



КОМПЮТЪРНО УЧИЛИЩЕ

PC SCHOOL

ПРОЧЕТИ & РАЗБЕРИ

Брой

4

вашият помощник в света на компютрите

Photoshop

Монтаж на изображение

Fireworks MX

Шоколадов текст

PHP

Запознаване с променливите

CSS

Заоблени ъгли

Java

Програмиране в нова светлина

Съдържание:

Photoshop - Монтаж на изображение	3
Photoshop - Обработка на изображения	6
Photoshop - Записване на изображения за Web	9
Photoshop - Златен текст	10
Photoshop - Кървав текст	12
Flash MX - Flash на цял екран	14
Fireworks MX - Шоколадов текст	16
Fireworks MX - Рисувана рамка	17
PHP - Запознаване с променливите	19
PHP - Оператори 1 част	22
PHP - Динамична промяна на типове 1 част	26
CSS - Заоблени ъгли	29
Java - Програмиране в нова светлина	31

Автори в този брой:

The ONE

www.pcschool.info
theone86@abv.bg

Росен Иванов

www.photoshopbg.com
info@photoshopbg.com

**Георги Гърдев
(GeorgeSG)**

georgesg.atspace.com
gogogardev@gmail.com

**Любомир Стефанов
(Shyne)**

www4u.search.bg
hack_shyne@gbg.bg

IvO_b0y

www.ivo.stih4e.com
fragless@gmail.com

**Стилиян Бешев
(Diabolic_Soul)**

dib23@abv.bg

Ако сте с по-стара или с по-нова версия на определена програма, недейте да пренебрегвате статиите, тъй като в повечето случай различните версии си приличат.

PC School не носи отговорност за изменения в поставените от нас линкове!

Всички права върху статиите принадлежат на авторите!

Photoshop - Монтаж на изображение


(Adobe **Photoshop 7**)

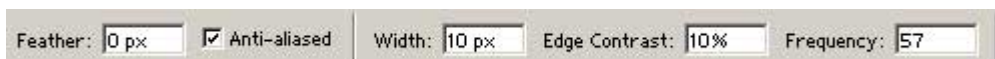
В този урок вие ще се научите да правите монтаж на изображение. Предимно всички си мислят, че това отнема доста време и е много сложно. Сега ще ви докажа, че всичко това не е вярно. Нека да започваме :).

Като за начало си свалете файловете за работа от:

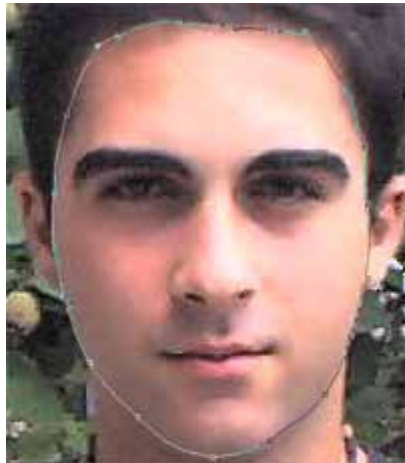
<http://photoshopbg.com/online%20education3/BG/Lessons/For%20advanced/lesson3/lesson3.rar>

И ги отворете във Photoshop за да ги разгледате. Отворете и final.psd, за да видите крайния ефект, който ще трябва да получите.

Оставете активен само файла 1.psd, защото ще работите с него. Активирайте инструмента Magnet Lasso Tool , който е под Lasso Tool, от кутията с инструменти. В менюто с опции за инструмента въведете същите стойности, които са по - долу на картинката.



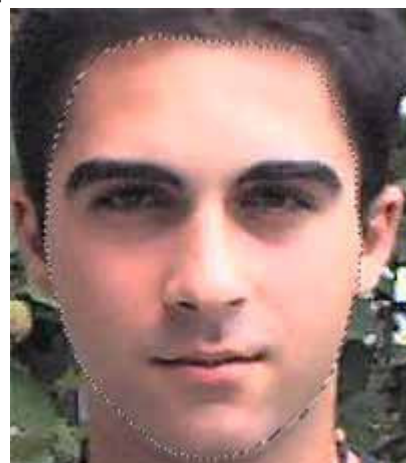
Вашата задача ще е да отрежете лицето на момчето. За тази цел с инструмента очертавате лицето много внимателно. Ако искате си увеличете снимката с инструмента Zoom за да ви е по - лесно да очертаете обекта. Резултата трябва да е нещо такова :



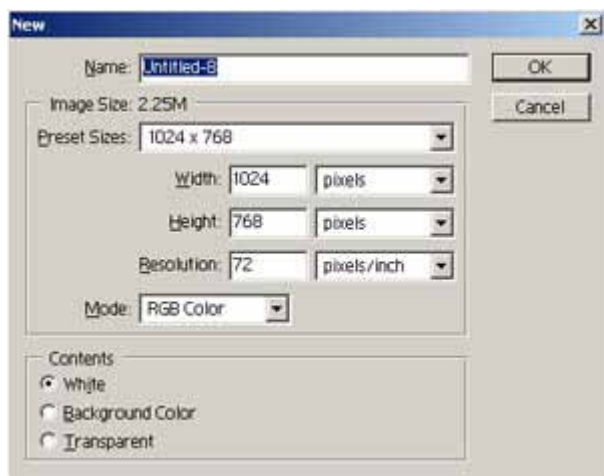
< Резултат

Вече сте направили стъпката по изрязването. Следва да съедините двата края на очертанието, което сте направили, за да се получи една селекция. Ето резултата

Резултат >

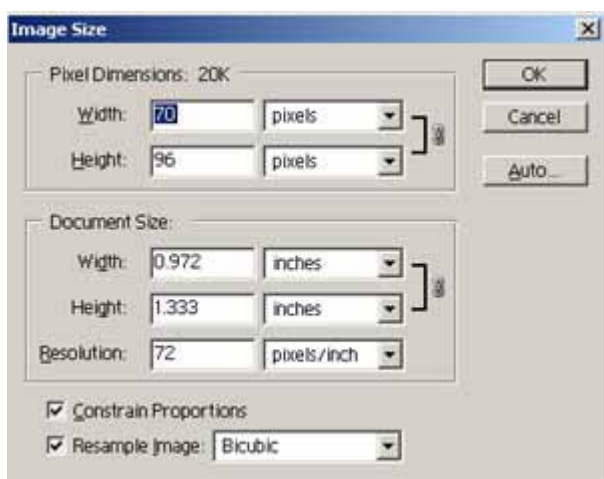


Направената селекция я копирайте и я прехвърлете в нов файл. Това ставаше от Edit > Copy и после File > New и натискате ОК, в прозореца, който ви се появява. Ето и пример



Когато новият файл е създаден от менюто Edit изберете Paste. Изрязаното изображение, ще се прехвърли автоматично.

Следващото нещо, което ще направите е да го смалите с Image Size. От менюто Image изберете Image Size.



Това е прозореца, който трябва да се отвори. Въведете същите стойности в Width и Height, които са на картинката по - горе и натиснете ОК. Резултата трябва да е такъв:



< Резултат

Ще трябва сега да завъртите изображението с 3 градуса на ляво. От менюто Image изберете Rotate Canvas и от там Arbitrary. Въведете стойността 3 и изберете CCW. Натиснете ОК и изображението ще се завърти.

Сега прехвърлете смаленото и изрязано лице на файла 2.psd. За тази цел използвайте инструмента Move. Както в предишните уроци, чрез влачене прехвърлете изображението в

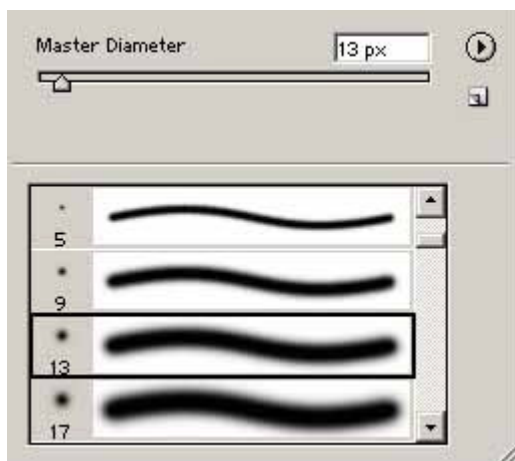
файла 2.psd. Можете и да го селектирате цялото (Ctrl + A) и да го копирате и пействате във файла (Ctrl + C и после Ctrl + V). Резултата трябва да е такъв.



< Резултат

Ако сте стигнали до тук това е много добре, ако ли не се върнете отначало и четете внимателно.

Следва другата стъпка. Да пооправите новото лице, така че да пасне добре на старото. Това можете да направите по най - лесния начин. С инструмента Eraser (гумичка). Активирайте го и изберете мека четка от менюто с опции за инструмента.



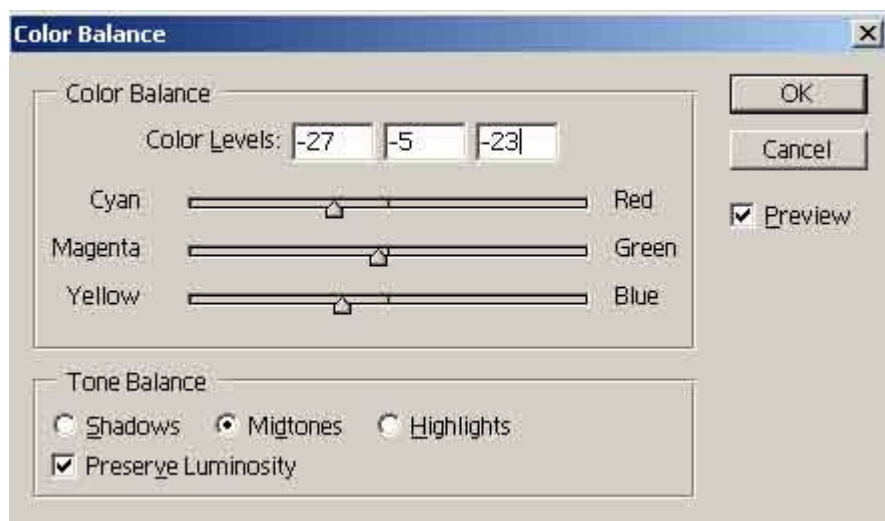
Използвайте, ако искате Zoom инструмента за да ви е по - лесно да изтриете ненужните части. След като сте направили всички тези неща престъпете към работа.

Можете да използвате и инструмента Smudge Tool, който е под Blur Tool за да раздърпате лицето на местата, където е нужно. Резултата трябва да е такъв:



Това е резултата, който се получава след като обработите изображението с гумичката.

Последната стъпка е да оправите цветовете на лицето за да съвпадат с цвета на тялото. Ето и какво трябва да направите. От менюто Image изберете Adjustment и от там Color Balance. Въведете същите стойности които и аз съм въвел отдолу на снимката.



Крайният резултат след всички обработвания трябва да е такъв:



Това е, което и вие трябва да сте постигнали.

В този урок вие се научихте как да правите монтаж, как да обработвате снимка за монтаж, как да поправяте цветовете и как да работите с Smudge Tool.

Автор: Росен Иванов

Photoshop - Обработка на изображения

(Adobe Photoshop 7)

В този урок ще научите как да обработвате снимки или изображения. Да поправяте цветовете, светлина, контраст и още много други неща. Нека да започваме.

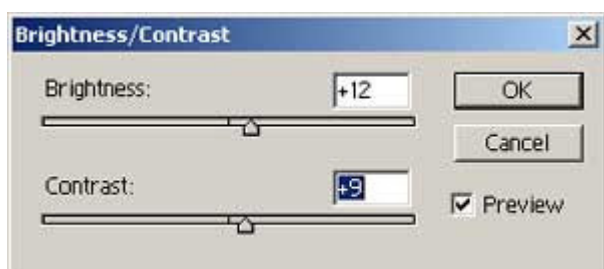
Като за начало си изтеглете двата файла за този урок от:

<http://photoshopbg.com/online%20education3/BG/Lessons/For%20advanced/lesson1/lesson1.rar>

Отворете двата файла във Photoshop и ги разгледайте. start.psd е файлът, с който ще работите. Final.psd показва финалния резултат, след обработката.

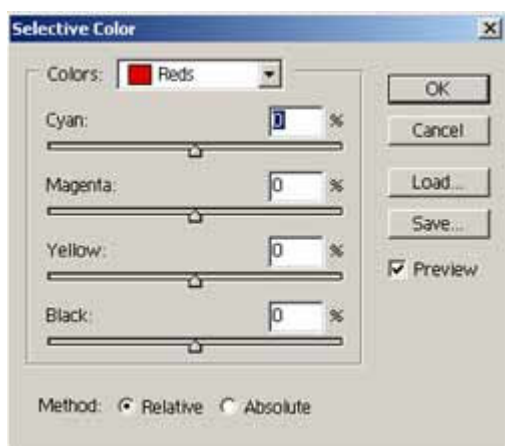
Затворете final.psd и оставете само другия файл активен. Като за начало ще трябва да пооправите цветовете с автоматичните настройки. Това става по следния начин. От менюто Image изберете Adjustments и после изберете Auto Level. След използването на auto level ще настъпят промени в цветовете, и осветяването. Следващата стъпка е да оправите само цветовете с автоматичните настройки. Това става по следния начин. От менюто Image изберете пак Adjustment и изберете Auto Color.

Следващата стъпка е да поправим и осветяването и контраста на снимката. От менюто Image изберете Adjustment и Brightness/Contrast



Въведете същите стойности каквито са на снимката и натиснете ОК. Снимката ще стане по - светла и малко по - контрастна.

Следва да оправите цветовете малко по - задълбочено, за да не са толкова ярки. Вземете за пример лицето на момчето. То е доста червено, а не трябва да е така. Ще използваме за тази цел Selective Color. Това се намира в Image > Adjustment > Selective Color




Това е менюто, което трябва да ви се отвори, след като изберете Selective Color. От тук сега ще оправите цветовете да не са толкова наситени. В Cyan въведете стойността +43 %, в Magenta въведете стойността -33 %, в Yellow въведете стойността -43 % и накрая в Black въведете -6 %. Оставете метода на прилагане да е Relative. Ако ви е изключена опцията preview, можете да я активирате, за да виждате какви промени настъпват преди да запазите настройките. Когато сте готови натиснете ОК. Вече снимката трябва да изглежда доста по - добре от преди. Следващата стъпка е да оправите цветовете от различните канали. Активирайте палитрата Channels. Ако не е видима изберете Window > Channels за да я активирате.



Това е което трябва да ви се покаже, след като я активирате. Главния проблем при цветовете е тяхната контрастност и наситеност. Сега вие ще трябва да оправите тези проблеми. Изберете канала Red. Снимката ще стане черно - бяла, но това си е напълно нормално. Изберете от менюто Image > Adjustment > Brightness/Contrast и въведете в полето Brightness -7 и натиснете OK когато сте готови. Активирайте канала Green, от палитрата с канали. Изберете пак Image > Adjustment > Brightness/Contrast и въведете в Brightness -12. Накрая изберете и канала Blue и пак отидете в менюто Brightness/Contrast и въведете в Brightness стойността -10. Натиснете OK и активирайте RGB канала, от палитрата с каналите, за да видите изображението вече цветно.

Това беше обработката на цветовете, контраста и светлината. Следва да се замъглят и изострят някои контури от изображението.

Като за начало активирайте инструмента Blur Tool , с който ще замъглявате. Сега изберете размер на четката какъвто на вас ви е удобен, от менюто с опции за инструмента. Изберете и силата с която ще замъглявате.



На снимката по - горе е показано колко трябва да е силата на замъгляване. Използвайте инструмента Zoom (лупа) за да увеличите изображението, за да ви е по - лесно. За да го смалявате, след като сте го увеличили натиснете Alt и после натискате където искате в снимката за да я намалите. Предимно можете да увеличавате и намаляте размера на видимост с копчета от клавиатурата (Ctrl плюс + и Ctrl плюс -). След като увеличите снимката можете да я разглеждате на всякъде като използвате инструмента Hand (ръчичка)



или можете да натиснете Space от вашата клавиатура (едно голямо дълго копче) и чрез влачене на мишката да разглеждате увеличеното изображение.

След като сте активирали Blur Tool можете да замъглите лицето на момчето, но не повтаряйте много пъти защото замъгляването ще стане много силно. След като обработите лицето обработете и ръцете, косата (много леко) и блузата. Когато сте готови можете да запишете изображението за всеки случай, ако нещо стане с компютъра ви. Това става от File > Save или Save as (ако искате да запишете изображението в различна папка или с ново име).

Другата стъпка е да заострите някои части от снимката. За тази цел активирайте инструмента Sharpen Tool. Той е под Blur Tool и активирането му става като натиснете с дясното копче на мишката върху активирания в момента инструмент (в случая Blur Tool). Ще ви се отвори по меню и от там изберете инструмента. Ако искате пак увеличете снимката с инструмента Zoom за да ви е по - лесно да работите. Сега задайте силата, с която ще изостряте. Това става пак от менюто с опции за инструмента. Въведете 50 % сила и започнете обработката. Рисувайте по стената с мазилката за да я изострите, а да стои така замазана и замъглена. След като получите ефект приемлив за вас рисувайте и по стената на гаража и по оградата. Накрая съвсем леко можете да минете и листата на дървото.

Това беше цялата работа, която трябваше да свършите в този урок.

Ако сте стигнали до тук благодаря за търпението, което сте имали :).

Автор: Росен Иванов

Photoshop - Записване на изображения за Web

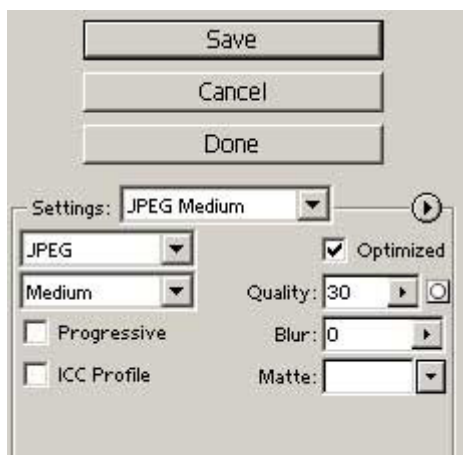
(Adobe **Photoshop 7**)

В този урок вие ще се научите как да записвате файлове, за web, как да оптимизирате различните формати, за да бъдат с по - малки размери и кои формати са за предпочитане. Нека да започваме.

За този урок можете да ползвате файла от предния урок (обработка на изображение). Отворете вашият файл start.psd или final.psd и изберете от менюто File > Save for Web. Главната задача тук ще бъде да разберете за какво служат всички бутони и менюта, които се намират в save for web.



Това представлява менюто на опцията Save for Web. Предимно вие можете да записвате дадено изображението под няколко формата : JPEG, GIF, PNG-8, PNG-24, WBMP. Можете да избирате качеството на записване : Low, Medium, High (ниско, средно, високо). Можете да избирате и ръчно качеството на записване като коефициент от 0 до 100. Можете да използвате Blur (замазване) за да намалите големината на файла след записа.



Ето, на снимката е показано менюто за записване. Показан е активен формат - JPEG, качеството Medium, Progressive и ICC Profile предимно трябва да за изключени. При Progressive се заличават цветовете, които не са видим за човешкото око, с цел намаляване на размера на файла. При ICC Profile обаче размера на изображението драстично се увеличава, защото при тази опция се премахват изострените контури и изображението изглежда по -

добре независимо дали е с ниско качество или високо.

Има няколко нива за преглеждане на изображението при web записване ето ги и тях:



Original - Показва оригиналното изображение, без никакви оптимизации.

Optimized - Показва оптимизираното изображение и възможност за ръчна настройка за оптимизиране.

2-Up - Показва два варианта на оптимизация. Предимно единия вариант е силно оптимизиран, а другия по - слабо.

4-Up - Показва четири варианта на оптимизация и вие можете да си изберете един от тях.

JPEG
18,58K
8 sec @ 28.8Kbps

При оптимизирането вие можете да гледате колко голям е файла и за колко време ще се зареди за дадена скорост в интернет. Ето пример на принт скрийна по - горе.

Предимно за web записване се използва формата jpeg, пред GIF, защото е много по- малък като размери и е с по - добро качество. Ако обаче сте направили някаква анимация на Image Ready няма да можете да я запишете под формат jpeg а само под GIF.

Вие научихте една важна стъпка в Photoshop. Сега вече можете спокойно да си оптимизирате файловете, които ще сложите във вашата страница.

Автор: Росен Иванов

Photoshop - Златен текст

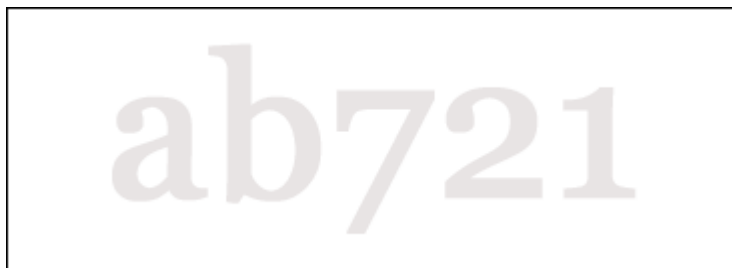
(Adobe **Photoshop 7**)

Тук ще разберете как лесно и бързо да направите "златен" текст в Photoshop.

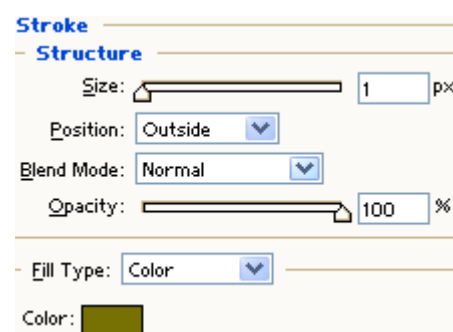
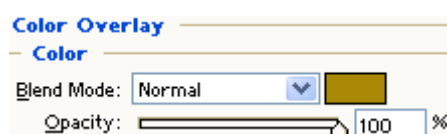
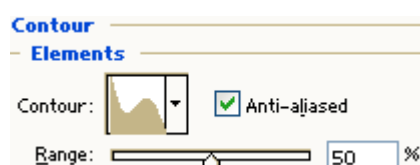
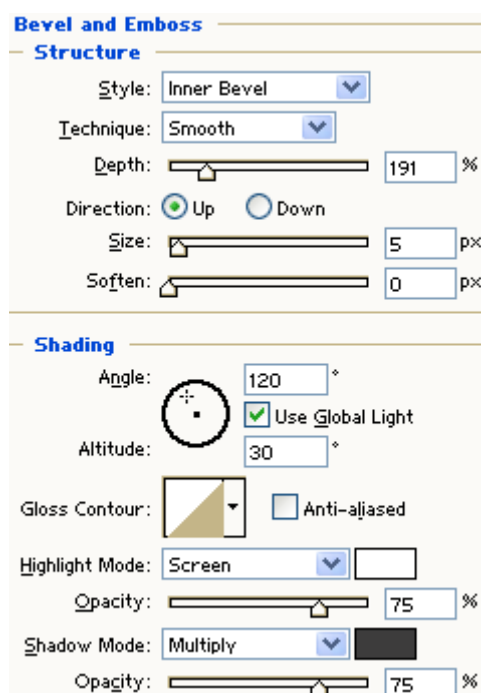
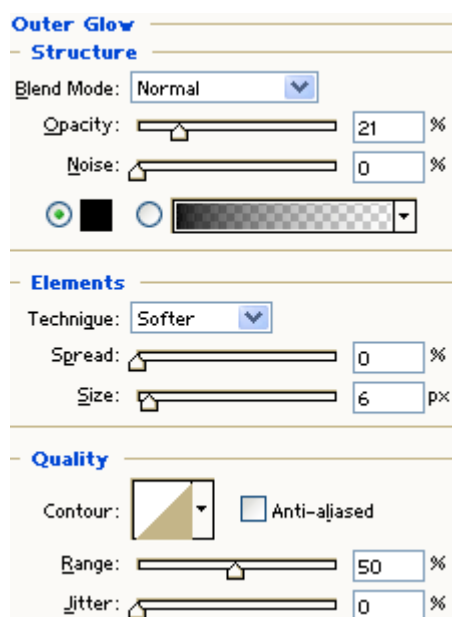
Изисквания: Photoshop 7 или по-добър.

1. Отворете нов файл с размери 360 на 130 пиксела и с бял фон. Нека резолюцията да бъде 72.

2. Направете нов layer и напишете нещо с "type text tool", независимо с какъв цвят. Ето какво получих:



3. Задайте следните опции:



Трябва да се получи нещо такова:



4. Направете нов layer, сложете го под layer-а на текста и ги слейте чрез Ctrl+E

5. Изберете Image > Adjustments > Hue and Saturation

Изберете Colorize

Hue: +1

Saturation: +31

Lightness: 0

Натиснете ОК

Ето какво получих аз:



Превод: Георги Гърдев (GeorgeSG) Източник: www.tutorialfx.com

Photoshop - Кървав текст

(Adobe Photoshop 6)

1. Отворете нов прозорец с бял фон. Напишете някакъв текст, направете го в **bold**
Отидете на Image > Mode > Greyscale
2. Select > Load Selection
При Channel изберете Transparency
3. Layer > Flatten Image
4. Select > Inverse
5. Filter > Pixelate > Crystallize - size 4

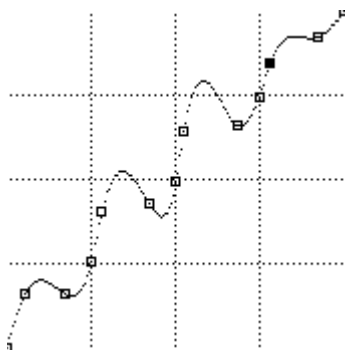
6. Select > Inverse
7. Noise > Add Noise

Настройте:
Guassian
Monochromatic
Amount: 70

8. Blur > Guassian blur (Radius 2.0)

9. Image > Adjust > Curve

Опитайте се да направите нещо такова:

**10.** Натиснете Ctrl + D, след това Image > Adjust > Invert**11.** Image > Rotate Canvas > 90 CW**12.** Filter > Stylize > Wind - настройте - method: wind, from the right. Натиснете OK. Натиснете ctrl+f**13.** Image > Rotate Canvas > 90 CCW, Image > Mode > RGB Color**14.** Image > Adjust > Hue Saturation, натиснете colorize

Hue:0
Saturation:100
Lightness:-5

**15.** Ако искате добавете Filter > Blur > Blur

Превод: Георги Гърдев (GeorgeSG) **Източник:** www.insidegraphics.com

Flash MX - Flash на цял екран

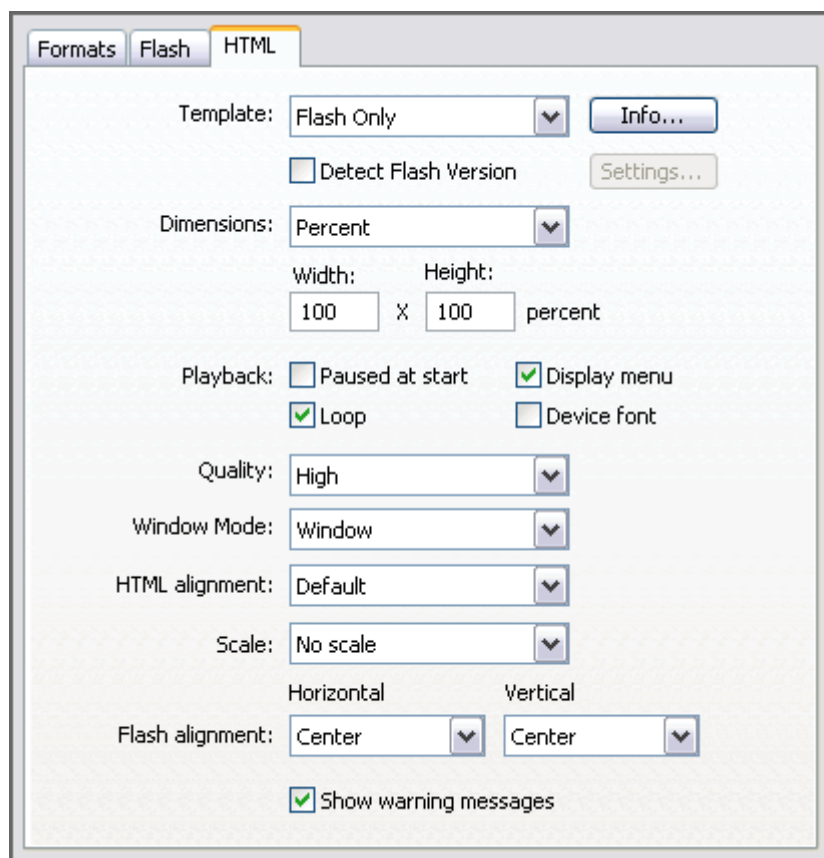
(Macromedia **Flash MX 2004**)

Елементарна статия, която ще ви научи да създавате Flash на пълен екран (full screen) в HTML. Може да ви се стори малко различно от обичайните full screen ефекти, които сте виждали. Тук не се копира височината и широчината на Flash файла, просто се разширява Flash страницата без да се скалират елементите отвътре. По този начин ако сте използвали на пиксел шрифтовете за текст, всичко ще бъде свежо и чисто.

В тази статия е използван Flash MX 2004 както и малко от CSS за да се отървем от стандартните граници.

Нека да започнем:

1. Отворете някакъв файл във Flash.
2. Отидете на File -> Publish Settings. Изберете HTML панела.
3. Щракнете падащо меню "Dimensions". Изберете "Percent" и стойност 100 за дължина и широчина.
4. Щракнете падащо меню "Scale" и изберете "No scale".
5. Щракнете "Publish" за да публикувате файловете.



(Снимка на "Publish" настройките във Flash MX 2004.)

6. Сега трябва да имате два нови файла: SWF и HTML файл.

Кода за HTML файла, трябва да изглежда подобно на това :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/
xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>myfile</title>
</head>
<body bgcolor="#ffffff">
<!--url's used in the movie-->
<!--text used in the movie-->
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000" codebase="http://
fpdownload.macromedia.com/pub/
shockwave/cabs/flash/swflash.cab#version=7,0,0,0" width="100%" height="100%" id="myfile"
align="middle">
<param name="allowScriptAccess" value="sameDomain" />
<param name="movie" value="myfile.swf" />
<param name="quality" value="high" />
<param name="scale" value="noscale" />
<param name="bgcolor" value="#ffffff" />
<embed src="myfile.swf" quality="high" scale="noscale" bgcolor="#ffffff" width="100%"
height="100%" name="myfile" align="middle" allowScriptAccess="sameDomain"
type="application/x-shockwave-flash" pluginspage="http://www.macromedia.com/go
/getflashplayer" />
</object>
</body>
</html>
```

(код на HTML файл)

7. Тази част не е задължителна. Повечето браузери поставят стойност по подразбиране за границите. Често тази стойност не е нулева, затова ние ще използваме CSS за да достигнем до нула.

Кодът е доста красноречив. Има се в предвид височината и широчината да идат на 100%, а стойността на границата до 0. Прибавете този CSS код в HTML файла (в head тага).

```
<style type="text/css">
<!--
body {
height: 100%;
width: 100%;
margin: 0;
}
-->
</style>
```

(въведете кода между head таговете)

Забележка:

Забележете, че CSS не може да работи в някои от по-старите браузери, които не го поддържат напълно.

Това е! Надявам се, че ви хареса статията.

Превод: The ONE **Източник:** www.kirupa.com

Fireworks MX - Шоколадов текст

(Macromedia **Fireworks MX 2004**)

За тези от вас, които искат шоколадов текст! Ефекта е доста елементарен. Може да се направи с Fireworks MX 2004, като се приложат малко вариации в настройките.

Стъпка 1

Напишете някакъв текст. Уверете се, че използвате шрифт, който е дебел и размера не е прекалено малък, тъй като ефекта трябва да е ясно видим. Шрифта използван в тази статия е **Forte** с размер - 100 и цвят - #620000.

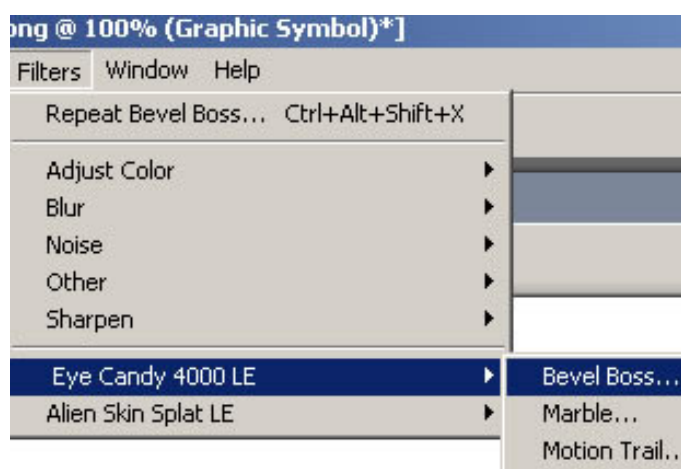
Стъпка 2

Текста трябва да бъде конвертиран в битмап. За да направите това, изберете текста и щракнете на: Modify -> Convert to Bitmap.

Забележка: Дори и да не намерите опцията, преминете към следващата стъпка. Там ще бъдете попитан за конвертиране.

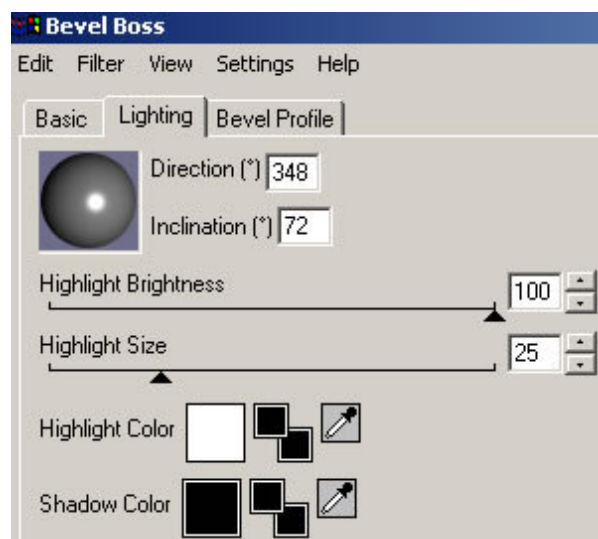
Стъпка 3

Селектирайте текста и щракнете на: Filters -> Eye Candy 4000 LE -> Bevel Boss...



Стъпка 4

След като се покаже прозорец **Bevel Boss**, щракнете на панел **Lighting** и изберете по-долните настройки.



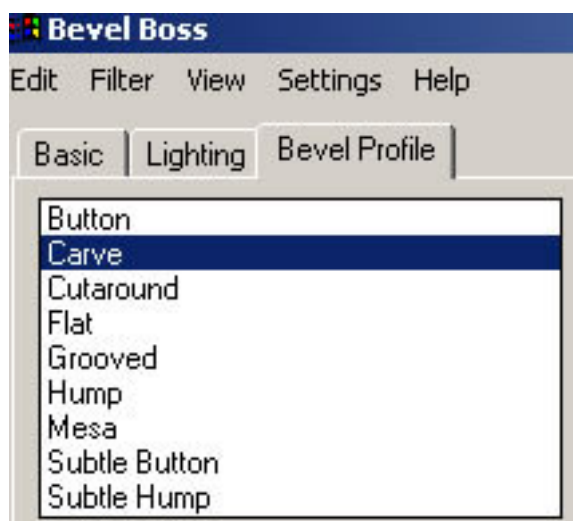
Променете стойностите на **Direction** и **Inclination**. Също друг начин за това е да премествате бялото петно на кръга. За стойностите на **Highlight Brightness** и **Highlight Size**, използвайте 100 и 25.

Стъпка 5

Highlight и **Shadow** цветовете, зависят от цвета на текста или както на вас ви харесва. Аз избрах настройките по подразбиране (черно и бяло) както видяхте по-горе. Когато променят нещо, текста веднага ще се актуализира веднага.

Стъпка 6

Щракнете върху панел **Bevel Profile** и изберете настройка **Carve**, после дайте **Settings** от менюто и изберете **Ridged Carve**.



Стъпка 7

Това е финалното изображение!

chocolate

Огладняхте ли.....?

Превод: The ONE Източник: www.entheosweb.com

Fireworks MX - Рисувана рамка

(Macromedia **Fireworks MX 2004**)

Fireworks MX има доста различни рамки. Може да променят атрибутите на писалка, молив и четка така, както рисувате и с вектор. Аз използвах някои от категориите рамки, за да създам пъстри граници и да ги запамеля като стил. Можете да прилагате стилове само към вектори. Също може да вмъкнете изображение във вектор стила, като рамка подобна на тази вдясно.

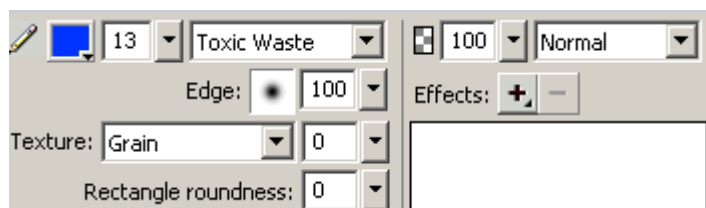


Как да създадем рамка и да я запазим като стил?

Следвайте тези стъпки, за да научите как да създавате ваши собствени рамки и как да ги запамятвате като стилове. Също ще разберете и как се експортират и импортират тези стилове.

Стъпка 1

Щракнете на форма правоъгълник (намира се в менюто вляво, раздел vector). Отидете при настройките (намират се долу - properties) и изберете категория **stroke** (в примера на снимката се намира вдясно от 13, там при Toxic Waste), вижте разнообразните атрибути, като запълване, текстура и т.н. и изберете каквото ви харесва. Аз направих рамка в розови оттенъци.



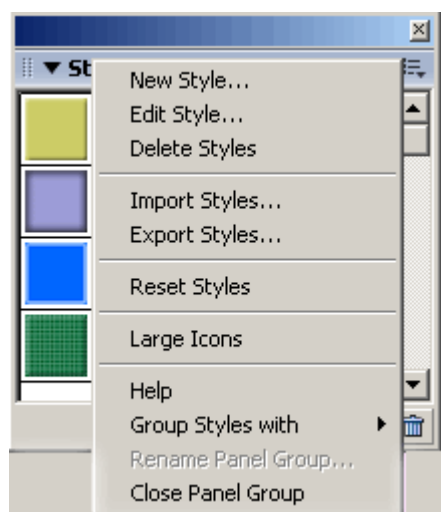
Стъпка 2

След като избрахте настройките, нарисувайте правоъгълника. Свойствата на вектора, ще се покажат. Може да приложите някои ефекти като сянка, топлина и т.н. (Намират се пак там долу на properties, търсете бутон Add Effects или бутон с плюс).

Стъпка 3

Сега нека да запишем стила, мащабирайте вектора (формата) до 100 px на 100 px. (при избран вектор отидете на: Modify -> Transform -> Scale). Отидете на **Styles Panel** (ако не се вижда щракнете: Window -> Styles). Щракнете падащото меню и изберете **New Style**. Въведете име за стила, проверете атрибутите, които искате да запаметите и натиснете **OK**.

Стъпка 4



Също може да експортирате стила и да го запишете в .stil файл. За да направите това, изберете стила, който сте създали (задръжте Shift за избор на много стилове) и щракнете на **Export Styles**, въведете име и местоположение. Щракнете **Save**.

Стъпка 5

За да импортирате стилове в **styles panel** изберете **Import Styles**.

PHP - Запознаване с променливите

Променливите са контейнери, които съдържат различни стойности във времето. Всеки език за програмиране следва определени правила при декларирането на променливи. Тези правила включват максимална дължина на името на променливите, дали могат да се състоят от цифри или от букви и цифри, дали името на променливата може да включва специален символ, както и дали може да започва с цифра. Ще разгледаме правилата, които трябва да следвате при задаване на име на променливата в PHP.

Правила за задаване име на променливата

В PHP за разлика от голяма част от останалите езици за програмиране, няма ограничение за дължината на името на променлива. Освен това можете да използвате букви и цифри, но трябва да се съобразявате със следните правила:

- Имената на променливите трябва да започват със символ за долар (\$).
- Имената на променливите могат да започват със символа '_'.
- Имената на променливите не могат да започват с цифра.

Имената на променливите трябва да са уместни и обясняващи значението на променливата. Имената като \$a или \$temporary може да са разбираеми и да се запомнят по време на писането на конкретната част от кода. Ще бъде трудно обаче да се разбере за какво са използвани, като друг програмист или вие разгледате кода след време. Ето и някои примери за правилно и неправилно зададени имена на променливи:

Правилни:

- \$prod_desc
- \$Intvar
- \$_Salesamt

Неправилни:

- \$9OctSales
- \$sales123
- \$asgs

След като се запознахте с правилата за задаване на имена на променливи, нека да видим как можете да декларирате променливи в PHP.

Деклариране и инициализиране на променливи

Декларирането на променливи представлява специфицирането на всички променливи, които възнамерявате да използвате в програмата си. В езиците за програмиране като C и C++ тези променливи трябва да се декларират изрично заедно с типа на данните си, например:

```
int Number=1010;
```

Обаче, в PHP не е необходимо да се декларира типа на данните за променливата. Можете да декларирате променливата при първото ѝ използване. Типът на данните на променливите се променя на базата на тяхното съдържание.

Инициализацията представлява присвояване на стойности на променливата. Тези стойности

може да са от различни типове. Типовете данни в PHP могат да се разделят на две: *скаларни* и *съставни*. Скаларните типове данни са простите типове, които имат повечето езици за програмиране. Тези типове формират градивните блокове на програма, написана на произволен език за програмиране. Скаларните типове включват:

- Низ
- Цяло число
- Булева стойност
- Реално число

От друга страна, комбинирането в едно на множество прости типове данни и съхраняването им под дадени име създава сложен тип данни. Тези типове данни са познати под името *съставни типове данни* и се използват за съхраняването на данни от различни типове и за обработката им. Съставните типове включват:

- Масив
- Обект

Нека да разгледаме поотделно всеки тип.

- **Цяло число.** Този тип представлява целите числа. Числата могат да са както положителни, така и отрицателни. Използва се символът '-' преди числото за представяне на отрицателни стойности.

- **Низ.** Низът представлява последователност от символи, като символите са с размер 1 байт. Можете да имате 256 различни символа за един низ. В PHP обаче не се налага ограничаване на низа.

- **Булева стойност.** Този тип се ограничава само до 2 стойности: True и False. Ако ви се наложи да разгледате тази стойност като числа, True=1, а False=0.

- **Реално число.** Този тип също се използва за съхраняване на числови стойности, но те могат да съдържат и десетични стойности.

- **Масив.** Масивът се използва за съхраняване на множество стойности от един и същи тип. Можете да зададете дължина на масива при декларирането му или по време на изпълнението на програмата. Всяка стойност, съхранена в масив, се нарича *елемент*. Имате достъп до съдържанието на елемент от масив чрез задаването на позицията (индекса) му в масива.

- **Обект.** Обектите са важна част от всеки език за програмиране. Те съдържат не само данни, но и функции за извършване на операция върху данните.

Ето няколко примера за присвояване на стойности на променливи:

```
$Mycomputer = "Това е пример за низ";  
$Myintval = 4346; // Цяло число  
$Myboolval = "True"; // Булева стойност  
$Myfloatval = 354.4654; // Реално число  
$Myarrval [0] = "My"; // Масив от три елемента  
$Myarrval [1] = "First"; // Тези три елемента са достъпни  
$Myarrval [2] = "Array"; // с едно име на променливата
```

В горните примери на променливите директно се присвояват стойности. Можете да

присвоявате и стойности на променливи чрез псевдоним.

Присвояване на стойности чрез псевдоним

По подразбиране всяка променлива има своя уникална стойност. Това означава, че всяка променлива има отделена памет, която е асоциирана в нея и съхранява текущата и стойност. Нека да разгледаме един пример: `$empname = Shyne`. Тук системата заделя част от паметта за променливата `$empname`, в която променливата съхранява стойност `Shyne`. Ако друга променлива трябва да използва тази стойност, трябва да присвои стойността. Така системата ще поддържа множество копия на едни и същи стойности, което води до излишък от данни. Можете да избегнете тази ситуация, като присвоите стойност чрез псевдоним. В този случай системата поддържа само едно копие на данните, към което се обръщат всички променливи, съхраняващи тази стойност. Така всички тези променливи отговарят на една и съща стойност в паметта. Можете да присвоите стойност чрез псевдоним, като поставите символа амперсанд (&) пред името на променливата. По този начин на променливата се присвоява *адресът на данните* в паметта, а не се създава ново копие на данните.

След като се запознахте с правилата на задаване на имена на променливите, ще разгледаме областта на действие на променливите. *Областта на действие (видимост)* на променлива в програмата е областта, в която можете да се обръщате директно към променливата.

Област на действие на променливите

Областта на действие на променлива в програма определя дали други програми могат да осъществят достъп до променлива извън родителската програма или дали части от същата програма мога да осъществят достъп до променливата. родителската програма е програмата, в която променливата е дефинирана и инициализирана. Областта на действие на променливата също така определя кога системата ще създаде променливата в паметта и кога ще я унищожи.

Ако декларирате променлива във функция или като аргумент на функция, областта на действие на променливата ще е *локална*. Това означава, че може да се използва само във функцията, във която е декларирана. Обаче ако декларирате променлива извън функцията, тогава областта на действие ще е *глобална*. Това означава, че можете да я използвате във всяка функция.

БЕЛЕЖКА Функциите се използват за съхраняване на част от кода под зададено име. Това позволява лесно използване на кода, тъй като трябва само да зададете името на функцията, където искате да използвате този код. Използването на функции също така улеснява процеса трасиране (debugging), тъй като трябва да направите промяна само на едно място. Всяка промяна, направена във кода на функцията, се отразява на всички места, където тя се извиква. Това прави кода по лесен за проследяване и разбиране.

В PHP можете да имате локални или глобални променливи. Областта на действие на всяка променлива, декларирана във функция, е самата функция. В PHP една променлива може да е глобална по декларация, но все пак ще бъде локална, ако не и присвоите стойност извън функция. В PHP трябва да декларирате променливите като глобални и да ги инициализирате във функциите, в които ще използвате. Една функция може да съдържа неопределен брой глобални променливи.

Променливи от обкръжението

Може би вече знаете, че повечето скриптове и някои програми се изпълняват във брауъра на потребителя. Определени скриптове обаче се изпълняват на WEB сърварите. Тези скриптове са познати под името сърварски скриптове. Тези сърварски скриптове използват *променливи от обкръжението*, за да предават информацията съхранена на WEB сървара към

външни програми изискващи информацията. Тези променливи съхраняват информацията, отнасяща се за сървара, като например информация за достъпа и идентификацията. Въпреки, че се наричат променливи от обкръжението, те не са свързани с променливите, поддържани от операционната система. Тези променливи се използват от операционната система единствено за предаване на информацията към програмата, която я е изисквала. Можете да задавате стойности на променливите от обкръжението. Имената и предназначението на някои от често използваните в PHP променливи от обкръжението са показани в таблицата:

\$argv	Съдържа аргументите, предавани на скрипта.
\$argc	Съдържа броя на аргументите, предавани на скрипта.
\$PHP_SELF	Съдържа името на текущия скрипт. Тази стойност е недостъпна при изпълнение от командния ред.
\$HTTP_GET_VARS	Съдържа масив от променливи, получени чрез метода HTTP GET и се съхранява в текущия скрипт.
\$HTTP_POST_VARS	Съдържа масив от променливи, получени чрез метода HTTP POST и се съхранява в текущия скрипт.
\$HTTP_COOKIE_VARS	Съдържа масив от променливи, получени чрез cookies и се съхранява в текущия скрипт.
\$HTTP_ENV_VARS	Съдържа масив от променливи, съхранени в текущия скрипт, като се използва родителското обкръжение.
\$HTTP_POST_FILES	Съдържа масив от стойности, който съхраняват информация за качените файлове. Чрез използването на метода HTTP POST можете да качите тези файлове.
\$HTTP_SERVER_VARS	Съдържа масив от променливи, предадени на текущия скрипт от HTTP сървара.

След като се запознахме с променливите, можете да ги използвате, за да манипулирате стойностите им. Манипулацията на стойностите включва прилагането на математически оператори, сравнява, слепвания и присвоявания. За да можете да манипулирате променливите, трябва да използвате **оператори**. PHP предоставя различни типове оператори за извършването на различните задачи.

Автор: Любомир Стефанов (Shyne)

PHP - Оператори 1 част

Оператора е част от кода, който използва една или повече променливи или елементи, за да осъществите изчисление или за да сравните изрази. Можете да използвате оператори за извършване на аритметични и логически операции, както и за сравнение. В PHP са налични следните оператори:

Аритметични оператори: Използват се за математически изчисления

Оператори за присвояване: Използват се за присвояване на стойности

Оператор за сравняване: Използва се за сравнения

Оператор за изпълнение: Използва се за сравнение на код, като команди на шела; резултатът от изпълнението се присвоява в променлива

Инкрементиращи/декрементиращи оператори: Използват се за Инкрементиращи (увеличаване) и декрементиращи (намаляване) стойности

Низови оператори: Използват се за конкатенация на 2 низа

Конкатениращи оператори: Използват се за разбора на низове

Сега ще се запознаем с тези оператори в подробности.

Аритметични оператори

Както може би се досещате от името, можете да използвате тези оператори за извършване на математически действия. Ето и операторите:

+ (Събиране) Изчислява сумата на 2 числа или конкатенира низове.

- (Изваждане) Изчислява разликата между две числа.

* (Умножение) Изчислява произведението на 2 числа.

/ (Делене) Изчислява частното на две числа

% (Остатък) Изчислява остатъка при делението на две числа

Ето ви и няколко примера за използване на аритметични оператори:

```
$Num1 = 9;  
$Num2 = 3;  
$Result = $Num1 + $Num2; //$Result = = 12  
$Result = $Num1 - $Num2; //$Result = = 6  
$Result = $Num1 * $Num2; //$Result = = 27  
$Result = $Num1 / $Num2; //$Result = = 3  
$Result = $Num1 % $Num2; //$result = = 0
```

В горния пример на променливата \$Num1 и \$Num2 са присвоени стойности и са използвани аритметични оператори.

Оператор за умножение

Оператора за умножение се използва за умножение на две числа. Синтаксисът на оператора за умножение е следния:

```
$Result = $Num1*$Num2;
```

В горния синтаксис PHP разглежда \$Num1 и \$Num2 като числа.

В следващите два примера, оператора за умножение се използва за намиране на произведението на две числа:

```
$Result = 3*5; // Връща стойност 15  
$Result = 556.29*262.34 // Връща стойност 145929.2484
```

Оператор за деление

Можете да използвате оператора за делене за намиране на частното на две числа. Оператора връща стойност реално число. Синтаксисът е следния:

```
$Result = $Num1 / $Num2
```

В горния синтаксис \$Num1 и \$Num2 са числа. Оператора връща резултата от делението на \$Num1 и \$Num2 като реално число.

Ето и няколко примера за използване на оператора за делене.

```
$Result = 19 / 3; // Връща стойност 6.3
```



```
$Result = 26 / 3; // Връща стойност 8.66667
```

В горните две примера първото цяло число се дели на второто и резултатът се съхранява в променливата \$Result

Оператор за деление на модул

Оператора за деление на модул се използва за намиране на остатъка при деление на две числа. Синтаксисът на оператора е следният:

```
sult = $Num1 % $Num2;
```

Където, \$Num1 и \$Num2 са числа. Оператора връща остатъка при деление на \$Num1 с \$Num2.

Ето няколко примера за използване на оператора за деление на модул:

```
$Result = 24 % 8; // Връща 0  
$Result = 18 % 8; // Връща 2  
$Result = 12 % 4.3; // Връща 3.4  
$Result = 47.9 % 9.35; // Връща 1.15
```

Ако използвате променливи от тип *float* като операнди за оператора делене по модул, тогава остатъка от делението е също реален тип.

Оператор за събиране (+)

Оператора за събиране се използва за намиране на сумата на две числа. Синтаксисът на оператора е следният:

```
$Result = $Variable1 + $Variable2;
```

Променливите в израза могат да са или числа или низове. Ако използвате числа, оператора връща сумата на \$Variable1 и \$Variable1. Ако променливите са низове, оператора връща конкатенацията им.

Когато използвате оператора за събиране, трябва да сте сигурни, че и двата операнда са от един и същи тип.

Разгледайте следващите примери:

```
$IntValue = 8 + 6; // Връща 14  
$IntValue = 348.45 + 5468; // Връща 5816.45  
$StrValue = "Hello" + "World"; // Връща Hello World
```

Ако операндите са цели числа или реални числа, както е в случая с горните два примера, Операторите сумират числата и връщат съответно цяла или реална стойност. В случаите, в които операндите са низове, оператора конкатенира двата низа.

Оператори за изваждане (-)

Операторът за изваждане се използва за намиране на разликата на две числа. Синтаксиса на оператора е следния:

```
$Result = $Num1 - $Num2;
```

Където \$Num1 и \$Num2 са числа. Разгледайте следните примери:

```
$IntResult = 15 - 4; // Връща 11  
$IntResult = 756.45 - 325.26; // Връща 431.19
```

Оператор за присвояване

В момента когато видите оператора **=**, вероятно го асоциирате с математическото равенство. Въпреки, че това може да е вярно за някои езици за програмиране, в PHP това е оператор за *присвояване* и се използва за задаване на стойност на променлива.

Например:

```
$Myvar = 12;  
// $Myvar става променлива от целочислен тип и има стойност 12.
```

```
$Myvar = World;  
// $Myvar става променлива от низов тип и има стойност 'World'.
```

PHP ви позволява да използвате повече от един оператор за присвояване в един израз. Например:

```
$Result = ($Num1 = 3) + ($Num2 = 1);  
// $Result, $Num1 и $Num2 са целочислени променливи и имат за стойност съответно 4, 3 и 1
```

В горния пример първо се присвояват стойности на \$Num1 и \$Num2, след което се изчислява тяхната сума и се присвоява на \$Result.

PHP също така поддържа и комбинирания оператори **+=** и **.=**. Тези оператори не само задават стойност на израза, но и съхраняват резултата от израза. Оператора **+=** се използва за числа, а оператора **.=** за низове.

Например:

```
$Num = 3;  
$Num+=1; // Увеличава стойността на $Num с 1  
echo $Num;
```

В горния код променливата \$Num се инициализира и на следващия ред текущата стойност на \$Num се събира с единица и резултатът се присвоява отново на променливата.

```
$Myvar = "Add";  
$Myvar .= " Something"; // $Myvar приема стойност "Add Something"
```

Горния израз може да се запише и като \$Myvar = "Add" . "Something" Операторът **.=** добавя нова стойност към текущата на \$Myvar.

Оператор за сравнение

Операторът за сравнение се използва за сравняване на изрази. Можете да проверявате дали стойността на един израз е по-малка или равна на стойността на друг.

Теоретичния оператор **?** е условен оператор в PHP. Може би вече сте го използвали за да сравните два изрази и да покажете трети израз, ако двата изрази имат равни стойности в някой друг програмен език. Синтаксисът е следния:

```
(expr1) ? (expr2) : (expr3);
```

където ако израза `expr1` има стойност `true`, то като резултат се връща стойността на променливата `expr2`, в противен случай стойността на `expr3`.

Оператори за сравняване

Равенство `==` Връща `true`, ако стойността на първата променлива е равна на втората

Идентичност `===` Връща `true`, ако стойността на първата променлива е равна на втората и типовете им са еднакви

Неравенство `!=` Връща `true`, ако стойността на първата променлива не е равна на втората

По-малко `<` Връща `true`, ако стойността на първата променлива е по-малка от втората

По-голямо `>` Връща `true`, ако стойността на първата променлива е по-голяма от втората

По-малко или равно `<=` Връща `true`, ако стойността на първата променлива е по-малка или равна на втората

По-голямо или равно `>=` Връща `true`, ако стойността на първата променлива е по-голяма или равна на втората

Да разгледаме един пример за използване на оператори за сравнение:

```
<?
$Num1 = 12.0;
$Num2 = 12;
echo "Num1, " == ", $Num2, " == ", $Num1 == $Num2;
// Връща False
```

```
$Num1 = 13;
$Num2 = 12;
echo "Num1, " == ", $Num2, " == ", $Num1 == $Num2;
// Връща True, защото операторът == не сравнява типовете им, а само стойностите на променливите
```

```
$Num1 = 12;
$Num2 = 12;
echo "Num1, " == ", $Num2, " == ", $Num1 == $Num2;
// Връща True
?>
```

Автор: Любомир Стефанов (Shyne)

RНР - Динамична промяна на типове 1 част

В RНР не е необходимо да декларируете променливите, като им зададете тип; техният тип се променя в зависимост от съдържанието им. Например, променлива с име `$test_item` може да се инициализира с низ, след което може да съдържа цяло число. Поради тази причина в RНР променливите могат да съхраняват произволен тип данни, без да се декларира изрично типът им. Можете също така да съхранявате число като низ, ако го поставите в кавички.

Ето и няколко примера:

```

$Myval = " 1500 ";      //Съхранява стойността като низ
$Myval = 1500;          //Съхранява стойността като цяло число
$Myval = " 145.215 ";   //Съхранява стойността като низ
$Myval = " 145.215 ";   //Съхранява стойността като десетично число

```

В горния пример типът на променливата `$Myval` се променя на базата на зададена ѝ стойност.

Освен промяната на типа на базата на съдържанието на променливите, можете също така да зададете изрично типове на променливите.

Явно задаване (промяна) на типовете

Езикът PHP предоставя функционалност за промяна на типовете, подобна на тази в C. При явната промяна на типовете (type casting) вие променяте явно типа на променливата. В PHP можете да извършвате следните промени на типовете:

Видове промяна на типовете

Предназначение	Преобразуване
За конвертиране на булева стойност	(bool), (boolean)
За конвертиране до целочислена стойност	(int), (integer)
За конвертиране до стойност десетична дроб	(float), (double), (real)
За конвертиране до масив	(array)
За конвертиране до обект	(object)

Можете също така да конвертирате стойност до низ, като я поставите в кавички. Когато преобразувате масив до низ, резултатът е низът "Array". Обаче ако конвертирате обект до низ, резултатът е: "Object". От друга страна, ако низ се конвертира до масив, низът, става първи елемент на масива, а когато конвертирате низ до обект, резултатът е атрибут с име 'scalar' и стойност конвертирания низ.

Например:

```

$Val = ' Hello ';
$Greet = (array) $Val;
echo $Greet [0] ;      // Резултатът е 'Hello'

```

```

$Val = 'Hello';
$obj = (object) $Val;
echo $obj -> scalar;    // Резултатът ' Hello'

```

В горните примери променливата `$Val` има стойност Hello. В първият пример променливата `$Val` се конвертира до масив с име `$Greet`. Тази променлива съдържа символите на низа като отделни елементи на масива. Във втория пример същата променлива `$Val` се преобразува до обект с име `$obj`, който може да се използва за визуализиране на стойността.

Можете да конвертирате променлива от произволен тип до булев, целочислен тип или низ. Трябва обаче да спазвате някои правила при конвертирането на от един тип в друг. Следващите секции разглеждат конвертирането до булев, целочислен и низов тип.

Конвертиране до булева стойност

PHP 4 предоставя възможност за конвертирането на стойностите до булеви. Булевата стойност се определя в зависимост от конвертираната стойност и може да е или True, или False.

Можете да използвате изразите (bool) или (boolean), за да конвертирате променлива до булева стойност. Но в PHP по-честият случай е на *неявно конвертиране* до булева стойност, ако извиканата функция го изисква.

По време на конвертирането до булева стойност, при следните условия получената стойност е False:

- Празен низ или низ съдържащ 0
- Целочислена стойност 0
- Променлива имаща стойност NULL
- Масив с 0 елемента
- Обект с 0 елемента

Във всички други случаи конвертираната стойност е True.

Конвертиране до целочислена стойност

Можете да използвате изразите (int) и (integer), за да конвертирате стойности до целочислени. Ако се опитвате да конвертирате реално число до цяло число, се взема само цялата част. Ако конвертирате булева стойност до цяло число, се взема само цялата част. Ако конвертирате булева стойност до цяло число, стойността True се конвертира до 1, а стойността False до 0. Както ще видите в случаите на конвертиране до низ, при низовете се взема началото, за да се определи каква ще бъде стойността при конвертирането.

Конвертиране на низ

Низ съдържащ цяло число може лесно да се конвертира до целочислен тип. Но ако низът съдържа типовете '.', 'e' или 'E'. той се конвертира до десетична дроб. При определяне на типа на конвертирания низ, се взема предвид началото на низа. Например, ако низът започва с число, то конвертираната стойност е число, ако низът започва с букви, то получената стойност е нула. Изключение от това правило се прави, когато низът освен букви съдържа и цифри, но разделени с интервали. В този случай резултата е само числовата част:

Например:

```
$item = " 15 boxes grain" + 2 ; // Резултатът е 17
$item = "15units" + 2;         // Резултатът е 15
$item = "12.4" + 2;            // Резултатът е 14.4
$item = "-12.4kg" + 2;         // Резултатът е -14.4
```

В горните примери, операторът за събиране се използва за събиране на низ с число. Както може би сте забелязали, ако низът съдържа цифра, цифрите се извличат и се събират с число.

Променящи се променливи

Променящи се променливи представляват променлива, съхраняваща име на променлива. Въпреки, че това може да звучи малко объркващо, това означава, че текущата стойност,

съхранена в \$greet, се използва за създаване на друга променлива. Това се обяснява от следващия пример.

```
$greet = "greeting";  
$$newgreet = "hello!";  
// Това е еднакво с реда $greeting = "hello";
```

В горния пример променливата \$greet съдържа стойността greeting. Вместо да се създаде нова променлива с име \$greeting и да и се присвои стойност, можете да използвате \$\$newgreet. По този начин се създава нова променлива на базата на текущата стойност на \$greet и ѝ се присвоява стойността hello!. По този начин можете да създадете динамични променливи и да им присвоите стойност.

Също така произволни данни съхранени в променлива, могат да станат име на функция.

Както споменахме по-рано, променливите в PHP се декларират при тяхното използване и не е необходимо да се декларира типът на данните им. В някой случай обаче може да ви се наложи да създадете тип на променлива. Това обикновено става в случаите, когато искате променливата да съхранява данни само от определен тип. Тогава трябва изрично да зададете тип на данните, които ще се съхраняват в променливата. За тази цел PHP предоставя 2 функции settype() и gettype(). Функцията gettype() извлича типа на променливата, а функцията settype() задава типа на дадена променлива. В следващата част ще разгледаме тези две функции по-подробно.

Автор: Любомир Стефанов (Shyne)

CSS - Заоблени ъгли

Въведение

Един от най-добрите уеб сайтове в интернет като функционалност е, и винаги е бил, Amazon (www.amazon.com). Но що се отнася до достъпност, той не е толкова добър.

Проблемът

Табовете в менюто, например, на Amazon, изглеждат наистина добре, но са **напълно недостъпни**. Първо, липсва им "ALT" атрибут. След това, указанията на W3C са ясни:

Когато съществува подходящ markup език, използвайте го вместо изображения, за да визуализирате информация

Това, по същество, значи **не използвайте изображения, за да представите текст** - посетителите с нарушено зрение са неспособни да променят размера на текста, който е представен чрез изображение. Ако все още не знаете как се променя размера на текста във вашият браузър, вижте:

<http://www.webcredible.co.uk/accessibility.shtml#resize-text>

Решението

CSS, както обикновено, идва на помощ. Увеличете размера на текста във вашия браузър. Хайде, направете го. Видяхте ли как табовете в менюто горе на екрана увеличиха размера на текста и пак **изглежда добре**. Днес, ще разберете как става това.

Ще започнем с един линк:

```
<a href="#">Home</a>
```

Ще приложим следния CSS код:


```
a { color: #000; background: #fb0; text-decoration: none }
```

Което ни дава:

Home

Има още работа, прави сте.

Добавяне на ляв ъгъл

Нуждаем се от малко изображение със същия цвят:  - ето едно, което направих по-рано. Да наречем този файл "left-tab.gif" и да го поставим като фон на този линк

```
a{
  color: #000;
  background: #fb0 url("left-tab.gif") left top no-repeat;
  text-decoration: none
}
```

Новият код казва, че изображението за фон "left-tab.gif" трябва да бъде в горния ляв ъгъл и не трябва да се повтаря. Резултатът?

Home

До тук добре, но трябва да преместим текста малко. Много е лесно с малко **padding**:


```
a {
  color: #000;
  background: #fb0 url("left-tab.gif") left top no-repeat;
  text-decoration: none;
  padding-left: 10px
}
```

Home

А сега десния ъгъл

Може да се задава само едно изображение за фон на таг, за това имаме нужда от нов таг и да зададем новото изображение. Можем да използваме:

```
<a href="#"><span>Home</span></a>
```

Сега само прилагаме това изображение за фон  (още едно, което направих по-рано) на `` и сме готови да започваме! Да наречем това изображение "right-tab.gif"

```
a span {
  background: url("right-tab.gif") right top no-repeat;
```

```
}
```

Този код означава, че всеки `` таг в `<a>` таг ще има това изображение за фон. И крайния резултат:

Home

Перфектно! Сега...изчакайте малко, можете ли да кажете защо не е перфектно? Точно така, забравихме да сложим малко **padding** на `` тага:

```
a span {  
    background: url("right-tab.gif") right top no-repeat;  
    padding-right: 10px;  
}
```

Което ни дава:

Home

Сега вече е перфектно! Променете размера на текста и вижте как изглежда!

Това е само началото

Тази статия засяга само основите на това, което можем да направим с CSS и навигация.

Превод: Iv0_b0y Източник: www.babailiica.com/articles

Java - Програмиране в нова светлина

Дълго време мислих как да представя Java езика ! От една страна това е не толкова известен език. От друга езика никак не е нов но хората продължават да не му обръщат внимание. После се сетих една мисъл на Дейвид Холмс - "Java е това, което ти искаш да бъде.". В такъв случай просто трябва да ви представя езика..... вие сами ще го разберете.

Първите слухове за Java бяха насочени към интернет. Платформено не зависимия език винаги ще бъде атракция и точно това направи Java. Славата на езика дойде от аpletите, който може да създавате с него. Aплета е малко приложение, което вие сваляте заедно с интернет страницата. Разбира се всеки човек би се замислил не е ли това опасно ? Не можели така без проблемно "тъмните" програмисти да направят вреда на компютъра ми ? Не ! Писането на аплети дава свободата да направите малко програма, която да се зарежда със страницата, но не ви дава възможността да пипате файлове на системата, на която е зареден аплета. С тези възможности езика започна своя възход в света на програмиста.

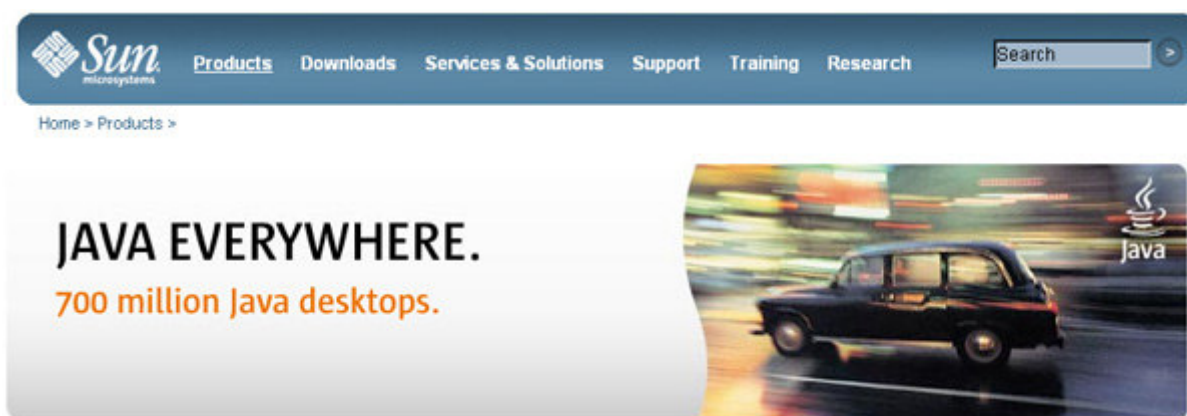
Но това заблуди хората, че Java е само за интернет програмиране. В същност езика може да се използва за всякакви среди. Без проблемно се справя с Win32 (вашия Windows), с Linux и с всички други среди, който може да се сетите. Но и това не е нещото, което да ви възхити. Още по интересното в случая е горе спомената платформено-независимост. Това Ви дава възможността като програмисти да пишете програмния код само веднъж, а след това

елементарно да го пренасяте за друга платформа. Това е малката магия, която изпреварва известния "C++". Дори и той не може да се сравнява в това отношение с Java.

Сега след като добихте представа, че Java е нещо интересно нека обясня как работи. За да работи една Java-програма на компютъра ви е нужно да си инсталирате виртуалната среда на Java. Сваля се напълно безплатно от сайта на създателите на Java, компанията SUN (www.sun.com). Това не е програма и няма нужда да я пускате или изключвате - само да я инсталирате. От тук на татък тя ще позволи на компютъра ви да разбира Java. След това Ви трябва SDK от същия сайт. Това е пакета, който ви е нужен за да програмирате. Но всеки добър програмист си има любима програма. Ще си позволя да ви препоръчам една, който използвам от дълги години - JCreator Pro. С една до две настройки ще превърнете тази програма в могъщ инструмент. Това е всичко, което ви е нужно за да станете елитен Java програмист.

Следващата стъпка разбира се е намирането на хубава книга. Тук съвета е труден защото всеки човек има любима книга. Затова аз ще ви препоръчам един автор, който е с толкова доказано име, че се съмнявам някой да възрази. Световно известния - Bruce Eckel, всяка негова книга ще ви превърне в мислещ програмист. А защо "мислещ" ? Ами това идва от специфичния начин, по който той кръщава книгите си. Тази която е нужна на вас е "Мислене в Java" (Thinking in Java). Това са началните неща, който ще ви "бутнат" в света на Java !!! От следващия брой очаквайте обширно въведение и първи стъпки в самото програмиране.

www.sun.com



(Гордите създатели на Java)

Автор: Стилиян Бешев (Diabolic_Soul)