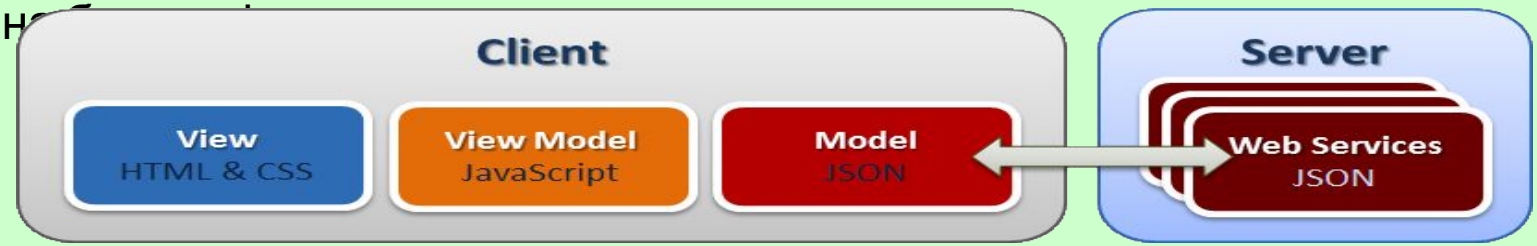
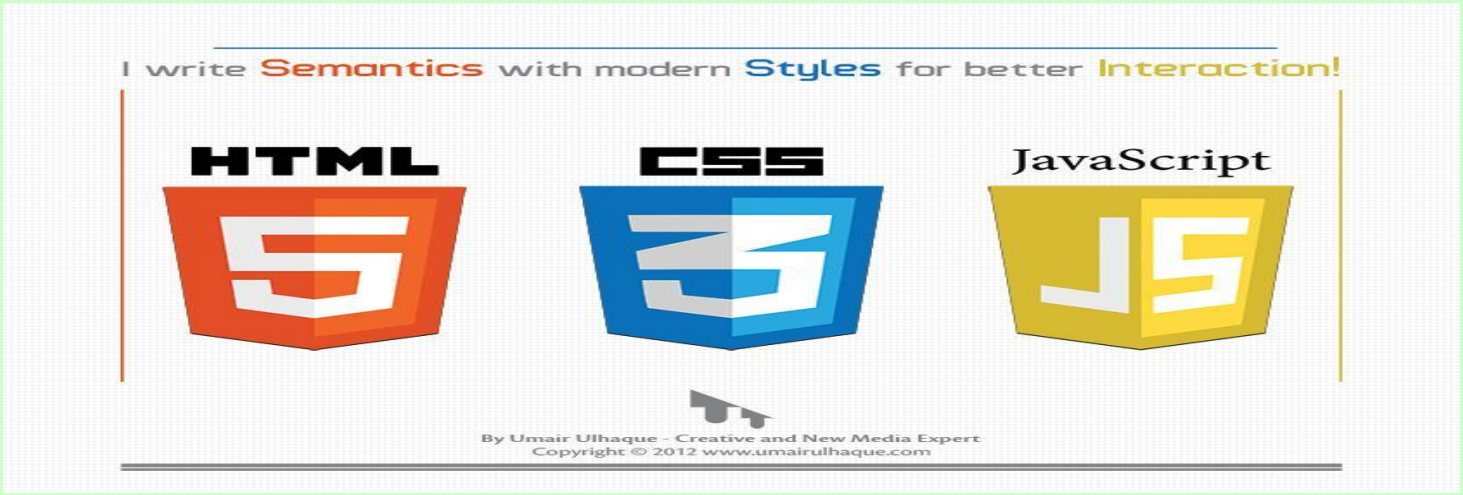
Графический редактор

Подготовил: Статьев Владлен!

Здравствуйте, сейчас я представлю вам свой проект , который называется «Графический редактор». Создавался мой Графический редактор как веб-проект, с помощью HTML5 (языка для структурирования и представления содержимого всемирной паутины) и JavaScript (языка программирования, как правило, выполняется на стороне клиента, придающий интерактивность веб-страницам) и CSS (формальный язык описания внешнего вида документа, написанного с использованием языка разметки). То есть на HTML5 были написаны все блоки(ячейки) и кнопки; на JavaScript все -- функции (рисование, отображение картинок и т.д.); а без CSS (красот сайта) мы бы видели на экране только пару кнопок н





Теперь посмотрим внутренность кода сайта. У меня есть два отдельных файла. В первом HTML и JS, а во втором CSS. И для того чтобы между ними была связь я в HTML указываю, что надо подключить к этому файлу файл CSS. На экране вы увидите разные цветные прямоугольники(блоки) и кнопки, которые были созданы в HTML. У меня есть canvas (элемент HTML5, предназначенный для создания двухмерного изображения при помощи скриптов), в котором происходит основные действия рисования. Как я уже говорил все функции графического редактора работают и были созданы только благодаря JS. В нем имеется множество различных функций, которые создаются по такому же алгоритму как и в паскале (задаем функцию, называем её, передаем переменные и указываем что мы хотим сделать). Но как функции понимают когда им надо работать? Не могут же они все сразу одновременно начать это делать, тогда смысла в редакторе не будет. Например мне надо сделать так, чтобы при нажатии на кнопку «круг», он появился на экране, тогда я в HTML при создании кнопки «круг» в ее параметрах кнопки указал событие onclick=“Название функции()”, т.е. при нажатие на эту кнопку на экране появится круг.

Вообще мой JS построен на объектно-ориентированном программирование. Объект представляет собой неупорядоченный набор пар вида «ключ-значение». Каждая такая пара называется свойством объекта (функции называются методами), каждое свойство должно иметь уникальное имя, которое может быть строкой или числом. Значение свойства может быть любым: как значением простого типа, так и другим объектом. Т.е. у меня есть объекты , которые имеют свои методы (функции про которые я упоминал) и события. Так вот этими событиями, которые я использовал являются: нажатие кнопки или блока(onclick), нажатие на левую часть мыши (mousedown),отпускание левой части мыши(mouseup) и передвижение мыши (mousemove). Но есть и много других. При[illegible]из двух других событий: mousedown и mouseup.

Разберем функции:

*если брать например выбор цвета или ширины следа: У нас есть функция в которой содержится переменная обозначающая цвет или ширину, а в самой функции указываем что мы меняем параметр следа на текущий, и этот текущий будет написан вместо переменной в событии параметра кнопки. Тогда при клике на ту кнопку параметр изменится на указанный.

*теперь взять например функцию которая отвечает за рисовании на экране трафаретов: тут в принципе нам тоже придется указать переменную которой мы будем передавать текущее значение, но также создать новый объект в событие(onload) которого мы и укажем ту самую переменную. Это необходимо потому что onload — событие которое происходит при загрузке документа (когда пользователь своими действиями вызвал загрузку веб-страницы). Т.е. это событие регулирует то, чтобы наше изображение по ширине и длине вместилось в нужный нам блок.

*а вот функции для сохранения текущего и восстановления сохраненного изображения еще интереснее: тут мы у объекта document вызываем метод createelement (а также свойство этого объекта добавляет объект канвас), метод createelement создает канвас элемент(переменную), но она не отображается на экране и имеет такую же длину и ширину как основной канвас. При нажатии на пнопку «Сохранить» мы копируем изображение с основного канваса и сохраняем в этот элемент. При нажатии на кнопку «Восстановить» мы копируем из нашего внеэкранного элемента значение, которое он хранит и вставляем его в основной экранный канвас. И теперь это у нас уже не как нарисованное что –то, а просто изображение чего –то нарисованного которое мы можем изменять.

*еще есть такие функции которые рисуют геометрические фигуры (два вида квадрата, треугольник и круг). Но они у меня появляются пока лишь в определенном месте. И чтобы не возникла такая проблема, как неожиданная смена цвета фигуры нужно задать переменные определенного цвета которые в конце функции мы снова будем присваивать цвету

Работа с мышью:

Основную работоспособность в редакторе самостоятельно рисовать выполняет мышь (перо), потому что именно оно оставляет след(тем самым мы как-будто рисуем).И для этого создавалось определенные объекты, а именно: *нам нужен объект который выделит рамки рисования(у нас в поле канваса) и тогда все необходимые действия (рисование, появление изображения, сохранение и т.д.) будут осуществляться именно в нем. Также необходим объект который будет выполнять функцию именно отображения происходящих событий на экране канваса в 2d. Также очень важно где же оставить перу след, для этого понадобится еще один объект у которого есть 2 свойства координаты по ширине и по высоте, которые будут передаваться когда произойдет событие нажатие на мышь и до тех пор пока не произойдет событие отпустили мышь, во все остальное время в этих свойствах будет показывать просто координату (0,0) (это задано по условию), тем самым наше поле канваса уже представляет из себя координатную плоскость. И последний объект который нам необходим это объект-флаг, потому что его значению присваивается true если было выполнено событие нажата мышь и false при её отпускании. Таким образом этот объект отображает был запущен процесс рисования или нет (только если запущен тогда будет что-то отображаться на экране). Но важно заметить, что чтобы что-то произошло при нажатии на кнопку мы в параметрах этой кнопки указывали onclick, а тут получается на один блок(поле) канваса необходимо сразу три функции mousdown, mousemove и mouseup чего сделать нельзя. Тогда в скрипте нам надо объекту анализирующему происходящее в канвасе задать три раза метод addEventListener который позволяет назначить на элемент обработчики событий , т.е. с его помощью, можно указать, например, что делать при клике по кнопке, или что делать при наборе текста в текстовом поле. А три раза это надо сделать потому что у нас три разных события с соответственно тремя разными функциями. В каждой из этих трьох функций мы проверяем или переприсваиваем объекту-флагу true или false, а также считываем текущие координаты мыши.

Это все что было необходимо в первом файле с HTML и JS! Во втором файле с CSS я просто реализовывал красоту сайта(где, какого размера и ц
выглядеть все блоки и весь сайт торой файл б зует
представление человека и взгляд -дизайн, т.е э часть!

А ТЕПЕРЬ ДАВАЙТЕ ПОСМОТРИМ «ГРАФИЧЕСКИЙ РЕДАКТОР»

В ДЕЛЕ!