

Лабораторна робота №10

Робота з зображеннями

Мета роботи: дослідити методи динамічного розміщення зображень на сторінці і навчитися створювати малюнки з допомогою сценарію.

Попереднє завантаження зображень

Зображення, що розміщаються на сайтах, повинні задовольняти двом таким вимогам. Вони повинні мати високу якість і швидко завантажуватися, але чим вище якість зображення, тим більше займає зображенням обсяг пам'яті і тим повільніше воно завантажується. Завдання побудови зображення, що має прийнятні обсяг і якість, вирішується за допомогою графічних пакетів прикладних програм, наприклад, Photoshop з підключеним до нього пакетом ImageReady.

Засобами JavaScript можна спочатку завантажити кілька зображень в кеш, а потім виводити їх по одному на екран по команді користувача. Такий прийом заснований на психології користувача Інтернету. Дослідним шляхом доведено, що людина не помічає паузи між моментами посилки запиту й одержанням відповіді, якщо ця пауза не більш чотирьох секунд. Користувач звик чекати кілька десятків секунд, поки завантажиться обрана ним сторінка, а при роботі з уже завантаженої сторінкою болісно реагує на виникаючі паузи.

Для завантаження зображення в пам'ять без відображення його на екрані служить об'єкт Image, якому не відповідає жоден елемент HTML-документа. Створюється і використовується об'єкт Image так:

```
var risunok = new Image (ширина, висота) // створення об'єкта  
risunok.src = "pict.jpg" // завантаження pict.jpg в пам'ять  
document.images [0] .src = risunok.src // відображення на екрані
```

Завдання 1. Додати сторінку (рис. 1), на якій користувач може переглядати фотографії, вибираючи їх назви з поля зі списком (тег <SELECT>).

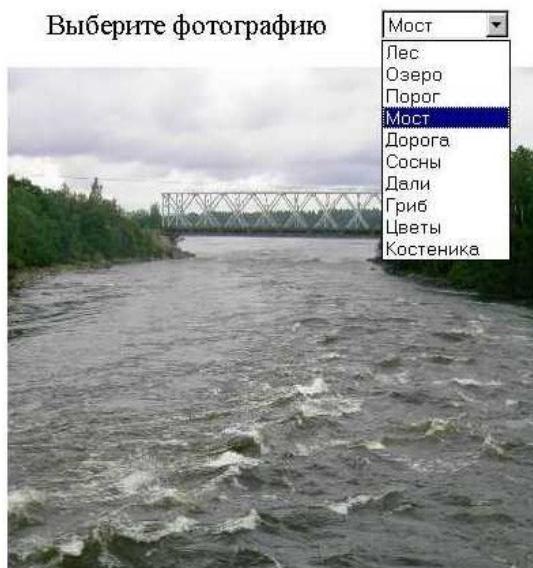


Рис. 1

Вказівки. Використовуйте асоціативний масив, елементами якого будуть об'єкти Image, тобто фотографії, а ключами - значення атрибути VALUE тега <SELECT>. Створення масиву і одного елемента робиться так:

```
var photo = new Array ()
```

```
photo [ "les" ] = new Image (400,550); // елемент масиву - об'єкт
photo [ "les" ]. src = "les.jpg"; // властивість об'єкта - ім'я файлу
з малюнком
```

Виклик функції, що служить для появи у вікні браузера обраної фотографія, проводиться при натисканні мишкою по елементу списку. Ця подія називається onchange. Як параметр функції передається об'єкт SELECT:

```
<SELECT onchange = "prosm (this)">
<option value = "les"> Ліс
```

Функція, правильно сприйнята браузерами Internet Explorer і Mozilla, складається з трьох рядків:

```
function prosm (list)
{D = list.options [list.selectedIndex] .value // d = "les"
document.fot.src = photo [d] .src; // fot - ім'я тега IMG}
```

Файли з фотографіями оберіть самостійно.

Малювання

В JavaScript зображення можна побудувати трьома способами:

- за принципом мозаїки, використовуючи в якості елемента маленьку картинку або пофарбований елемент <div> (мінімальний розмір елемента мозаїки - один піксель);
- засобами мови SVG (від англ. Scalable Vector Graphics - масштабована векторна графіка);
- за допомогою елемента canvas мови HTML5.

Крім того, відрізок вертикальної прямої можна зобразити за допомогою тега <DIV>, задавши ширину і висоту контейнера рівними товщині і довжині лінії відповідно. Горизонтальний відрізок будеться за допомогою тега <DIV> і вкладеного в нього тега <HR>.

Приклади малювання усіма трьома способами можна подивитися [тут](#)

Ідея малювання першим способом кривої, описаної аналітично, полягає в наступному:

- створюється картинка розміром 1 піксель, залита чорним або якимось іншим кольором;
- ця картинка розмножується по заданій траєкторії.

Замість картинки можна використовувати елемент <DIV>.

У прикладі 1 описаними методами будуться чотири прямих.

```
<HTML>
<HEAD>
<TITLE>Прямые линии</TITLE>
<script>
line(100,100,400,400,3)
line(100,400,400,100,10)
function line(x1,y1,x2,y2,n)
{ var xstr=""
  var xstr0=
  '<IMG SRC="Tochka.bmp" WIDTH='+n+' HEIGHT='+n+
  ' STYLE="position:absolute;"'
  var k=(y2-y1)/(x2-x1)
  x=x1
  while (x<=x2)
  { xstr+=xstr0+'top:'+(k*(x-x1)+y1)+';left:'+'x+'>';
    x++
  }
  document.write(xstr)
}
```

```

</script>
</HEAD>
<BODY>
</div>
<div style="font-size:20pt;padding-left:50">
<B>Рисование прямых линий</b></div>
<DIV id="dd" STYLE="position:absolute;top:250;left:100;
width:300;height:1;"><hr noshade></div>
<div ID="kv" STYLE="position:absolute;top:100;left:250;
width:1;HEIGHT:300;background-color:#ff0099"></div>
</BODY></HTML>

```

Всі прямі перетинаються в одній точці (рис. 2). Горизонтальна і вертикальна прямі побудовані без використання скрипта. Похилих прямі побудовані за допомогою функції line (x1, y1, x2, y2, n), де x1, y1 - координати початку відрізка прямої, x2, y2 - координати кінця, n - товщина лінії.

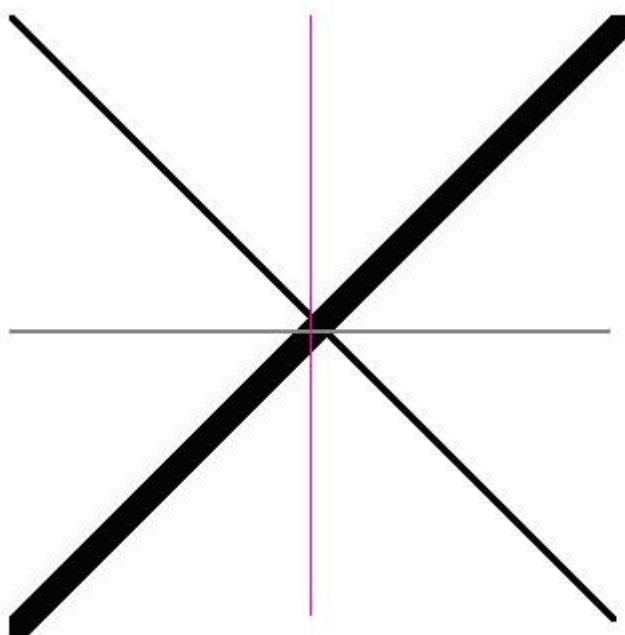


Рис.2

Завдання 2. Додати сторінку (рис. 3), на якій будується еліпс з розмірами великої і малої півосей, які задаються користувачем. Еліпс потрібно побудувати усіма трьома зазначеними вище способами.

Вказівки. Для побудови еліпса першим способом зручно використовувати його параметричні рівняння:

$$x = a * \cos(t); y = b * \sin(t),$$

де: a, b - півосі, $0 \leq t \leq 2\pi$

Построение Эллипса

Задайте длины полуосей

горизонтальная $a =$

вертикальная $b =$

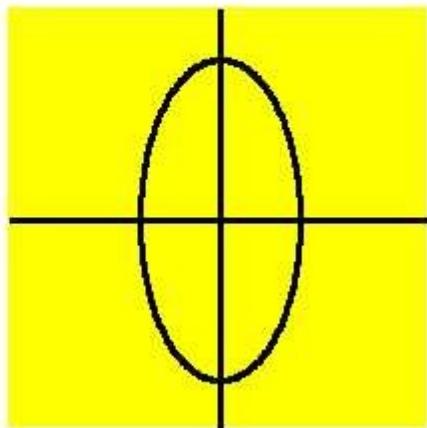


Рис.3