ООП

JS не имеет многих средств для реализации объектоориентированного подхода как в классических языках ООП, таких как C++, C#, Java. Поэтому, во многом в JS эмитируется реализация парадигмы ООП. Так как JS прототипный язык программирования, мы имеем возможность расширять базовые типы данных собственными методами. Это один из походов в реализации объектного подхода

Расширение базовых типов данных JavaScript.

Эта программа, расширяет базовый тип данных String добавляя в него метод, который возвращает строку, состоящую из кодов тех символов из которых состояла строка, к которой применяли этот метод. Причём, так как мы расширяем базовый тип данных, то метод можно вызывать непосредственно у сроки через точку.

Код

```
( function(){
//проверка наличия метода с данным именем в нативном JavaScript
    if(typeof(String.prototype.code) !== 'function'){
    String.prototype.code  = function(){
    var beginStr = this;
    var str = '';
        for(var i= 0; i < beginStr.length-1; i++){
            str += beginStr[i].charCodeAt();

        }
        return str;
    }
} else {console.log('Имя данной фуекции реализованно в нативном JavaScript')}
})()
```

В этом коде для обеспечения безопасности, с помощью анонимного замыкания создаётся локальная область видимости. В самом начале проверяется, не реализован ли метод с таким же названием в нативном коде JS. Если метод с таким названием уже есть, об этом выводится сообщение и метод не реализуется. Затем алгоритм проходит по всем элементам строки и записывает их коды в отдельный массив, который и возвращается в качестве результата функции.

Вот небольшой пример использования данного подхода

Код

Как видно метод code применён непосредственно к строке.

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>String</title>
    <script src="./script.js"></script>
</head>
<body>
<h1>Введите текст</h1>
<div id="text" contenteditable style = 'width: 300px; height: 200px; background-color:#cccccc;  padding: 5px;
margin-bottom: 15px'></div>
<input type="button" value = 'Получить код' onclick = 'getCode()'>
<div id="code"></div>
<script>
 function getCode(){
    var inpText = '';
    var inpText = document.getElementById("text").innerHTML;
    document.getElementById('code').innerHTML = inpText.code();
 }
</script>
</body>
</html>
```

Так это выглядит в окне браузера

Данный метод несёт в себе потенциальную угрозу, если метод с таким же названием будет реализован в нативном JS, тем не менее существуют библиотеки реализующие подобный подход, например, Sugar