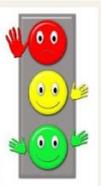


<b>Школа:</b>		
<b>Дата:</b>	<b>ФИО учителя:</b>	
<b>Класс:</b>	<b>Участвовали:</b>	<b>Не участвовали:</b>
<b>Тема урока:</b> Работа со строками, с массивами. Поиск и замена подстрок и.т.д. Преобразование типов		
<b>Цели обучения, которые достигаются на данном уроке</b>	В этом уроке познакомятся работой со строками, с массивами, поиском и замен подстрок языка Ruby	
<b>Цели урока</b>	<b>Все учащиеся смогут:</b> <ul style="list-style-type: none"> <li>Узнать работу со строками, с массивами <i>Ruby</i></li> </ul> <b>Большинство учащихся смогут:</b> <ul style="list-style-type: none"> <li>научатся находить среду <i>Ruby</i></li> </ul> <b>Некоторые учащиеся смогут:</b> <i>Написать программу</i>	
<b>Критерии оценивания</b>	<b>Грамотность -1балл</b> <b>Оригинальность -1балл</b> <b>Время -1 балл</b>	
<b>Воспитаниеценностей</b>	Воспитание уважения к мнению друг друга, ответственность, коммуникативные способности, критическое мышление	
<b>Предварительныезнания</b>	История Ruby, «Интерпретация» Компиляция и интерпретация, для программистов, Что умеет Ruby? Что не умеет Ruby?	
<b>Межпредметные</b> связи	Информатика, математика	
<b>Запланированные этапы урока</b>	<b>Запланированная деятельность на уроке</b>	<b>Ресурсы</b>
<b>Начало урока</b> _5__ мин	<b>Организационный момент.</b> Приветствие учащихся. Создание положительного эмоционального фона. Настрой на рабочий лад. Показываю тему урока и цель урока, озвучиваю критерии для цели урока <b>Деление на подгруппы.</b> Стратегия «Нумерация»	Карточки с цифры
<b>Середина урока</b> _30__ мин	<b>Организационный момент.</b> <i>Определите смысл этих слов, что они из себя представляют:</i> <i>CSS / HTML</i> <i>PHP</i> <i>SQL</i> <i>JavaScript</i> <i>Python</i> <i>Go</i> <i>Java</i> <i>Ruby</i> <i>C++</i> <i>C</i> <ul style="list-style-type: none"> <li>Детям раздаются карточки с вышеуказанными словами, дети в группе определяют что это и где они используются, они могут воспользоваться мобильными средствами или классным</li> </ul>	Карточки для стратегии светофор 

## компьютеров на его исполнение некоторые ограничения...



Строка создаётся при помощи ограничительных символов. Для этих целей чаще всего используются `"` (программистская кавычка) и `'` (машинный апостроф). Их смысл различен. Строка в апострофах гарантирует, что в ней будет содержаться тот же текст, что и в коде программы, без изменений. Строка в кавычках будет проходить предварительное преобразование. Будут раскрыты конструкции «вставка» и «специальный символ».

Давайте будем называть строки в апострофах «ленивыми», а строки в кавычках — «работящими».

Вставка — это хитрая конструкция, которая вставляется между ограничительными символами внутри строки. Она состоит из комбинации `#{` и двух фигурных скобок (`{ 'здесь был Вася' }`). Внутри неё можно писать не только `'Здесь был Вася'`, но и любой программный код. Результат программного кода будет преобразован к строке и вставлен на место вставки.

Вставка жизнью заправляет:  
Код программный выполняет,  
Тихо результат считает,  
Вместо вставки подставляет.

«Вставка» работает только в момент создания строки. После создания придётся придумывать другие способы подстановки данных в строки.

Специальный символ начинается со знака `\` (обратная косая черта). Самые популярные из них: `\n` (переход на новую строку), `\t` (табуляция), `\\` (обратная косая черта) и `\"` (двойная кавычка).

Хотя специальный символ и пишется, как два знака, но на деле это всего один символ. Доказать это можно выполнением простенького кода: `"\n".size #=> 1`.

Массив создаётся как минимум тремя способами. Первый способ:

```
[1, 2, 3, 4, 5, 6]
```

Вы просто перечисляете элементы массива через запятую, а границы массива обозначаете квадратными скобками. С таким методом создания массива мы уже встречались. А теперь попробуем второй способ, через вызов метода `.new` класса `Array`:

```
Array.new(6){ |index| index + 1 } #=> [1, 2, 3, 4, 5, 6]
```

Параметром метода `.new` является количество элементов будущего массива (в данном случае это число 6). В фигурных скобках указано, как мы будем заполнять массив. В данном случае значение элемента массива будет больше на единицу его индекса. Третий способ заключается в создании объекта типа `Range` (диапазон) и вызове метода `.to_a`:

	<p>(1..6).to_a #=&gt; [1, 2, 3, 4, 5, 6]</p> <p>Есть ещё много способов, но эти три используются чаще всего.</p>	
<p><b>Конец урока</b> <b>_5_ мин</b></p>	<p><b>Рефлексия:</b> Метод «Рефлексивные карточки» Было интересно... Я понял, что... У меня получилось... Расскажу дома, что...</p>	<p>«Рефлексивные стикеры»</p> 
<p><b>Дифференциация – каким образом Вы планируете оказать больше поддержки? Какие задачи Вы планируете поставить перед более способными учащимися?</b></p>	<p><b>Оценивание – как Вы планируете проверить уровень усвоения материала учащимися?</b></p>	<p><b>Охрана здоровья и соблюдение техники безопасности</b></p>