert(name);// "Александр"
```

Функции (глобал+локал var)

```
var name='Иван'
function sayhello(){
var name='Петя'; // переменная локальна
alert('Привет, '+ name);// Привет, Петя
name='Александр';
}
sayhello();
alert(name);// "Иван"
```

Параметры функции

```
function sayhello(say , toname){//Через запятую, сколько угодно
alert(say+', '+ toname);//
}
sayhello('Привет',"море"); //Привет, море
sayhello('Здравствуй',"лето"); //Здравствуй, лето

sayhello("Привет", "море", "лето") //?
```

Возврат значений

```
function calc(a,b){
return a+b;//Вернет значение a+b
}
alert(calc(2,4));
```

Если отсутствует, то вернет undefined

Несколько return

```
function calc(a,b){
return a+b;//Вернет значение a+b
return a*2; // Ничего не вернет, т.к. return это
   аналог break для функции
}
alert(calc(2,4));
```

Если отсутствует, то вернет undefined

Несколько return 2

```
function calc(a,b,action){
    if (action=='sum'){
        return a+b;//Вернет значение a+b
    }
    if (action=='mul'){
        return a*b; // Вернет a*b
    }
}
alert(calc(2,4,'sum')); // 6
alert(calc(2,4,'mul')); //8
```

Псевдомассив "arguments"

```
function calc(a,b){ // 2 обязательных
    sum=a+b;//Вернет значение a+b
    for (var i = 2; i < arguments.length; i++) {
        sum+=arguments[i];// += короткий
          синтаксис аналогичен "sum=sum+"
    }
    return sum;
}
calc(1) // NaN      calc(1,2,3,4,5,6) //18
calc(1,2) // 3      calc(0,-1,-3) // -4
```

Псевдомассив "arguments"

```
function calc(){
var sum=0;
    for (var i = 0; i < arguments.length; i++) {
        sum+=arguments[i];// += короткий
          синтаксис аналогичен "sum=sum+"
    }
    return sum;
}
```

- calc(1) // 1 calc(1,2,3,4,5,6) //18
- calc(1,2) // 3 calc(0,-1,-3) // -4

Псевдомассив ВАЖНО!

```
function calc(a,b){ // 2 обязательных
    console.log(arguments[0]+','+a);
    a=a+1;
    console.log(arguments[0]+','+a);
    arguments[0]--;
    console.log(arguments[0]+','+a);
}
```

- calc(2,5) //В режиме strict это разные значения!

Перегрузка функций

```
function calc(a,b){
return (a+b)
}
function calc(a,b){
return (a*b)
}
calc(2,4);
```

Функции, как методы объекта

```
cat = {}
cat.hunger=100;
cat.feed = function () {
cat.hunger-=20;
alert('Myppp....')
}
console.log(cat.hunger); // 100
cat.feed(); //Myppp....
console.log(cat.hunger); // 80
```

Рекурсия

Рекурсия – это вызов функции внутри самой себя.

Простой пример:

Сумма чисел от 1 до N

```
function sum(n){
if (n==1) {return 1;}
return n+sum(n-1);//Запустит себя же
}
```

- sum(10) // 55

Вот так мы плавно подошли к ООП

Задачи

1. Создать объекта боец, у которого есть имя, фамилия и прочие атрибуты, а также атрибут "рюкзак", в который нужно положить всё необходимое (Определить атрибуты рюкзака). Положить в рюкзак удочку, наживку, леску, грузила и крючки.

1*. + предметы в рюкзаке имеют вес

+ функция/метод, выводящая этот вес

+ метод положить/выбросить в рюкзак

+ количество для одинаковых предметов

Стой, стрелять буду!

2. Функция проверки авторизации по паре логин-пароль, которая выводит сообщения:

Здравствуй, [login]!

Неверная пара логин-пароль

логин-пароль должен хранится в объекте "security" с полями login и password.

2 * написать функцию регистрации пользователей (+имя, фамилия, о себе и т.д.)

Админ может смотреть список пользователей и пары логин-пароль

Игра морской бой

- 3. Создать объект поля морской бой (10*10)

и заполнить его кораблями

Написать функцию, которая будет по координатим (x,y) проверять попадание в цель

3 * + Добавить отображение полей боя для двух игроков + изменение поля при "попал"

+ Вода синего цвета, корабли зеленого.

+ После попадания в корабль цвет меняется на красный(клетки).Утопленный –> черный.

Смартфон

- 4. Создать объект "Сматфон" у которого заданы поля: имя, память(hdd), память(ram) и список приложений (имя, hdd, ram)

Добавить список запущенных приложений.

Посчитать потребление памяти (hdd, ram)

4. * Возможность устанавливать НОВЫЕ приложения из объекта market (Если хватает памяти+устройство поддерживается[список])

Запуск и остановка приложений

Удаление приложений и очистка памяти

Рекурсия

- 5. Посчитать значения факториала числа при помощи функции рекурсивно

N! = N*(N-1)! , 1! = 1

- Цифры числа записать справа-налево

//153 => 351

- Определить является ли указанное число степенью заданного числа

12 , 2 NO

16 , 2 YES

Рекурсия 2

* Разложить число на простые множители в порядке возрастания и с учетом кратности

* J: Палиндром

Дано слово, состоящее только из строчных латинских букв. Проверьте, является ли это слово палиндромом. Выведите YES или NO.

radar – yes

Yes - no