

Laboratorium 7

PHP dysponuje grupą funkcji, które pozwalają otwierać, wyświetlać i manipulować obrazami w przeglądarkach WWW. Poniżej zostały omówione niektóre z tych funkcji:

imagestring	umieszcza tekst na rysunku, identyfikator tekstu jest pierwszym argumentem; następne to: numer czcionki, współrzędne x i y, od których rozpoczyna się pisanie, tekst to wypisania i jego kolor
imageinterface	wskazuje, czy rysunek określony pierwszym parametrem ma być wyświetlony z przeplotem; jeżeli tak, drugi argument powinien mieć wartość TRUE
imagecolortransparent	dla rysunku określonego pierwszym argumentem ustala kolor, który będzie przezroczysty (stanowi on drugi argument)
createimagefromgif	funkcja zwraca zmienną określającą – rysunek zawarty w pliku, którego nazwa stanowi argument funkcji
imagesize	funkcja zwraca szerokość rysunku, określonego przez identyfikator – parametr
imageheight	funkcja zwraca wysokość rysunku, określonego przez identyfikator – parametr
getimagesize	funkcja zwraca tablicę, zawierającą cztery pola: szerokość rysunku, jego wysokość, typ oraz łańcuch znaków do wykorzystania w znaczniku IMG

Ćwiczenie 7.1.

Utwórz program, który przygotuje i wyświetli ilustrację w formacie JPG, przy założeniu, że będzie to biała kratka na czarnym tle, o rozmiarach 100x100.

W ćwiczeniu użyto następujących funkcji:

header	przesyła nagłówek, określany jako parametr
imagecreate	tworzy rysunek, parametry określają jego rozmiar w pikselach a funkcja zwraca zmienną, będącą jego identyfikatorem
imagecolorallocate	definiowanie koloru dla rysunku, kolejnymi argumentami są: zmienna, będąca identyfikatorem rysunku oraz składowe koloru: R, G i B (od 0 do 255). Funkcja zwraca identyfikator koloru.
imagefill	wypełnienie rysunku (identyfikator jest pierwszym argumentem) kolorem stanowiącym czwarty argument. Drugi i trzeci argument to współrzędne, od których rozpoczyna się wypełnianie obszaru (jako obszar rozumiane są piksele jednego koloru, a jakiegokolwiek inny kolor stanowi brzeg)
imageline	w rysunku, którego identyfikatorem jest pierwszym argumentem, rysuje linię o współrzędnych początku i końca podanych w kolejnych czterech parametrach; kolor linii jest określony szóstym parametrem.
imagejpg	wyświetlenie ilustracji w formacie jpg; argument funkcji jest identyfikatorem ilustracji
imagegif	wyświetlenie ilustracji w formacie gif argument funkcji jest identyfikatorem ilustracji
imagepng	wyświetlenie ilustracji w formacie png argument funkcji jest identyfikatorem ilustracji

Całość programu przedstawia się następująco:

```
//Listing 7-1.php
<?
// Program tworzy ilustrację z białą kratką na czarnym tle.
header("Content-type: image/jpeg");
$rysunek = imagecreate (100,100);
$kolorbialy = imagecolorallocate ($rysunek, 255, 255, 255);
$kolorczerwony = imagecolorallocate ($rysunek, 0, 0, 0);
```

```
imagefill ($rysunek, 0, 0, $kolorczarny);
for ($i=1; $i<10; $i++) {
    imageline ($rysunek, 10*$i, 0, 10*$i, 100, $kolorbialy);
    imageline ($rysunek, 0, 10*$i, 100, 10*$i, $kolorbialy);
}
imagejpeg ($rysunek);
?>
```

Efektem działania jest rysunek w formacie jpg o wymiarach 100x100, przedstawiający szachownicę.

Ćwiczenie 7.2.

Utwórz podobną szachownicę jak w ćwiczeniu 7.1. Grafikę wyświetl w formacie GIF, a każde pole szachownicy wypełnij losowym kolorem.

W pętli, dla każdego ze 100 pól szachownicy, będziemy losować kolor. Jeżeli chcemy, by wartość każdej ze składowych zawierała się w przedziale 0-255, wywołanie funkcji powinno mieć następującą postać:

```
imagecolorallocate($rysunek, rand()%256, rand()%256, rand()%256);
```

Nie należy zapominać o zainicjowaniu generatora liczb pseudolosowych funkcją **srand**.

```
//Listing 7-2.php
<?
// Program tworzy ilustrację z białą kratką. Każde pole
// jest wypełnione losowym kolorem.
header("Content-type: image/gif");
$rysunek = imagecreate (100,100);
$kolorbialy = imagecolorallocate ($rysunek, 255, 255, 255);
$kolorczarny = imagecolorallocate ($rysunek, 0, 0, 0);
imagefill ($rysunek, 0, 0, $kolorczarny);

for ($i=1; $i<10; $i++) {
    imageline ($rysunek, 10*$i, 0, 10*$i, 100, $kolorbialy);
    imageline ($rysunek, 0, 10*$i, 100, 10*$i, $kolorbialy);
}

srand(time());
for ($x=0; $x<10; $x++) {
    for ($y=0; $y<10; $y++) {
        $kolorlosowy = imagecolorallocate
            ($rysunek, rand()%256, rand()%256, rand()%256);
        imagefill ($rysunek, 5+$x*10, 5+$y*10, $kolorlosowy);
    }
}
imagegif ($rysunek);
```

?>

Tworząc własne grafiki przy pomocy modułu **GD**, można kontrolować kolor każdego piksela (pamiętając o ograniczenia każdego z formatów graficznych).

Ćwiczenie 7.3.

Napisz program, który w rysunku w formacie **GIF** o wymiarach 100x100 wylosuje i zakoloruje 300 punktów czerwonych i 300 czarnych (punkty w kolejnych losowaniach mogą się powtarzać).

Została wykorzystana funkcja, która określa kolor pojedynczego piksela.

imagepixel	dla rysunku określonego przez pierwszy argument definiuje kolor piksela o współrzędnych podanych przez drugi i trzeci argument. Czwarty argument określa kolor.
-------------------	---

```
//Listing 7-3.php
<?
// Program losuje 300 punktów czerwonych i czarnych i zaznacza je.
header("Content-type: image/gif");
$rysunek = imagecreate (100,100);
$kolorbialy = imagecolorallocate ($rysunek, 255, 255, 255);
$kolorczerwony = imagecolorallocate ($rysunek, 255, 0, 0);
$kolorczarny = imagecolorallocate ($rysunek, 0, 0, 0);

imagefill ($rysunek, 0, 0, $kolorbialy);
srand(time());

for ($i=1; $i<=300; $i++)
{
    imagepixel ($rysunek, rand()%100-1, rand()%100-1, $kolorczerwony);
    imagepixel ($rysunek, rand()%100-1, rand()%100-1, $kolorczarny);
}
imagegif ($rysunek);
?>
```

Ćwiczenie 7.4.

Napisz program, który w rysunku w formacie **GIF** o wymiarach 100x100 narysuje pięć okręgów czerwonych i pięć czarnych w losowych miejscach.

Nie ma funkcji rysującej okrąg, ale można skorzystać z bardziej ogólnej, która rysuje wycinek elipsy.

imagearc	rysuje wycinek elipsy; kolejnymi argumentami są: identyfikator rysunku, współrzędna x środka łuku, współrzędna y środka łuku, szerokość łuku, wysokość łuku, kąt początku, kąt końca, kolor.
-----------------	--

**INSTYTUT TECHNIKI
ZAKŁAD ELEKTROTECHNIKI I INFORMATYKI**

mdymek@univ.rzeszow.pl

Jeżeli w wywołaniu funkcji szerokość i wysokość będą sobie równe, a łuk będzie pełen (od 0 do 360 stopni), to rysowany łuk okaże się okręgiem.

```
//Listing 7-4.php
<?
    // Program tworzy ilustrację z pięcioma czerwonymi i pięcioma
    // czarnymi okręgami.
    header("Content-type: image/gif");
    $rysunek = imagecreate (100,100);
    $kolorbialy = imagecolorallocate ($rysunek, 255, 255, 255);
    $kolorczerwony = imagecolorallocate ($rysunek, 255, 0, 0);
    $kolorczarny = imagecolorallocate ($rysunek, 0, 0, 0);
    imagefill ($rysunek, 0, 0, $kolorbialy);
    srand(time());

    for ($i=1; $i<=5; $i++) {
        imagearc ($rysunek, rand()%100-1, rand()%100-1,
            20, 20, 0, 360, $kolorczerwony);
        imagearc ($rysunek, rand()%100-1, rand()%100-1,
            20, 20, 0, 360, $kolorczarny);
    }
    imagegif ($rysunek);
?>
```

Ćwiczenie 7.5.

Napisz program, który wyświetli formularz pozwalający wpisać liczbę. Po jej wpisaniu wyświetli graficzną prezentację wpisaną liczbę.

Liczbę należy konstruować z poszczególnych cyfr. W katalogu cyfry należy umieścić wszystkie cyfry w formacie **GIF** o rozmiarze 15x20 pikseli.

Wykorzystane są tutaj trzy nowe funkcje:

imagecreatetromgif	tworzy rysunek na podstawie już istniejącego; argumentem jest nazwa pliku; funkcja zwraca identyfikator utworzonego rysunku
imagecopyresized	kopiuje do rysunku określonego przez pierwszy argument inny – określony przez drugi; następne argumenty określają: współrzędne x i y miejsca, w którym należy rysunek umieścić, współrzędne x i y oraz rozmiar dx i dy w rysunku docelowym i wielkość ddx i ddy w rysunku źródłowego (pozwalające na przeskalowanie)
imagedestroy	likwiduje identyfikator rysunku, jednocześnie zwalniając pamięć z nim związaną

```
//Listing 7-5.php
<?
    // Program wyświetla formularz, pozwalający wpisać liczbę.
    //Jeżeli liczba jest wpisana, wyświetla ją w postaci graficznej
    function printliczba ($numer) {
        // funkcja wyświetla liczbę w postaci grafiki, korzystając
        // z grafik - cyfr, zawartych w katalogu cyfry.
```

**INSTYTUT TECHNIKI
ZAKŁAD ELEKTROTECHNIKI I INFORMATYKI**

mdymek@univ.rzeszow.pl

```
$liczbacyfr=1; $l = $numer;
while ($l >= 10) {
    $liczbacyfr++;
    $l=floor($l/10);
}
header("Content-type: image/gif");
$rysunek = imagecreate (15*$liczbacyfr,20);
for ($i=$liczbacyfr; $i>=1; $i--) {
    $cyfra = floor($numer/pow(10,$i-1));
    $numer = $numer % pow(10,$i-1);
    $rysunekcyfra = imagecreatefromgif ("cyfry/$cyfra.gif");
    imagecopyresized
        ($rysunek,$rysunekcyfra,($liczbacyfr-$i)*15,0,0,0,15,20,15,20);
    imagedestroy ($rysunekcyfra);
}
imagegif ($rysunek);
}
if ($liczba>0) { # jest wpisana jakaś wartość w formularzu
    printliczba ($liczba);
} else { # nie ma wpisanych danych, wyświetlamy formularz
    print '<HTML>';
    print '    <HEAD>';
    print '        <META HTTP-EQUIV="Content-Type" CONTENT="text/html; ' ;
    print '        charset=iso-8859-2">';
    print '        <TITLE>Liczba graficznie</TITLE>';
    print '    </HEAD>';
    print '    <BODY>';
    print '        <FORM ACTION="5-05.php" METHOD=GET>';
    print '        <INPUT TYPE="text" NAME="liczba">';
    print '        <INPUT TYPE="submit" VALUE="Wyślij">';
    print '    </FORM>';
    print '    </BODY>';
    print '</HTML>';
}
?>
```

Do graficznej prezentacji danych nieoceniona jest funkcja, która rysuje wypełniony prostokąt.

imagefilledrectangle	rysuje wypełniony prostokąt; argumentami są: identyfikator rysunku, współrzędne x i y lewego górnego rogu, współrzędne x i y prawego dolnego rogu oraz kolor
-----------------------------	--

Zadania

1. Napisz skrypt, który wypełni tablicę dziesięcioma losowymi liczbami z zakresu 0-9. Następnie zilustruj wylosowane wartości na wykresie słupkowym.
2. Napisz skrypt, który utworzy ilustrację (o rozmiarach 80x600). Zapełnij ją wieloma różnokolorowymi kwadratami. Narysuj też linię i wypisz jakiś tekst. Wyświetl ilustrację z przeplotem.
3. Napisz skrypt, który utworzy ilustrację z czerwonym kółkiem na białym tle. Spraw, by biały kolor był przezroczysty. Wyświetl ilustrację na stronie HTML, która posiada tło innego niż biały koloru.
4. Dla istniejącego rysunku wyświetl go w dokumencie HTML, automatycznie określając jego szerokość i wysokość, a także nadając atrybuty WIDTH i HEIGHT.
5. Napisz skrypt, który przeskalowuje ilustrację. Porównaj jakość takiej ilustracji z przeskalowaną za pomocą programu graficznego.