

Графический редактор

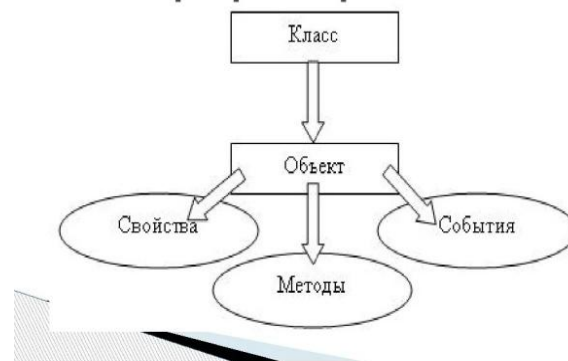
Подготовил: Статъев Владлен!

Здравствуйте, 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 на белом фоне. Использовались знания, полученные на факультативах МАН по веб-программированию.



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. У меня есть canvas (элемент HTML5, предназначенный для создания двухмерного изображения при помощи скриптов), в котором происходят основные действия рисования. В JS 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()". Вообще мой JS построен на объектно-ориентированном программировании. 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. У меня есть объекты, которые имеют свои методы (функции) 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).

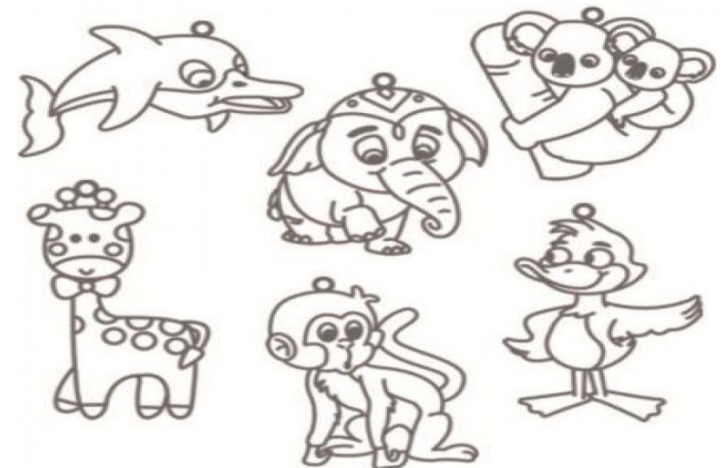
Объектно-ориентированное программирование



Разберем некоторые интересные функции:

*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 вместились 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 кнопку «Сохранить» мы копируем изображение с основного канваса и сохраняем в этот элемент. При нажатии на кнопку «Восстановить» мы копируем из нашего внеэкранный элемент значение, которое он хранит и вставляем его в основной экраный канвас.



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 создавались 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.}{У меня есть блок источника(мышь), блок принятия(канвас) и блок обработки(метод объекта) событий}. 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), тем самым наше поле канваса уже представляет из себя координатную плоскость. И последний объект который нам необходим это объект-флаг, который отображает был запущен процесс рисования или нет (только если запущен тогда будет что-то отображаться на экране). В каждом событии мыши мы считываем её текущие координаты.

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 цвета и как будут выглядеть все блоки и весь сайт в целом), т.е это дизайнерская часть!



А ТЕПЕРЬ ДАВАЙТЕ
ПОСМОТРИМ
«ГРАФИЧЕСКИЙ РЕДАКТОР»
В ДЕЛЕ!