

## Да научиш Javascript с играта PEXESO - PAIRS

### PEXESO - PAIRS

Тази страница описва играта Pairs (Двойки).

Играта **Pairs** е написана с помощта на **JavaScript** и стандартни **HTML** команди.

Играта е базирана на принципа на популярната компютърна игра "Двойки" или още "Чифтове". Задачата на играча е да открие две еднакви карти (или две карти с еднаква тема във версията "Pairs Mix"). Играчът може да щракне върху карта и да види картинката. Когато щракне върху две карти с еднакъв символ, картите остават отворени. Когато двете карти са различни, те се обръщат с гръб и скриват картинките.

Играта започва с щракване върху **Start/Начало** и може да бъде прекратена с щракване върху **Стоп/Край**.

Преди играта да започне, играчът може да настрои **нивото** (от 0 до 9). Нивото оказва влияние върху времето, през което картинките на картите са видими. По време на играта може да се следи изминалото време, броят на опитите и броят на откритите двойки.

И двете версии на играта показват теми, свързани с ИТ - свободен и платен софтуер. В едната версия играчът търси еднакви символи, а в другата - сходна функция на свободния и платения софтуер (напр. **Microsoft Word - Open Office Writer**).

Бутонът **Ръководство на учителя** извежда кратко описание на играта и нейното предназначение.

---

### Сваляне на играта

Играта може да бъде копирана и модифицирана според вашите собствени умения.

Щракване с десния бутон на мишката ще изведе меню, от което трябва да се избере опцията **View Source** (опцията може да фигурира в менюто и под различно име, според уеб браузъра, който се използва).

Щракване с десния бутон на мишката ще изведе меню, от което трябва да се избере "View Source". Този елемент от менюто може да е изписан по различен начин, в зависимост от това какъв браузър се използва. Текстът се маркира чрез стандартни съкратени команди (напр. Ctrl+A) и копиран в клипборда (напр. чрез Ctrl+C). Отваря се какъвто и да е текстов редактор, който може да обработва неформатиран текст (plain text) и в него се поставя копираният текст (Ctrl+V). Текстът в редактора се запазва във файл с произволно избрано име, но задължително с разширение htm или html - стандартните разширения, които интернет браузърите разпознават.

---

### Javascript код

Javascript кода се съхранява отделно във файла **pexeso.js**.

Файлът не може да се сваля по описания по-горе начин, но може да бъде отворен като в адресната лента на браузъра се извика <http://.../pexeso.htm> и се замени с <http://.../pexeso.js>.

## Да научиш Javascript с играта PEXESO - PAIRS

-->

Картинките представляват основна част на играта. Те са запазени в поддиректория, наречена **images**.

Списъкът на файловете с картинки:

### Общи картинки:

0.gif, 1.gif, 2.gif, 3.gif, 4.gif, 5.gif, 6.gif, 7.gif, 8.gif, 9.gif, 10.gif – изображенията на числата и фона

plusminus1.gif, plusminus2.gif, plusminus3.gif, plusminus4.gif – изображенията **плюс** и **минус**.

startstop1.gif, startstop2.gif, startstop3.gif, startstop4.gif – изображенията за бутоните **Start** и **Stop**.

attempts.gif, found.gif, level.gif, s.gif, time.gif – изображения с темите

blank.gif – фон

card.png – изображение на гърба на картите

### Кarti за версията **Pairs**

pic1.jpg, pic2.jpg, pic3.jpg, pic4.jpg, pic5.jpg, pic6.jpg, pic7.jpg, pic8.jpg – изображения на картите

### Кarti за версията **Pairs Mix**

pic1a.jpg, pic2a.jpg, pic3a.jpg, pic4a.jpg, pic5a.jpg, pic6a.jpg, pic7a.jpg, pic8a.jpg  
pic1b.jpg, pic2b.jpg, pic3b.jpg, pic4b.jpg, pic5b.jpg, pic6b.jpg, pic7b.jpg, pic8b.jpg – изображения на картите

Картинките могат да бъдат свалени с промяна на адреса в адресната лента на браузъра и се замени <http://.../pexeso.htm> (или [pexeso1.htm](http://.../pexeso1.htm)) с файл с поддиректорията и файла с картинка, например <http://.../images/0.gif>.

(пример: [www.ingot.org/javascript/domino/pexeso.htm](http://www.ingot.org/javascript/domino/pexeso.htm) [1] се заменя с [www.ingot.org/javascript/pexeso/images/0.gif](http://www.ingot.org/javascript/pexeso/images/0.gif) [2])

След натискането на клавиш Enter на клавиатурата ще се появи диалогов прозорец с въпрос за запазването на файла. Файлът трябва задължително да се запази със същото име и разширение и в същата директория като предходния файл.

Картинките трябва да се запазват в поддиректория **"images"**, която предварително трябва да се създаде (ръчно).

Правилното местонахождение и функциониране на свалените компоненти може да се провери чрез стартиране на локалния файл **pexeso.htm** (или **pexeso1.htm**).

---

### Замяна на картинки

Картинките в играта могат да бъдат заменени с ваши собствени, а фонът на основното поле на играта може да се променя, при следните условия:

- файловете с картинки трябва да носят същите имена и разширения като оригиналните картинки (по-опитните програмисти могат да използват различни файлове, като след

---

```
(function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){(i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)})(window,document,'script','/www.google-analytics.com/analytics.js','ga'); ga('create','UA-46896377-2','auto'); ga('send','pageview');
```

Page 2 of 3

- това модифицират изходния код);
- фонът представлява елемент, който се състои от множество повторения;
- фонът трябва да бъде прозрачен (описание как да се добави прозрачност можете да потърсите в интернет, това попада е извън обхвата на тази игра и инструкциите за възпроизвеждането/промяната й);
- картинките трябва да имат идентична ширина и височина, измерени в пиксели.

---

### Промяна на името на директорията и използване на различни файлови формати за изображенията

Ако се налага използването на различни имена на файлове с картинки или на име на директорията, в която се намират, е необходимо да се модифицира кодът чрез редактиране на файла **pexeso.htm**. Това действие е запазено за по-опитни програмисти.

Във файла **pexeso.js** трябва да се открие следният ред: **var IMG\_PATH = "images/";**

Изразът може да се промени с името на друга директория, в която са поставени файловете с изображения. Директорията трябва да е поддиректория на папката, в която се намират файловете **pexeso.htm** (или pexeso1.htm) и **pexeso.js**

Навсякъде в кода разширенията **gif** трябва да се заместят с разширението на новите файлове, например **png** или **jpg** (с изключение на фона, който трябва да бъде прозрачен).

---

### Кратко описание на Javascript функциите на играта

Както беше споменато по-горе, играта е написана с помощта на **JavaScript** и **HTML** команди. Кратко помощно описание на функциите и елементите, както и описание на променливите, се намират директно в кода (в коментирани редове). За да бъдат разбрани функциите, са необходими основни познания за Javascript и HTML.

HTML кодът за показване на графични и текстови елементи, линкове и бутони се намира във файла **pexeso.htm** (респективно pexeso1.htm)

---

### Javascript код:

Дефиниране на променливи:

```
// Game settings, check out pexeso.htm (line 8)
// <body bgcolor="#FFFF99" onload="loadImages(8, 0, 0);">
//                               | | |
//                               | | |-- highscore mode (js/php)
//                               | |---- game mode, see above
//                               |----- count of image pairs
//
```

Настройка на версията на играта:

За версия Pairs се избира режим (mode) 0.

## Да научиш Javascript с играта PEXESO - PAIRS

-->

(Настройката е във файла pexeso.htm, ред <BODY onload="loadImages(8, 0, 0);" >)

За версия Pairs Mix се избира режим (mode) 1.

(Настройката е във файла pexeso1.htm, ред <BODY onload="loadImages(8, 1, 0);" >)

```
// game mode 0 : match of two identical pictures,  
//      filenames pic1.jpg, pic2.jpg ...  
// game mode 1 : match of two similar pictures,  
//      filenames pic1a.jpg, pic1b.jpg, pic2a.jpg, pic2b.jpg ...  
//
```

Настройване на поддиректорията с картинките:

```
var IMG_PATH   = "images/";           // Path of the pictures  
var arrHighScore = new Array();       // Array of highscore objects  
var imgBackside = new Image();        // The backside of a card  
var imgArrStartStop = new Array(4);   // Start and stop button  
var imgArrPlusMinus = new Array(4);   // Plus and minus button  
var imgArrNumber = new Array(11);     // Ciphers 0 to 9  
var bRunning    = false;              // State of the game  
var nLevel      = 4;                  // Speed level  
var nSeconds    = 0;                  // Duration of the game in seconds  
var nAttempts   = 0;                  // Number of attempts  
var nHit        = 0;                  // Number of hits  
var nSelected1  = -1;                 // First selected picture  
var nSelected2  = -1;                 // Second selected picture  
var bShowCard   = false;              // State of the card (uncovered or not)  
var strPlayerName = "xxx";            // Name of player for highscore  
var strDate     = "";                 // Date for highscore  
var nPoints     = 0;                  // Points for highscore  
var bCookies    = false;              // Are cookies enabled  
var nHighscoreMode = 0;                // 0=Cookies 1=PHP  
  
var IMG_START, IMG_STOP, IMG_PLUS, IMG_MINUS;  
var IMG_LEVEL, IMG_SEC, IMG_ATTEMPTS, IMG_HIT  
var IMG_OFFSET, IMG_MODE, imgArrField, nImages, nSumImages;
```

Функция за качване на изображенията:

```
/*  
 * Preload of images  
 */  
  
function loadImages(nPairs, nImageMode, nScoreMode)  
{  
    nHighscoreMode = nScoreMode;  
    IMG_MODE = nImageMode;  
    nImages = nPairs;  
    imgArrField = new Array(nImages * 2);  
    nSumImages = nImages * 2 + 4 + 4 + 11;  
  
    IMG_START = 2 + nImages * 2;  
    IMG_STOP = 4 + nImages * 2;  
    IMG_PLUS = 12 + nImages * 2;  
    IMG_MINUS = 8 + nImages * 2;  
    IMG_LEVEL = 10 + nImages * 2;
```

```
IMG_SEC  = 17 + nImages * 2;
IMG_ATTEMPTS = 25 + nImages * 2;
IMG_HIT  = 33 + nImages * 2;

if(document.images)
{
    imgBackside.src = IMG_PATH + "card.png";

    // Load start and stop button
    for(var i = 0; i < 4; i++)
    {
        imgArrStartStop[i] = new Image();
        imgArrStartStop[i].src = IMG_PATH + "startstop" + (i + 1) + ".gif";
    }

    // Load plus and minus button
    for(var i = 0; i < 4; i++)
    {
        imgArrPlusMinus[i] = new Image();
        imgArrPlusMinus[i].src = IMG_PATH + "plusminus" + (i + 1) + ".gif";
    }

    // Load ciphers
    for(var i = 0; i < 11; i++)
    {
        imgArrNumber[i] = new Image();
        imgArrNumber[i].src = IMG_PATH + i + ".gif";
    }

    // Load pictures
    for(var i = 0; i < nImages; i++)
    {
        if(IMG_MODE == 1)
        {
            imgArrField[i * 2] = new Image();
            imgArrField[i * 2].src = IMG_PATH + "pic" + (i + 1) + "a.jpg";
            imgArrField[i * 2 + 1] = new Image();
            imgArrField[i * 2 + 1].src = IMG_PATH + "pic" + (i + 1) + "b.jpg";
        }
        else
        {
            IMG_MODE = 0;
            imgArrField[i * 2] = new Image();
            imgArrField[i * 2].src = IMG_PATH + "pic" + (i + 1) + ".jpg";
            imgArrField[i * 2 + 1] = new Image();
            imgArrField[i * 2 + 1].src = IMG_PATH + "pic" + (i + 1) + ".jpg";
        }
    }
}
```

```
nLevel = 4;
nSeconds = 0;
nAttempts = 0;
nHit     = 0;
searchFirstImage();
clearField();
updateAll();
setTimeout("checkLoading()", 1000)
```

```
}  
}
```

Function for shuffling the cards:

```
/*  
 * Shuffle all pictures  
 */  
  
function shuffle()  
{  
  if(document.images)  
  {  
  
    // Swap two pictures (random index)  
    var j = Math.floor(new Date().getSeconds() * Math.random() + 60);  
    for(var i = 0; i < j; i++)  
    {  
      n1 = Math.round(Math.random() * (nImages * 2 - 1));  
      n2 = Math.round(Math.random() * (nImages * 2 - 1));  
      img = imgArrField[n1];  
      imgArrField[n1] = imgArrField[n2];  
      imgArrField[n2] = img;  
    }  
  }  
}
```

Start game:

```
/*  
 * Start new game  
 */  
  
function startGame()  
{  
  if(document.images)  
  {  
    if(!bRunning)  
    {  
      shuffle();  
      clearField();  
      nSeconds = 0;  
      nSelected1 = -1;  
      nSelected2 = -1;  
      nAttempts = 0;  
      nHit = 0;  
      id = setInterval("countSeconds()", 1000)  
      bRunning = true;  
      bShowCard = false;  
      updateAll();  
    }  
  }  
}
```

Край на играта:

```
/*
 * Stop game
 */

function stopGame()
{
  if(document.images)
  {
    if(bRunning)
    {
      clearInterval(id);
      bRunning = false;
      updateAll();
    }
  }
  return;
}
```

Измерване на времето:

```
/*
 * Count seconds
 */

function countSeconds()
{
  nSeconds++;
  showNumber(nSeconds, IMG_SEC + IMG_OFFSET, 5);
}
```

Function for figures display:

```
/*
 * Show numbers
 */

function showNumber(nNumber, nPosition, nCount)
{
  if(document.images)
  {
    nNumber += "";
    while(nNumber.length < nCount) nNumber = " " + nNumber;
    for(var i = 0; i < nCount; i++)
    {
      var n = nNumber.charAt(i);
      if(n == " ")
      {
        document.images[nPosition + i].src = imgArrNumber[10].src;
      }
      else
      {
        document.images[nPosition + i].src = imgArrNumber[n].src;
      }
    }
  }
}
```

Функция за изобразяване на гърба на картите:

```
/*
 * Set the backside of all cards
 */

function clearField()

{
  if(document.images)
  {
    for(var i = 0; i < nImages * 2; i++)
    {
      document.images[i + IMG_OFFSET].src = imgBackside.src;
    }
  }
}
```

Функция за показване на избраните карти:

```
/*
 * Show card
 */

function showCard(nImage)

{
  if(document.images)
  {
    if(bRunning && !bShowCard)
    {

      // Uncover a card, if not allready two are selected
      if(nSelected1 == -1 || nSelected2 == -1)
      {
        // Is the clicked not uncovered ?
        if(document.images[nImage + IMG_OFFSET].src == imgBackside.src)
        {

          // Uncover selected card
          document.images[nImage + IMG_OFFSET].src = imgArrField[nImage].src;
          if(nSelected1 == -1)
          {
            nSelected1 = nImage;
          }
          else
          {
            nSelected2 = nImage;
          }
        }
      }
    }

    // There are two cards uncovered
    if(nSelected1 != -1 && nSelected2 != -1)
```

```
{
  showNumber(++nAttempts, IMG_ATTEMPTS + IMG_OFFSET, 5);

  // Are the cards equal ?
  var pic1 = document.images[nSelected1 + IMG_OFFSET].src;
  var len1 = pic1.length;
  var pic2 = document.images[nSelected2 + IMG_OFFSET].src;
  var len2 = pic2.length;
  if(pic1.substr(0, len1 - IMG_MODE - 4) == pic2.substr(0, len2 - IMG_MODE - 4))

  {
    // If there are equal, incement the counter
    showNumber(++nHit, IMG_HIT + IMG_OFFSET, 5);
    nSelected1 = -1;
    nSelected2 = -1;

    // Are all pictures uncovered ?
    if(nHit == nImages)
    {
      stopGame();
      nPoints = Math.round(100000 * (nLevel + 1) / nSeconds / nAttempts);
      strMsg = "Tvoje skřre je " + nPoints + " bodř.";
      window.alert(strMsg);
    }
  }
  else
  {
    // The two cards are not identical, start now the timer for hide the cards
    bShowCard = true;
    setTimeout("clearCard()", 2000 - nLevel * 200);
  }
}
else
{
  if(!bRunning)
  {
    alert("Press key Start !");
  }
}
}
```

Функция за обръщане на различаваща се двойка карти:

```
/*
 * Turn the uncovered cards
 */
```

```
function clearCard()

{
  document.images[nSelected1 + IMG_OFFSET].src = imgBackside.src;
  document.images[nSelected2 + IMG_OFFSET].src = imgBackside.src;
```

```
nSelected1 = -1;
nSelected2 = -1;
bShowCard = false;
}
```

Функция за настройка на нивото:

```
/*
 * Set the level
 */

function setLevel(nValue)

{
  if(document.images && !bRunning)
  {
    nLevel += nValue;
    if(nLevel < 0) nLevel = 0;
    if(nLevel > 9) nLevel = 9;
    showNumber(nLevel, IMG_LEVEL + IMG_OFFSET, 1);

  }
}
```

Ъпдейт на броячите:

```
/*
 * Update all counter
 */

function updateAll()

{
  if(document.images)
  {

    showNumber(nLevel, IMG_LEVEL + IMG_OFFSET, 1);
    showNumber(nSeconds, IMG_SEC + IMG_OFFSET, 5);
    showNumber(nAttempts, IMG_ATTEMPTS + IMG_OFFSET, 5);
    showNumber(nHit, IMG_HIT + IMG_OFFSET, 5);
    if(bRunning)
    {
      document.images[IMG_START + IMG_OFFSET].src = imgArrStartStop[1].src;
      document.images[IMG_STOP + IMG_OFFSET].src = imgArrStartStop[2].src;
      document.images[IMG_PLUS + IMG_OFFSET].src = imgArrPlusMinus[1].src;
      document.images[IMG_MINUS + IMG_OFFSET].src = imgArrPlusMinus[3].src;
    }
    else
    {

      document.images[IMG_START + IMG_OFFSET].src = imgArrStartStop[0].src;
      document.images[IMG_STOP + IMG_OFFSET].src = imgArrStartStop[3].src;
      document.images[IMG_PLUS + IMG_OFFSET].src = imgArrPlusMinus[0].src;
      document.images[IMG_MINUS + IMG_OFFSET].src = imgArrPlusMinus[2].src;
    }
  }
}
```

```
}  
}
```

Премахване на празни пространства в думата:

```
/*  
 * Remove all white spaces  
 */  
  
function strTrim(str)  
{  
    var strReturn = "";  
  
    for(var i = 0; i < str.length; i++)  
    {  
        if(str.charAt(i) != " ")  
        {  
            strReturn += str.charAt(i);  
        }  
    }  
    return strReturn;  
}
```

Първоначално нулиране:

```
/*  
 * Add leading zero  
 */  
  
function addLeadingZero(value, nTotalLength)  
{  
    value += "";  
    while(value.length < nTotalLength) value = "0" + value;  
    return value;  
}
```

Търсене на индекса на първите карти:

```
/*  
 * Search for index of first image  
 */  
  
function searchFirstImage()  
{  
    for(var i = 0; i < document.images.length; i++)  
    {  
        if(document.images[i].name == "memory_id")  
        {  
            IMG_OFFSET = i + 1;  
            break;  
        }  
    }  
}
```

Брояч на изиграните карти:

```
/*
 * Count of preloaded images
 */

function countLoadedImages()
{
    var nCompleted = 0;
    for(var i = 0; i < 2 * nImages; i++)
    {
        if(imgArrField[i].complete)
        {
            nCompleted++;
        } //else alert(imgArrField[i].src);
    }

    for(var i = 0; i < 4; i++)
    {
        if(imgArrStartStop[i].complete)
        {
            nCompleted++;
        } //else alert(imgArrStartStop[i].src);
    }

    for(var i = 0; i < 4; i++)
    {
        if(imgArrPlusMinus[i].complete)
        {
            nCompleted++;
        } //else alert(imgArrPlusMinus[i].src);
    }

    for(var i = 0; i < 11; i++)
    {
        if(imgArrNumber[i].complete)
        {
            nCompleted++;
        } //else alert(imgArrNumber[i].src);
    }
    return nCompleted;
}
```

Индикиране на зареждането на картинки:

```
/*
 * Progressbar (state of preloading images)
 */

function checkLoading()
{
    var nLoaded = countLoadedImages();
    if(nLoaded < nSumImages)
    {
        document.images[IMG_OFFSET - 1].width = 456 - (456 / nSumImages * nLoaded);
        window.status = "" + nLoaded + " of " + nSumImages + " pictures loaded";
        setTimeout("checkLoading()", 250);
    }
    else

```

## Да научиш Javascript с играта PEXESO - PAIRS

-->

---

```
{
  document.images[IMG_OFFSET - 1].src = IMG_PATH + "blank.gif";
  document.images[IMG_OFFSET - 1].width = 5;
  window.status = "";
  updateAll();
}
}
```

**Source URL:** <https://theingots.org/community/node/26775>

### Links

- [1] <http://www.ingot.org/javascript/domino/pexeso.htm>
- [2] <http://www.ingot.org/javascript/pexeso/images/0.gif>