Прототипы. Наследование.

Прототипы

Каждая функция создается со свойством prototype - объектом, содержащим свойства и методы, которые должны быть доступны в экземплярах конкретного ссылочного типа. Этот объект в буквальном смысле является прототипом для объекта, создаваемого при вызове конструктора.

Преимущество использования прототипа в том, что все его свойства и методы общие для объектов. Вместо того чтобы назначать атрибуты объекту в конструкторе, их можно назначить непосредственно прототипу

Прототипы

Когда мы хотим прочитать свойство из object, а оно отсутствует, JavaScript автоматически берёт его из прототипа. В программировании такой механизм называется «прототипным наследованием».

Свойство [[Prototype]] является внутренним и скрытым, но есть много способов задать его.

Прототипы

```
let animal = {

  eats: true

};

let rabbit = {

  jumps: true

};

rabbit.__proto__ = animal;
```

Прототипы

Цикл for..in проходит не только по собственным, но и по унаследованным свойствам объекта.

```
// Object.keys возвращает только собственные ключи
alert(Object.keys(rabbit)); // jumps

// for..in проходит и по своим, и по унаследованным ключам
      for(let prop in rabbit) alert(prop); // jumps, затем eats
```

Прототипы

Если унаследованные свойства нам не нужны, то мы можем отфильтровать их при помощи встроенного метода obj.hasOwnProperty(key): он возвращает true, если у obj есть собственное, не унаследованное, свойство с именем key.

Прототипы

новые объекты могут быть созданы с помощью функции-конструктора new F().

Если в F.prototype содержится объект, оператор new устанавливает его в качестве [[Prototype]] для нового объекта.

Прототипы

```
let animal = {
  eats: true
};

function Rabbit(name) {
  this.name = name;
}

Rabbit.prototype = animal;

let rabbit = new Rabbit("White Rabbit"); //
rabbit.__proto__ == animal

alert( rabbit.eats ); // true
```

Установка `Rabbit.prototype = animal` буквально говорит интерпретатору следующее: "При создании объекта через `new Rabbit()` запиши ему `animal` в `[[Prototype]]`".

Альтернативный синтаксис прототипов

```
function Person(){ }

Person . prototype = {

        name : "Nicholas",

        age : 29,

        job : "Software Engineer" , sayName : function ( ) {

        alert (this . name ) ;

} };
```

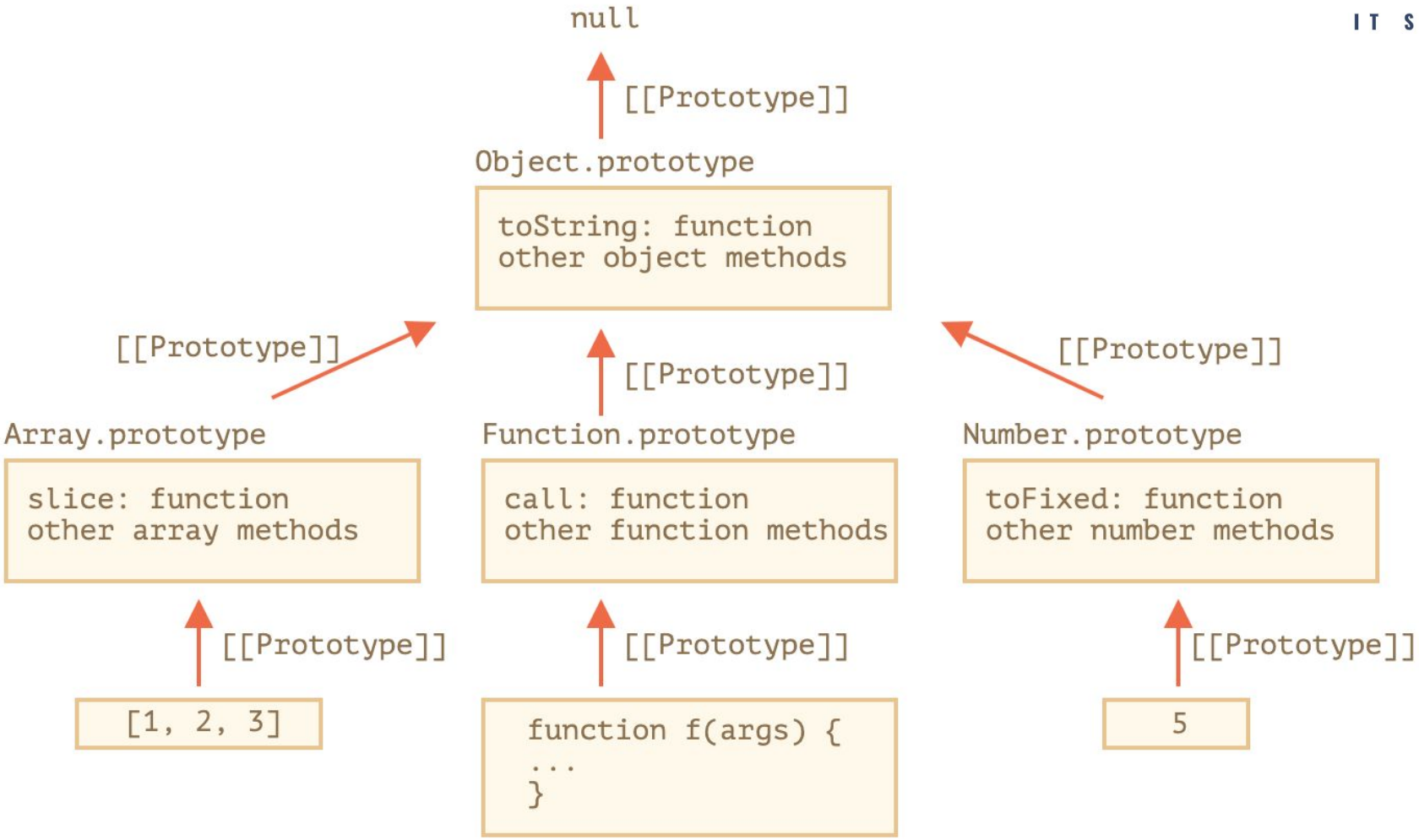
Прототипы

С помощью прототипов встроенных объектов можно получать ссылки на методы, предлагаемые по умолчанию, и определять новые методы. Кроме того, эти прототипы можно изменять, что позволяет добавлять методы к встроенным объектам.

Прототипы

obj = new Object(), где Object – встроенная функция-конструктор для объектов с собственным свойством prototype, которое ссылается на огромный объект с методом toString и другими.

Другие встроенные объекты, такие как Array, Date, Function и другие, также хранят свои методы в прототипах.


null
[[Prototype]]
Object.prototype
toString: function
other object methods
[[Prototype]]
[[Prototype]]
[[Prototype]]
Array.prototype
slice: function
other array methods
Function.prototype
call: function
other function methods
Number.prototype
toFixed: function
other number methods
[[Prototype]]
[[Prototype]]
[[Prototype]]
[1, 2, 3]
function f(args) {
...
}
5

Прототипы

```
function Animal() {
    this.run = function () {
        alert('run');
    }
}

function Rabbit() {
    this.jump = function () {
        alert('jump');
    }
}

Rabbit.prototype = new  Animal();
let rabbit = new Rabbit();

rabbit.run(); // run
```

Методы прототипов

- Object.getPrototypeOf(obj) – возвращает свойство [[Prototype]] объекта obj.
- Object.setPrototypeOf(obj, proto) – устанавливает свойство [[Prototype]] объекта obj как proto.

Методы прототипов

- alert(Object.getPrototypeOf(rabbit) === animal);
- Object.setPrototypeOf(rabbit, {});

Задача 1

Создать функцию конструктор Dog, который наследует метод eat из объекта Animal

IT SHATLE

Задача 2

Реализовать на основе прототипного наследования создание геометрических фигур (например, базовая функция фигура, от которой наследуются функции рисования круга, квадрата, прямоугольника) - у фигур должны быть свойства цвета, размера, положения на странице, методы нарисовать фигуру, вычислить площадь фигуры

Задача 3

Реализовать на основе прототипного наследования создание модальных окон (например, базовая функция модальное окно, с методами показа и скрытия, от которого наследуются функции создания предупреждающего окна, запрещающего окна, окна с успешным выполнением)