|  |  |
| --- | --- |
|  | <!DOCTYPE html> http://sekretov-net.ru/stroim-veb-sajt/css-3/plavnaya-smena-izobrazhenij-css<**html**>     <**head**>        <**title**>Плавная смена изображений</**title**>        <**meta** charset="windows-1251">        <**meta** name="viewport" content="width=device-width, initial-scale=1.0">                 <**style** type="text/css">            @import url("styles.css");        </**style**>             </**head**>        <**body**>        <**div** class="div-style">        <**img** src="/image-1.jpg" title="Image 1" class="img-style im-1">        <**img** src="/image-2.jpg" title="Image 2" class="img-style next im-2">        <**img** src="/image-3.jpg" title="Image 3" class="img-style next im-3">        <**img** src="/image-4.jpg" title="Image 4" class="img-style next im-4">     </**div**>  </**body**>        </**html**> |



На диаграмме значение Y=1 соответствует непрозрачности изображения - показу, а Y=0 полной прозрачности. Диаграмма каждого изображения отображена своим цветом. Все изображения будут отображаться по 4 секунды, а длительность перехода 2 секунды. Общее время выполнения всего цикла можно будет в дальнейшем менять, отношение показ/переход нет. Рисуем слева направо и сверху вниз.

1-е изображение 0-4 показ, 4-6 вниз, далее прозрачность. Эту диаграмму закончим когда нарисуем 4-ю

2-е изображение 0-4 прозрачность, 4-6 вверх, 6-10 показ, 10-12 вниз, прозрачность до конца

3-е изображение 0-10 прозрачность, 10-12 вверх, 12-16 показ, 16-18 вниз, прозрачность до конца

4-е изображение 0-16 прозрачность, 16-18 вверх, 18-22 показ, 22-24 вниз

Поскольку 4-е изображение является последним полный цикл получился 24 секунды и в интервале 22-24 секунды диаграмма 1-го изображения должна уйти вверх.

Посчитаем процентное значение времени каждой переходной точки:

4-я секунда 4/24\*100=16%

6-я секунда 6/24\*100=25%

10-я секунда 10/24\*100=41%

12-я секунда 12/24\*100=50%

и так же для всех остальных. Осталось написать CSS-стили

**CSS3**

|  |  |
| --- | --- |
| 123456789101112131415161718192021222324252627282930313233343536373839404142434445464748495051525354555657585960616263646566676869707172737475767778798081 | @keyframes fadeIm-1{    from    { opacity: 1; }    16%     { opacity: 1; }    25%     { opacity: 0; }    91%     { opacity: 0; }    to      { opacity: 1; }} @keyframes fadeIm-2{    from    { opacity: 0; }    16%     { opacity: 0; }    25%     { opacity: 1; }    41%     { opacity: 1; }    50%     { opacity: 0; }    to      { opacity: 0; }} @keyframes fadeIm-3{    from    { opacity: 0; }    41%     { opacity: 0; }    50%     { opacity: 1; }    66%     { opacity: 1; }    75%     { opacity: 0; }    to      { opacity: 0; }} @keyframes fadeIm-4{    from    { opacity: 0; }    66%     { opacity: 0; }    75%     { opacity: 1; }    91%     { opacity: 1; }    to      { opacity: 0; }} .div-style{    position: relative;     text-align: center;     width: 99%; } .img-style{    width: 33.3%;    margin-top: 8px;    border: 1px solid #0000ff;         animation-duration: 24s;    animation-iteration-count: infinite;} .img-style.next{    position: absolute;    left: 33.3%;    opacity: 0;} .img-style.im-1{     animation-name: fadeIm-1;} .img-style.next.im-2{     animation-name: fadeIm-2;} .img-style.next.im-3{    animation-name: fadeIm-3;} .img-style.next.im-4{    animation-name: fadeIm-4;} |

Сначала настроим ключевые кадры - *@keyframes* - согласно построенным диаграммам - строки 1-37. Значения *opacity* на интервалах должны соответствовать значениям линий на диаграммах. Разберем на примере *@keyframes fadeIm-1:*

*о*т *from* до 16% *opacity: 1;*

от 16% до 25% *opacity* меняется от 1 до 0;

от 25% до 91% *opacity: 0;*

от 91% до конца (и начала следующего цикла) *opacity* поднимается от 0 до 1.

В стиль общий для всех рисунков добавляем параметры анимаций - в строке 52 устанавливаем время выполнения анимаций - там вовсе не обязательно должно быть 24, в строке 53 указываем, что анимации повторяются в непрерывном цикле. В стиль каждого рисунка добавляем свои анимации - строки 63-81. Для того, чтобы это работало с браузерами на движке *webkit* необходимо продублировать каждую анимацию с префиксом *-webkit-*, как показано ниже, и добавить их в стили. Обратите внимание на два дефиса.

|  |  |
| --- | --- |
| 1234567891011121314151617181920212223 | @keyframes fadeIm-1{    from    { opacity: 1; }    16%     { opacity: 1; }    25%     { opacity: 0; }    91%     { opacity: 0; }    to      { opacity: 1; }} @-webkit-keyframes fadeIm-1{    from    { opacity: 1; }    16%     { opacity: 1; }    25%     { opacity: 0; }    91%     { opacity: 0; }    to      { opacity: 1; }} .img-style.im-1{     animation-name: fadeIm-1;    -webkit-animation-name: fadeIm-1;} |

Достоинства такого подхода очевидны - один раз написать анимации и потом использовать их где надо как обычные *CSS*-стили. Эти же настройки я использовал для плавной смены фоновых изображений - статья [CSS3 Плавная смена фона](http://sekretov-net.ru/stroim-veb-sajt/css-3/plavnaya-smena-fona-css3). Можно создать несколько блоков - для разного количества картинок и для удобства поместить их в отдельный файлы. Такое решение полностью свободно от недостатков, присущих реализации плавной смены картинок с помощью *JavaScript*. Более того, плавная смена картинок продолжается при потере окном браузера фокуса.

Алгоритм прост и понятен, не надо быть программистом чтобы разобраться, как это работает. И на его реализацию требуется совсем немного времени. Чтобы заменить плавную смену картинок в шапке сайта с *JavaScript* на *CSS3,* начиная с построения диаграмм и заканчивая проверкой в пяти браузерах, мне потребовалось всего 40 минут! Ниже рабочие диаграммы.



Параметры смены изображений такие: изображений - 10, непрозрачность - 4с., переход - 3с. Полное время цикла - 70с.