Занятие 1 . Рисование массива шариков

Задание.

Нарисовать на экране 30 шариков случайного размера (от 10 до 40px) и случайного цвета, используя массив объектов.

В этом примере будет использоваться API-интерфейс Canvas для рисования шарика на экране.

Очень простой HTML-документ с элементом <h1>, элементом <canvas> для рисования наших шаров и элементами CSS и JavaScript.

Первая часть скрипта выглядит так:

```
const canvas = document.querySelector('canvas');

const ctx = canvas.getContext('2d');

const width = canvas.width = window.innerWidth;
const height = canvas.height = window.innerHeight;
```

Этот скрипт получает ссылку на элемент <canvas>, затем вызывает для него метод getContext (), чтобы дать нам контекст, в котором мы можем начать рисовать. Результирующая константа (ctx) является объектом, который непосредственно представляет область рисования холста и позволяет нам рисовать на нем 2D-фигуры.

Затем мы устанавливаем константы, называемые ширина и высота, а также ширину и высоту элемента canvas (представленного свойствами canvas.width и canvas.height) равными ширине и высоте области просмотра браузера (области, на которой отображается веб-страница). - это можно получить из свойств Window.innerWidth и Window.innerHeight).

Моделирование шарика

В нашей программе в будущем будет много мячей, прыгающих по экрану. Поскольку все эти шары будут вести себя одинаково, имеет смысл представлять их объектом. Давайте начнем с добавления следующего конструктора в конец нашего кода.

```
function Ball(x, y, velX, velY, color, size) {
  this.x = x;
  this.y = y;
  this.velX = velX;
  this.velY = velY;
  this.color = color;
  this.size = size;
}
```

Здесь мы включаем некоторые параметры, которые определяют свойства, необходимые каждому мячу для функционирования в нашей программе:

координаты x и y - горизонтальные и вертикальные координаты, где начинается шарик на экране. Это может варьироваться от 0 (верхний левый угол) до ширины и высоты области просмотра браузера (нижний правый угол).

горизонтальная и вертикальная скорость (velX и velY) - каждому шару присваивается горизонтальная и вертикальная скорость; в реальном выражении эти значения регулярно добавляются к значениям координат x / y, когда мы анимируем шары, чтобы перемещать их на столько на каждом кадре.

цвет - каждый шар получает цвет.

размер - каждый шар получает размер - это его радиус в пикселях.

Это обрабатывает свойства, но как насчет методов? Мы хотим, чтобы наши шары действительно что-то делали в нашей программе.

Рисование шарика

Свойство **Object.prototype** представляет объект прототипа Object(прототипы - это механизм, с помощью которого объекты JavaScript наследуют свойства друг от друга).
JavaScript часто описывают как язык **прототипного наследования** — каждый объект, имеет объект-прототип, который выступает как шаблон, от которого объект наследует методы и свойства.
Добавим следующий метод draw() в прототип Ball():

```
Ball.prototype.draw = function() {
  ctx.beginPath();
  ctx.fillStyle = this.color;
  ctx.arc(this.x, this.y, this.size, 0, 2 * Math.PI);
  ctx.fill();
}
```

Функциональное Выражение

Согласно документации ECMA синтаксис определения функции следующий:

Объявление функции:
```
function Идентификатор () { Тело Функции }
```

Функциональное выражение:
```
function () { Тело Функции }
```

Оно выглядит вот так:

```
let sayHi = function() {
  alert( "Привет" );
};
```

В коде функция создаётся и явно присваивается переменной, как любое другое значение. По сути без разницы, как мы определили функцию, это просто значение, хранимое в переменной sayHi.

Задание:

1. **Разместите Canvas во весь экран на странице.**
2. **Определите объект Шарик (Моделирование шарика).**
3. **Создайте новый экземпляр шара:**

```
let Ball-1 = new Ball(50, 100, 4, 4, 'blue', 10);
```

4. **Опишите функцию рисования шарика (Рисование шарика).**

5. **Выведите свойства шарика и нарисуйте его:**
```
Ball-1.x, Ball-1.y, Ball-1.velX, Ball-1.velY, Ball-1.size, Ball-1.color;
Ball-1.draw();
```

6. **Создайте второй экземпляр шара со случайными значениями свойств.**

```
let Ball-2 = new Ball(…);
```

7. **Создайте массив для 30 шариков и добавьте в массив экземпляр шара со случайными значениями свойств, нарисуйте все шарики из массива:**

```
let Balls = [];
```