Московский государственный
технический университет
имени Н.Э. Баумана

Кафедра ИУ5
«Системы обработки информации
и управления»

ИТ-каникулы

Алгоритмы в языке программирования JavaScript

Канев Антон Игоревич
преподаватель кафедры ИУ5

aikanev@bmstu.ru

JSON

- JSON (JavaScript Object Notation) — текстовый формат обмена данными, основанный на JavaScript.

```xml
<person>
  <firstName>Иван</firstName>
  <lastName>Иванов</lastName>
  <address>
    <streetAddress>Московское ш., 101, кв.101</streetAddress>
    <city>Ленинград</city>
    <postalCode>101101</postalCode>
  </address>
  <phoneNumbers>
    <phoneNumber>812 123-1234</phoneNumber>
    <phoneNumber>916 123-4567</phoneNumber>
  </phoneNumbers>
</person>
```

```json
{
   "firstName": "Иван",
   "lastName": "Иванов",
   "address": {
       "streetAddress": "Московское ш., 101, кв.101",
       "city": "Ленинград",
       "postalCode": 101101
   },
   "phoneNumbers": [
       "812 123-1234",
       "916 123-4567"
   ]
}
```

```xml
<person firstName="Иван" lastName="Иванов">
  <address streetAddress="Московское ш., 101, кв.101" city="Ленинград" postalCode="101101" />
  <phoneNumbers>
    <phoneNumber>812 123-1234</phoneNumber>
    <phoneNumber>916 123-4567</phoneNumber>
  </phoneNumbers>
</person>
```

JavaScript

- Мультипарадигменный язык программирования
- Объектно-ориентированный, императивный и функциональный стили
- Наиболее широко используется в браузерах для выполнения скриптов и интерактивности страниц
- Пример внутри страницы:

```
<script type="application/javascript">
  alert('Hello, World!');
</script>
```

JavaScript

- Расположение внутри тега

```
<a href="delete.php" onclick="confirm('Вы уверены?'); return false;">
  Удалить
</a>
```

JavaScript

- Включение в отдельный файл

```
<body>
  <script type="application/javascript" src="http://Путь_к_файлу_со_скриптом">
  </script>
</body>
```

Строки

```
var str1="string1";
var str2 = new String("string2");
```

```
// вывод жирной зачеркнутой строки
document.write("Hello, world".strike().bold();
```

Массив

```
 var array = new Array ("red","green","blue"); // создаем массив
из трех элементов
 array[4]="alfa"; // добавляем 4 элемент
 array.sort(); // сортируем
 document.write(array.join(", ").bold);
```

```
var empty=[]; // пустой массив
var color = ["blue","red"];
document.write(color.join(", ").italics());
```

Ассоциативные массивы

```
var items={"item":1,
           "zero":function(){return 0;},
           "rgb":["red","green","blue"]
           };
document.write(items["rgb"][1]);
```

Условный оператор

```
/*
общая форма оператора if
if(cond) operator1 else operator2
*/
if(x==5)
  {
  y=3.14;
  z=2*3.14;
  }

if(x!=20)
  y=3.14;
else
  {
  y=3.14*2;
  z=0;
  }
```

Оператор выбора

```
/* общая форма оператора выбора
switch(expr) {
case val1: .... break;
...
case valn: ... break;
default: ... // метка по умолчанию
}
*/
var x="number"; // попробуйте разные значения
switch(x) {
  case 3: document.write("case 3:"); break;
  case 3.14: document.write("case 3.14:"); break;
  case "number": document.write("case number:"); break;
  default: document.write("default:");
}
```

Циклы

```
for(var i=0;i<110;i++)
    x=func(i);


//--
var x=1;
for(i=0,j=10;i<50 && j<50;i++,j+=5)
    x+=i+j;
document.write(x);
```

```
/* общая форма оператора выбора
switch(expr) {
case val1: .... break;
...
case valn: ... break;
default: ... // метка по умолчанию
}
*/
var x="number"; // попробуйте разные значения
switch(x) {
  case 3: document.write("case 3:"); break;
  case 3.14: document.write("case 3.14:"); break;
  case "number": document.write("case number:"); break;
  default: document.write("default:");
}
```

Циклы

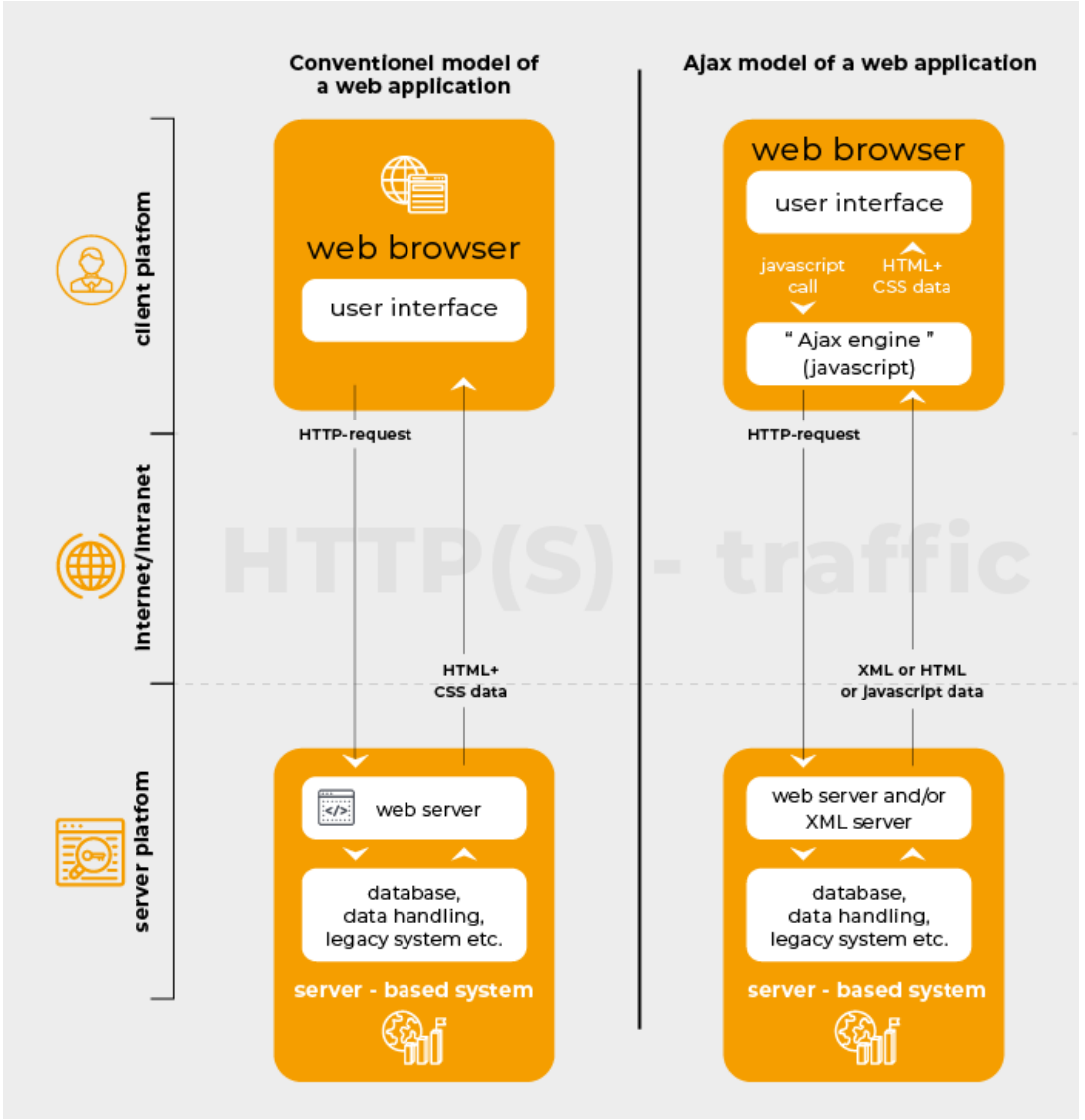
```
/* общая форма по элементного цикла
for (variable in [object | array])  operator
*/
var obj = new Object(); // новый объект
var cnt=0;
// добавляем три свойства к объекту
obj.prop1=1;
obj.prop2="dfsdf"
obj.prop3=3.14;
for(i in obj) // подсчитываем число пользовательских свойств
   cnt++;
document.write(cnt);

//--
var array = new Array(1,2,3,4,5,6,7,8,9), cnt=0;
for(i in array)
   cnt+=array[i];
document.write(cnt);
```

Функции

```
function sum() {
  var l=arguments.length,r=0;
  for(i=0;i<l;i++)
    r+=arguments[i];
  return r;
}

document.write("sum="+sum(2,3,20));
```

AJAX

- **AJAX**, Ajax (Asynchronous Javascript and XML — «асинхронный JavaScript и XML») — подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с веб-сервером.
- В результате при обновлении данных веб-страница не перезагружается полностью, и веб-приложения становятся быстрее и удобнее.



Практическое задание

```
<body>
  <!-- Рисуем игровое поле -->
  <canvas width="750" height="585" id="game"></canvas>
      <img src="assets/ball.png" id="ball"></img>
      <img src="assets/paddle.png" id="paddle"></img>
      <img src="assets/background.jpg" id="background"></img>
      <h5 class="logging"></h5>
      <h1 class="datasetSize">0</h1>
      <div class="download">
          <img src="assets/download.svg" width="50px"><a download="ping_pong_dataset.csv" href='none'></a></img>
      </div>
      <h1 class="score">
              Current: <a>0</a><br/>
              Best score: <a>0</a><br/>
              Average: <a>0</a>
```

Практическое задание

```
  </h1>
      <p class='upload'>
     <label for="file_input">
      <img src="assets/upload1.svg" width="50px"></img>
       <img src="assets/upload2.svg" width="50px" hidden></img>
      ONNX file
     </label>
     <input type="file" id='file_input' class="modelFile" accept=".onnx"/>
  </p>
      <script src="js/render.js"></script>
      <script src="js/pause.js"></script>
      <script src="js/controls.js"></script>
      <script src="js/logging.js"></script>
      <script src="js/scores.js"></script>
      <script src="js/download.js"></script>
      <script src="js/onnx.min.js"></script>
      <script src="js/ai.js"></script>
      <script src="js/engine.js"></script>
</body>
```

Практическое задание

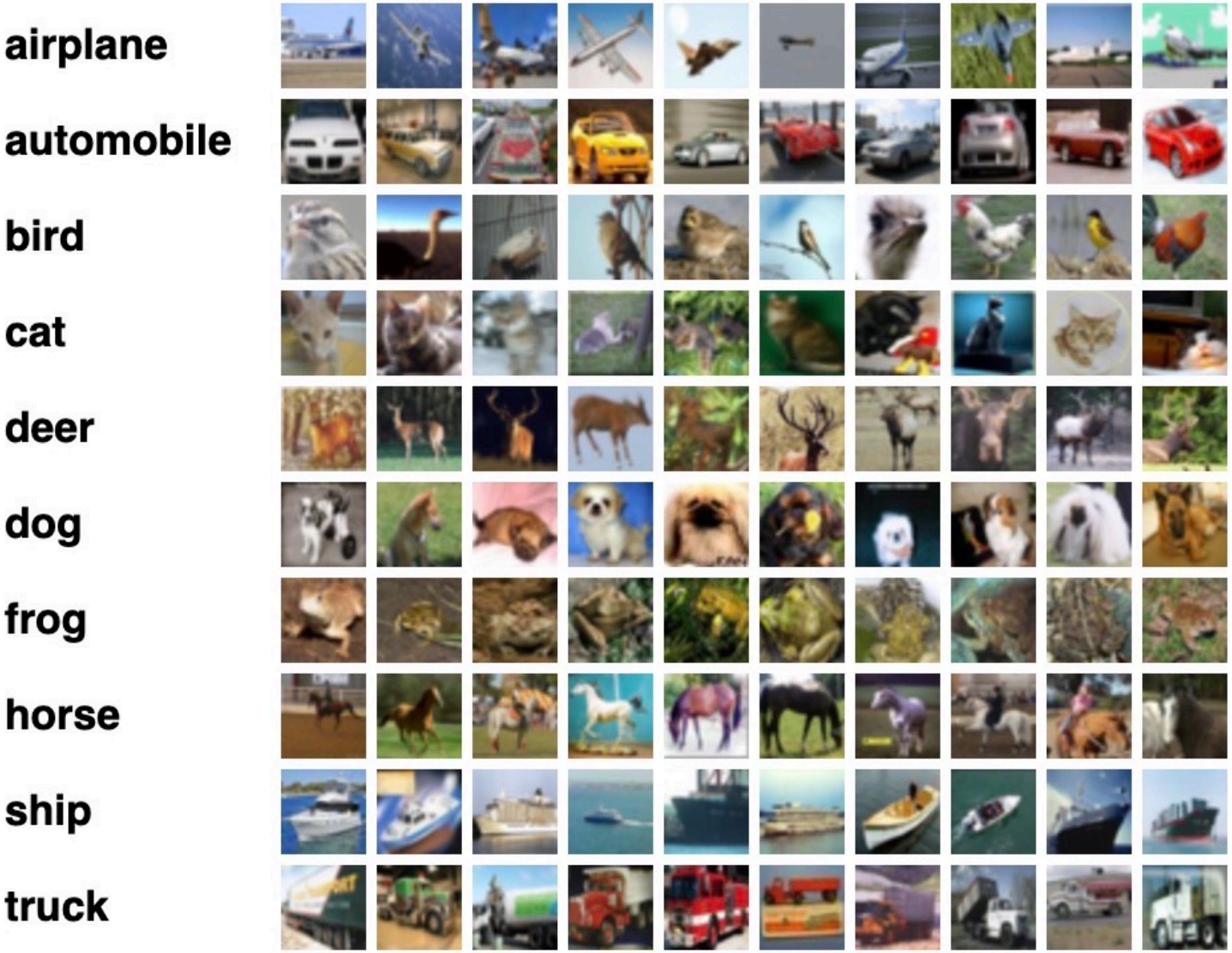
```
var isPaused = false;
// https://stackoverflow.com/questions/16554094/canvas-requestanimationframe-pause
window.requestAnimFrame = (function() {
    return (
        window.requestAnimationFrame ||
        window.webkitRequestAnimationFrame ||
        window.mozRequestAnimationFrame ||
        function(callback) {
            window.setTimeout(callback, 1000 / 60);
        }
    );
})();

// Отслеживаем нажатия клавишы
document.addEventListener("keydown", function(e) {
    // Если нажата клавиша ESC,
    if (e.which !== 27) return;
    console.log('Paused');
    isPaused = !isPaused;
    // Рисуем иконку паузы
    pauseDisplay();
});
```

Практическое задание

```
// Перепишем отрисовку как функцию для переиспользования
const redraw = () => {
    // Очищаем холст от предыдущего кадра
    context.clearRect(0, 0, canvas.width, canvas.height);

    // Рисуем содержимое заднего фона на холст
    context.drawImage(backgroundImg, 0, 0, backgroundImg.width, backgroundImg.height, 0, 0, canvas.width, canvas.height);

    // Рисуем левую ракетку на холсте
    context.drawImage(paddleImg, 0, 0, paddleImg.width, paddleImg.height, leftPaddle.x, leftPaddle.y, leftPaddle.width, leftPaddle.height);

    // Рисуем мячик
    context.drawImage(ballImg, 0, 0, ballImg.width, ballImg.height, ball.x, ball.y, ball.width, ball.height);

    // Рисуем правую ракетку на холсте
    context.drawImage(paddleImg, 0, 0, paddleImg.width, paddleImg.height, rightPaddle.x, rightPaddle.y, rightPaddle.width, rightPaddle.height);

    // Рисуем стены
    context.fillStyle = "lightgrey";
    context.fillRect(0, 0, canvas.width, grid);
    context.fillRect(0, canvas.height - grid, canvas.width, canvas.height);

    // Рисуем сетку посередине
    for (let i = grid; i < canvas.height - grid; i += grid * 2) {
        context.fillRect(canvas.width / 2 - grid / 2, i, grid / 2, grid / 2);
    };
};
```

Cifar100

- Набор данных, состоящий из цветных изображений 100 классов
- Размер 32 на 32 пикселя
- 3 цвета



ONNX

https://github.com/iu5git/MPPR

Step 3. Select class labels and get predictions

Выбрать файл cifar100_CNN.onnx Select ONNX file

Выбрать файл c25c94fe96_1000.jpg

Class label 54 Add 10 Random Undo Reset

0,50,54