***16-ma’ruza.*** **PHP dasturlash tilida satr va massivlar.**

**Reja:**

1. PHP dasturlash tilida satr va massivlar tipi.
2. Massivlar inisializasiyasi.

|  |  |
| --- | --- |
| **importance_of_education_1** | **O`quv modullari**  *PHPda massivlar, satrlar, massivlar inisializasiyasi, array* |

## Satrlar bilan ishlash.

World Wide Web birinchi o’rinda matnlar bilan ishlaydi. Faqatgina matnga bog’liq bo’lmagan ravishda WWW boshqa komponentalarga ham juda boy hisoblanadi. Shuning uchun ham PHPda juda ham ko’plab funkstiyalar mavjud bo’lib, ularning yordamida qatorlarni formatlash hamda ularni boshqarish mumkin.

Bu ma’ruza davomida asosan quyidagilarni o’rganiladi:

* qatorlar formatini berish;
* qator uzunligini aniqlash;
* qatordan satrni qidirish;
* satrni qismlarga ajratish;
* qator boshi va oxiridan bo’sh joylarni olib tashlash;
* qatorlar o’rnini almashtirish;
* qator registrlarini o’zgartirish;

## Qator formatini berish.

Ushbu ma’ruzaga qadar barcha qatorlarni brauzer oynasiga to’g’ridan-to’g’ri berildi. Biroq PHP tilida ekranga chiqarish jarayonida satrlar formatini berish uchun ikkita funkstiya mavjud. Bu funkstiyalar berilgan haqiqiy sonning verguldan keyingi belgilar sonini berilgan aniqlikgacha chegaralash hamda maydon ichidagi qatorni sozlash yoki ma’lumotlarni biror sistemaga bog’liq ravishda chiqarish imkonini beradi. Endi esa, quyida printf() va sprintf() funkstiyalarining xususiyatlari va imkoniyatlarini ko’rib o’tamiz.

## printf() funkstiyasi bilan ishlash.

Agar o’quvchi S tili bilan tanish bo’lsa, demak printf() funkstiyasi deyarli tanish hisoblanadi. Lekin S tilidagi Printf() funkstiya PHPdagi funkstiya bilan biroz farq qiladi. Bu funkstiyaga qatorli argument beriladi. Lekin printf() funkstiyasiga turli tipdagi qo’shimcha argumentlarni berish mumkin. Quyida keltirib o’tadigan misolda butun sonni haqiqiy ko’rinishda chiqarish uchun printf() funkstiyasidan foydalaniladi.

*printf(“Men kiritgan son %d”, 55);*

*// natija: Men kiritgan son 55*

Bu dasturda maxsus simvollarni boshqarish belgisidan foydalanildi.

Printf() funkstiyasiga beriladigan argument tipini o’zgartirish uchun maxsus prostent (%) belgisidan foydalaniladi. Argumentga beriladigan simvollarni ixtiyoriy tipga almashtirish mumkin. Buning uchun berilgan har bir argumentga alohida (%) tip belgisi berilishi lozim.

Quyida ikkita sonni ekranga chop qilishga oid misolni ko’rib o’tamiz:

*Printf(“Birinchi son: %d <br>\n Ikkinchi son: %d<br>\n”,55,66);*

*//natija:*

*// Birinchi son: 55*

*// Ikkinchi son: 66*

Misolda keltirilgan % belgisidan keyingi «d» simvoli argumentni o’nlik sanoq sistemasida chiqarish uchun ishlatiladi.

## printf() va tiplarni aniqlash.

Yuqorida tipni aniqlovchi «d» belgisi bilan tanishib o’tildi va uni ma’lumotlarni 10-lik formatda chiqarilishi ma’lum. Quyidagi jadvalda esa, qolgan tip aniqlovchi belgilar keltirilgan.

3.1-jadval. Tip aniqlovchi belgilar.

|  |  |
| --- | --- |
| **Aniqlovchi** | **Keltirilishi** |
| d | 10-lik sistemada qiymat qaytariladi. |
| b | 2-lik sistemada qiymat qaytariladi. |
| c | ASCII-ekvivalentida qiymat qaytariladi. |
| f | Haqiqiy sondan iborat qiymat qaytariladi. |
| o | 8-lik sistemada qiymat qaytariladi. |
| s | Satrli kattalikdan iborat belgi qaytariladi. |
| x | 16-lik sistemada qiymat qaytariladi. |
| X | 16-lik sistemada katta harfdagi qiymatlar qaytariladi. |

Quyida keltirib o’tiladigan dasturimizda bitta sonni chiqarish uchun misol keltiriladi. Shu sonni yuqorida (3.1-jadval) keltirib o’tilgan tiplar asosida chaqiriladi. Dasturda nafaqat tipi aniqlangan belgilar, balki boshqa matnni ham brauzerga printf() funkstiyasi yordamida chiqarish mumkin.

Misol 16.1. Tiplarni aniqlashga doir misol.

*<html>*

*<head>*

*<title>Tiplarni aniqlashga doir misol</title>*

*<haed>*

*<body>*

*<?php*

*$number=543;*

*printf ("O'nlik ko'rinish :%d<br>",$number);*

*printf ("Ikkilik ko'rinish :%d<br>",$number);*

*printf ("Haqiyqiy sonli ko'rinish :%f<br>",$number);*

*printf ("Sakkizlik ko'rinish :%o<br>",$number);*

*printf ("Qatorli ko'rinish :%s<br>",$number);*

*printf ("O'n oltilik ko'rinish (quyi registr)::%xd<br>",$number);*

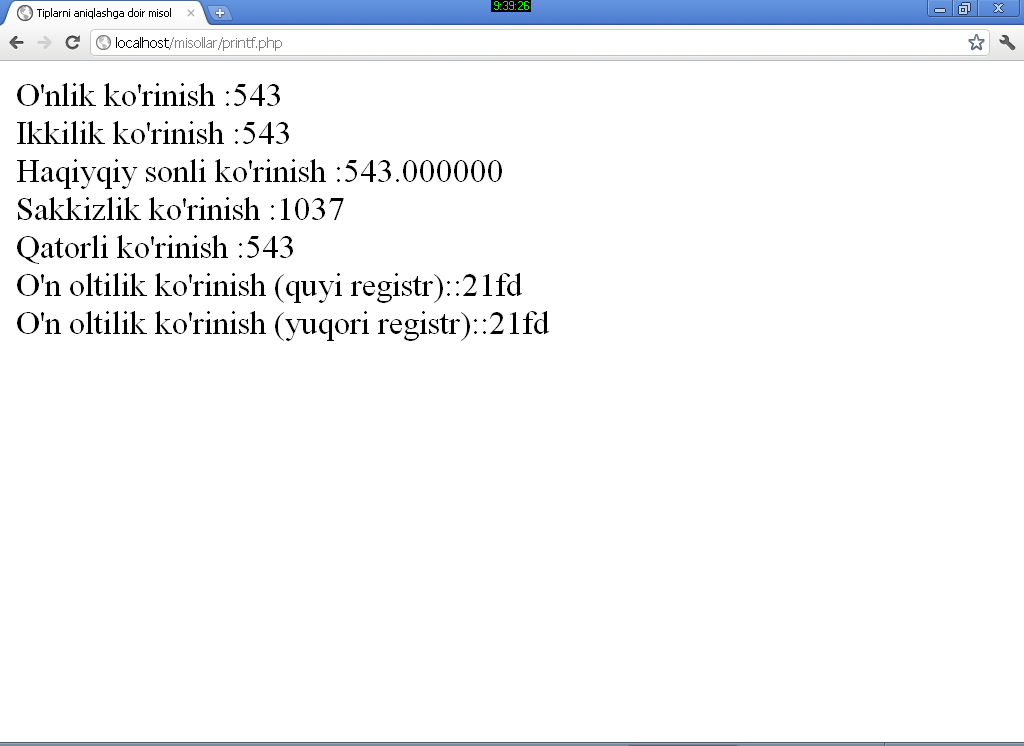
*printf ("O'n oltilik ko'rinish (yuqori registr)::%xd<br>",$number);*

*?>*

*</body>*

*</html>*

Quyida keltirib o’tilgan 5.1-rasmda esa, keltirib o’tilgan dasturning natijasini beriladi.



*16.1-rasm. 16.1-misolda keltirilgan dasturning natijasi.*

Printf() funkstiyasidan bir tipdan boshqa tipga o’tkazib simvollarni brauzerga chop qilishda foydalanish mumkin.

HTML da ranglarni berish uchun 16-lik sanoq sistemasidagi sonlardan foydalaniladi. Ular 00 dan FF gacha bo’lgan belgilardan iborat. 0..255 gacha bo’lgan sonlarni ham print() funkstiyasidan foydalanib 16-lik sanoq sistemasida ifodalash mumkin.

*$ Red=204 ;*

*$ Green=204;*

*$ Blue=204;*

*Print f (“# % x % x % x, $red,$green,$Blue);*

*//Natija: “#cccccc” buladi.*

Printf() funkstiyadan 10-lik sanoq sistemasidagi sonlarni 16-lik sanoq sistemasiga o’tkazish uchun ham foydalanishi mumkin. Shuningdek, brauzerga chop qilinayotgan belgi formatini, ya’ni maydon chegarasini ham berish mumkin.

## Qator formatini berish

Printf() funkstiyasida biz o’zimiz bergan argumentni formatini ham bera olishimiz mumkin. Bunda chiqish formati chegarasini % belgisidan keyin beriladi. Agar berilgan formatdan chiqadigon simvollar soni oz bo’lsa, shu sonlar oldida nol belgisi beriladi.

Masalan;

*Printf(“%04d”,36);*

*//Natija 0036 chikadi* .

Agar ekranga beriladigon simvol oldida bo’sh joy qolish lozim bo’lsa, u holda % belgisidan keyin bitta bo’sh joy tanlanadi. Masalan

*Printf(“%..4d”,36);*

*// Natija «..36»chikadi.*

Agar berilayotgan formatda nol yoki bo’sh joy o’rnida biror belgi chiqishi lozim bo’lsa, u holda % belgisidan keyin shu simvol qo’yish lozim:

*Printf(“%x4d”,36);*

*//Natija “xx36” buladi.*

*Endi biz ranglarni berishda ham format berishimiz mumkin.*

*$red=1;*

*$green=1;*

*$Blue=1; printf(“#02x%02x%02x”,$red,$green,$Blue);*

*//Natija ”#010101”buladi.*

Agar 2-ta simvol uchun kattaroq format berilsa, u holda albatta shu simvol oldida nol qiymatlari chiqadi.

## Chiqadigan qiymat uchun maydon kengligini berish.

Printf() funkstiyasida brauzerga chiqadigan simvollar uchun joy ajratish ham mumkin. Maydon kengligini esa, % belgisidan keyin butun qiymat berish orqali bajariladi. Quyida keltirib o’tmoqchi bo’lgan dasturimizda maydon uchun 20 ta joy ajratiladi.

Misol 16.2. Tiplarni aniqlashga doir misol.

*<html>*

*<head>*

*<title>Printf funksiyasiga doir misol</title>*

*<haed>*

*<body>*

*<?php*

*print "<pre>";*

*printf("%20s\n","Kitoblar");*

*printf ("%20s\n","CD");*

*printf("%20s\n","O'yin");*

*printf ("%20s\n","Jurnal");*

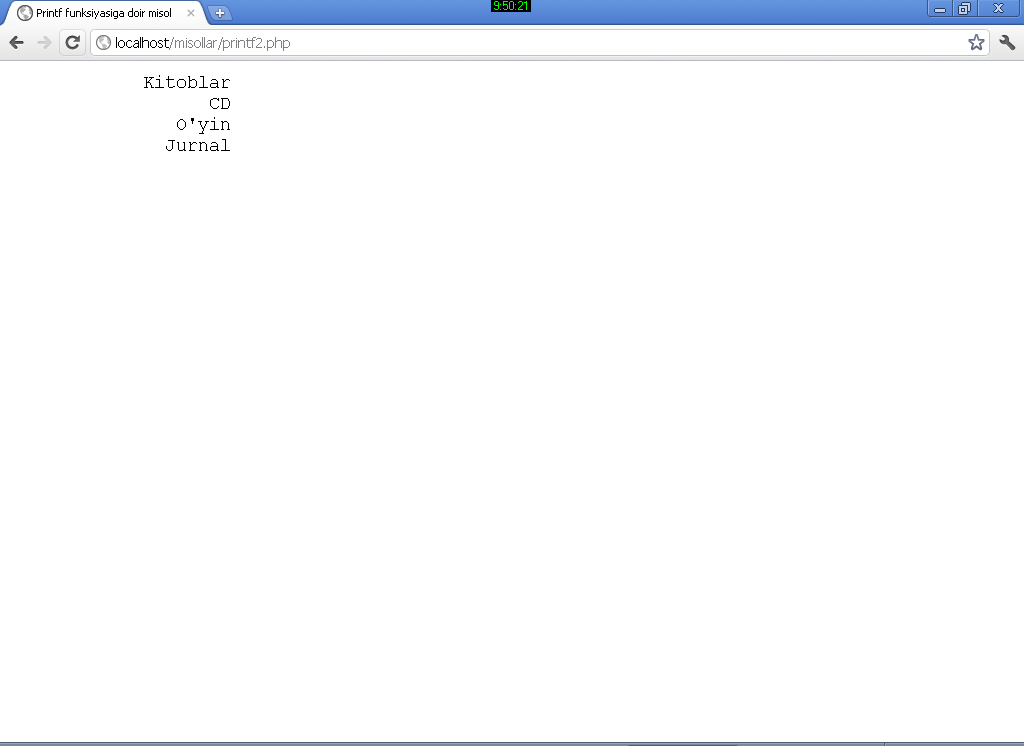
*print "</pre>";*

*?>*

*</body>*

*</html>*

Quyidagi rasmda ushbu dasturning natijasi keltiriladi.



*16.2-rasm. Maydon kengligini aniqlash yordamida matnni chapga tekislash.*

Aktiv holda esa, chiqadigan matn o’ng tomonga tekislanadi. Format berganda matnni chap tomonga tekislash uchun esa, minus (-) belgisidan foydalaniladi.

Masalan,

*Printf “%-30s\h”, “chapdan tekislash”;*

## Haqiqiy sonlarni aniqligini berish

Agar xakikiy sonlarni brauzerga chop kilmokchi bo’lsak,biz uni verguldan keyingi rakamlarini belgilab (chegaralab) quyishimiz mumkin.Lekin bunday formatni berish fakat “f” belgisi bilan ishlagandagina mumkin. Masalan,

*printf(“%2f”, 5.333333),*

*// Natija 5,33 chikadi.*

Yuqoridagi fikrlarimizni yanada mustahkamlash maqsadida mahsulotlar va ularning narxi, ro’yxatini chiqarish uchun printf() funkstiyasidan foydalanishga doir misolni ko’rib o’taylik.

Misol 16.3 Tovar va ularning narxi ro’yxatini chiqarish uchun printf() funkstiyasidan foydalanish.

*<html>*

*<head>*

*<title>* Tovar va ularning narxi ro’yxatini chikarish uchun printf() funkstiyasidan foydalanish

*</title>*

*<head>*

*<body>*

*<?php*

*$products=array("creen armchair"=>222.4,*

*"candlestick"=>4,*

*"coffee table"=>80.6*

*);*

*printf ("</pre>");*

*printf ("$-20s%23s\n","íîìëàíèøè ","íàðõè");*

*printf ("$'-43s\n","");*

*printf ("</pre>");*

*foreach($products as $key=>$val)*

*printf ("$-20s%-20.2f\n", $key ,$val);*

*printf ("</pre>");*

*?>*

*</body>*

*</html>*

Bu dasturni bajarilishi natijasida esa ekranda quyidagi ro’yxat hosil bo’ladi.



*16.3-rasm. Printf() funkstiyasi yordamida formatlangan tovar va narxlar ro’yxati.*

## Qatorlarni formatlash va saqlab turish

printf() funkstiyasi natijani brauzer oynasiga chiqarib beradi, lekin o’zgargan tipdan dastur davomida foydalanib bo’lmaydi. Chunki, tip faqat ekranga chiqarishda o’zgartiriladi. PNRda printf() funkstiyasi kabi foydalaniladigan shunday funkstiya borki, u sprintf() deb nomlanadi. Bu funkstiya printf() funkstiyasi bilan vazifasi bir xil undan farqi berilgan argument qiymatini o’zgaruvchiga saqlab, uni kelgusida (dastur davomida) foydalanish mumkin. Quyidagi misolda sprintf() funkstiyasidan foydalaniladi. Bu misolda qiymat formati “%.2f” bilan beriladi va uni $dosh o’zgaruvchisida saqlanadi.

*<?php*

*$dosh=sprintf("%.2f", 2.334454);*

*print "Sizda $dosh son qoldi";*

*?>*

sprintf() funkstiyasidan ma’lumotlarni faylga yozishda ham foydalanish mumkin. Buning uchun sprintf() funkstiyasi o’zlashtirilgan o’zgaruvchini fputs() funkstiya yordamida faylga berish kifoya.

## Qatorlarni tekshirish

Dasturchi har doim ishlatayotgan ma’lumotlari to’g’risida to’liq ma’lumotga ega bo’lmaydi. Qatorlar turli xil manbalardan yoki foydalanuvchi tomonidan berilishi mumkin.Tashqi manbalardan keladigan ma’lumotlar bilan ishlashdan avval, bu ma’lumotlar haqida ma’lumotga ega bo’lishga to’g’ri keladi. PHP tilida qatorlar haqida ma’lumot beruvchi juda ham ko’plab funkstiyalar mavjud. Shuningdek, qatorlar sifatida tasavvur qilish ham mumkin. Masalan, qatordagi har bir elementga (simvolga) alohida massiv elementi kabi murojat qilish mumkin.

*$test=”sgallywag”;*

*print $test[0] ;//Natija “s”*

*print $test[2]; // Natija “a”*

Demak, simvollarni rakamlash “0” dan boshlanadi

## strlen() funkstiyasi yordamida qator uzunligini aniqlash

strlen() funkstiyasi qator uzunligini aniqlash uchun foydalaniladi. Bu funkstiya qator argumentini oladi va simvollar sonini qiymatda qaytaradi.

strlen() funkstiyasi ko’proq foydalanuvchi tomonidan kiritilgan simvollar sonini aniqlashda ishlatiladi. Quyidagi misol yordamida $misol o’zgaruvchisiga kiritilgan simvol sonini tekshiraylik. Agar simvol 4 ta bo’lsa “rahmat”, aks holda keyingi ma’lumotlar berilsin.

*if(strlen($misol) == 4){*

*print("rahmat") ;}*

*else*

*{print("Kiritilgan belgi 4 ta simvol bo’lishi lerak");}*

## strstr() funkstiyasi yordamida qatordan belgilarni aniqlash

Bu funkstiya yordamida berilgan qatorda izlanayotgan belgi bor yoki yo’qligini aniqlash mumkin. strstr() funkstiyasi ikkita argument talab qiladi. Bular berilgan qator va qidirilayotgan belgilar. Agar qidirilayotgan belgi qatordan topilmasa false, aks holda topilgan belgi qaytariladi. Masalan,

*$ms= "pAB7";*

*if (strstr($ms, "AB"))*

*print "Raxmat, topildi";*

*else*

*print "Topilmadi";*

Bu misolda qidirilayotgan «AV» simvollar “pAB7” qatordan topiladi. Agar «AV» o’rniga «rav7»ni qidirilsa, strstr() funkstiyasi false natija qaytaradi. Bunga sabab, strstr() funkstiyasi belgilarni katta-kichikligini ajratadi. Agar harflarni katta yoki kichikligiga ahamiyat bermay qidirilsa, u holda stristr() funkstiyasidan foydalanish maqsadga muvofiq bo’ladi.

## strpos() funkstiyasi yordamida qidirilayotgan simvol o’rnini aniqlash

strpos() funkstiyasi nafaqat qatordan belgini qidiradi, balki shu belgini qatordargi o’rnini ham aniqlab beradi. Bu funkstiyaga ikkita argument beriladi, bular: berilgan qator va qidirilayotgan belgilar. Yana funkstiyaga butun son oluvchi uchinchi argument ham berish mumkin. Agar qatordan qidirilayotgan belgi topilmasa false, aks holda belgi nomeri qaytariladi. Quyida keltirib o’tiladigan misolda qator “mz” simvollari bilan boshlanganligini tekshirish uchun strpos() funkstiyasidan foydalaniladi.

*$ms="mzooxyz";*

*if (strpos($ms,"mz")===0)*

*print "Salom mz!";*

Dasturda foydalanilgan “= = =” belgisi, shu belgining chap va o’ng tomonidagi belgilar bir xil tipga ega va o’zaro tengligini tekshirish uchun ishlatiladi.

## substr() funkstiyasi yordamida qatordan belgilarni qirqib olish

Substr() funkstiyasi berilgan pozistiyadan belgilangan pozistiyagacha bo’lgan simvollarni qirqib oladi va qaytaradi. Funkstiyaga ikkita argument beriladi, bular: berilgan qator va qidiriladigan qator. Bundan tashqari uchinchi argument ham bo’lib, unda simvollar sonini berish mumkin. Lekin bu shart bo’lmagan parametr hisoblanadi. Agar bu argument berilmasa, Substr() funkstiyasi berilgan pozistiyadan qator oxirigacha bo’lgan simvollarni hammasini qaytaradi. Masalan:

*$test="Toshkent";*

*print substr($test,4)."<br>"; // natija “kent” bo’ladi*

*print substr($test,4,3); // natija “ken” bo’ladi*

Agar funkstiyaga beriladigan ikkinchi argumentga manfiy son berilsa, u holda simvollarni qidirish chapdan emas, balki o’ng tomondan boshlanadi. Quyida keltirib o’tiladigan misolda qatordan “.uz” simvoli oxirgi pozistiyadan boshlab qidiriladi va ekranga natija sifatida beriladi.

## $test = "Nammqi.uz";

## if (substr($test,-3)== ".uz")

## print "adres.uz da";

## else print "adres.uz da emas";

## strtok() funkstiyasi yordamida qatorlarni belgilash

strtok() funkstiyasi yordamida qatorni so’zlarga ajratish mumkin. Funkstiyaga ikkita argument beriladi, bular: bo’linadigan va bo’luvchi qator. Bu erda bo’luvchi qator ixtiyoriy simvol bo’lishi mumkin. strtok() funkstiyasi birinchi topilgan so’zni qaytaradi. Bir marta ishlatilganda funkstiya birinchi so’zni qaytaradi, shuning uchun bu funkstiya ko’pincha stiklda ishlatiladi.

Fikrimizni yanada mustahkamlash maqsadida quyidagi dasturni ko’rib o’taylik.

Misol 16.4. strtok() funkstiyasi yordamida satrni so’zlarga ajratish.

*<html>*

*<head>*

*<title> strtok() funksiyasi yordamida satrni so'zlarga ajratish</title>*

*<haed>*

*<body>*

*<?php*

*$text = "http://www.deja.com/qs.xp?Op=dnquery.xp$st=MS$DBS=2&QRY=developer+php";*

*$delims = "?&";*

*$word=strtok($text,$delims);*

*while (is\_string($word))*

*{*

*if ($word)*

*print "$word <br>";*

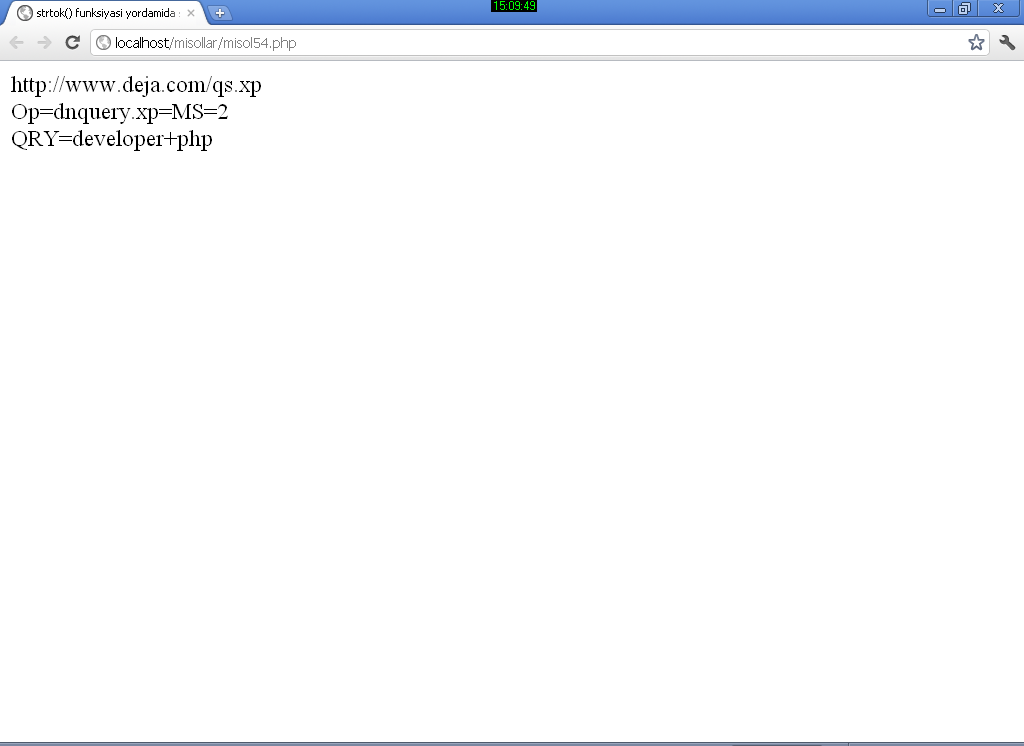
*$word=strtok($delims);*

*}*

*?>*

*</body>*

*</html>*

**

*16.4-rasm.*

Bu dasturda strtok() funkstiyasi URL - adresdagi so’zlarni ajratib olish uchun ishlatiladi.

Strtok() funkstiyasidan foydalanish biroz noqulayroq hisoblanadi, chunki bu funkstiyani ishlatish uchun bir necha usullarni bilish talab qilinadi.

## Trim() va ltrim()funkstiyalari yordamida qatordagi bo’shliqlarni tozalash

Ko’pincha fayldan o’qilgan yoki foydalanuvchi kiritgan matn boshi yoki oxirida bo’sh joy (probel) yo’qligini aniqlash qiyin. Trim() funkstiya qator boshi va oxiridagi bo’shliqlarni, barcha bo’shliqlarni va tabulyastiya belgilarni olib tashlaydi. Bu funkstiya bo’shliqlardan tozalangan qatorni qaytaradi.

Masalan:

*$text = "\t\tyest";*

*$text = trim($text);*

*print $text; // natija “yest”*

Ba’zan bir qatordagi barcha bo’shliqlarni tozalash shart emas. Shunda, qator boshidagini qoldirib oxiridagi bo’shliqni o’chirish mumkin. Xuddi Trim() kabi ishlovchi chop() funkstiyasidan foydalanish mumkin. Bu funkstiya xuddi Trim() kabi ishlaydi. Lekin faqat qator oxiridagi bo’shliqni o’chiradi.

*$text=”\t\tni tozalash ”;*

*$text=chop($text);*

*print $text;*

*//natija“ni tozalash”*

## substr\_replace() funkstiyasi yordamida satr qismini almashtirish

Substr\_replace() funkstiyasi substr() funkstiyasiga o’xshaydi. Undan asosiy farqi, bu funkstiyada satrni ixtiyoriy qismini o’zgartirish imkonini beradi. Bu funkstiyaga uchta argument beriladi, bular: qayta ishlanadigan qator, qo’shiladigan matn va boshlang’ich indeks. Bulardan tashqari, zarur bo’lmagan to’rtinchi argument ham mavjudki, u qirqib olinadigan qator uzunligini beriladi. Substr\_replace() funkstiyasi berilgan matnni ko’rsatilgan indeks bo’yicha mavjud qatorga qo’shadi va belgilangan satrni mavjud qatordan olib tashlaydi. Natija sifatida esa, yangi qo’shilgan satr bilan birgalikda yangi qator qaytaradi.

Quyida keltirib o’tilgan misolda o’rtadagi ikkita raqamni o’zgartirish keltirib o’tiladi:

*<?*

*$kod = "mx99xyz";*

*$kod = substr\_replace($kod,"00", 2, 2);*

*print "Yangi hosil bo'lgan son: $kod"; // natija "Yangi hosil bo'lgan son: $kod”*

*?>*

## str\_replace() funkstiyasi yordamida butun satrni almashtirish

str\_replace() funkstiyasi ko’rsatilgan barcha satrni yangi satrga almashtirish uchun foydalaniladi. Bu funkstiya uchta argument talab qiladi, bular: qidirilayotgan satr, o’zgaradigan satr va qo’shiladigan satr. Quyida keltirib o’tadilagan misolda str\_replace() funkstiyasi 1999 raqamini 2000 ga almashtirishga oid ish keltirib o’tiladi:

*$string= "All right reserved, 1999";*

*$string.= "Hamma himoyalar mavjud, 2000";*

*print str\_replace("1999", "2000", $string);*

## Registrni o’zgartirish

PHPda foydalanuvchi tomonidan registrdagi satrni o’zgartirish imkonini beruvchi bir nechta funkstiyalar mavjud. Ma’lumotlar bazasiga ma’lumotlarni yozilganda keyin ularni qidirib topishda yoki solishtirishda qiyinchilik tug’ilmasligi uchun hamma simvollarni katta yoki kichik harflarga o’tkazib qo’yish lozim bo’ladi. strtoupper() funkstiyasi kichik harflarni katta harflarga almashtirish uchun foydalaniladi. Bu funkstiya faqat bitta argument talab qilib, u ham bo’lsa katta harfga almashtiriladigan satr hisoblanadi.

*$membership = "mz99xyz";*

*$membership=strtoupper($membership);*

*print "$membership <p>"; // natija “MZ99XYZ”*

Aks holda, kichik harflarni katta harflarga almashtirish uchun esa, strtolower() funkstiyasidan foydalaniladi. Bu funkstiya ham yuqoridagi funkstiya kabi bitta argumentni oladi va natija sifatida kichik harfdagi satrni qaytaradi.

*$home\_url=*[*WWW.CORROSIVE.CO.UK*](http://WWW.CORROSIVE.CO.UK)*;*

*$home\_url=strtolower($home\_url);*

*if (! (Strpos($home\_url,”http://”)===))*

*$home\_url=”http:// $home\_url”;*

*print $home\_url;*

*// natija “*[*www.corrosive.co.uk*](http://WWW.CORROSIVE.CO.UK)*”*

Shuningdek, PHPda shunday funkstiya ham mavjudki u berilgan satrdagi barcha so’zlarni bosh harfini katta harfga aylantirib berish uchun ishlatiladi. Bu funkstiya ucword() hisoblanadi. Bu funkstiyaga oid quyidagi misolni ko’rib o’taylik:

*$full\_name="salom do'stim odiljon";*

*$full\_name=ucwords($full\_name);*

*print $full\_name; // natija ”Salom Do’stim Odiljon” chikadi*

## explode() funkstiyasi yordamida qatorlarni massivlarga ajratish

explode() funkstiyasining nomi qaysidir tomondan strtok() funkstiyasiga yaqinga o’xshab ketadi. Bu funkstiyaning asosiy farqi esa, u ajratib olingan satrni massivga tashlaydi. Keyin esa, bu massivdagi satr ustida ixtiyoriy amallarni bajarish mumkin. explode() funkstiyasi asosan ikkita argument talab qiladi, bular: qatordan ajratib olinuvchi satr va berilgan qator. Qatordan ajratib olinadigan satrda bitta simvol bo’lishi ham mumkin. Quyidagi keltirib o’tiladigan misolda sana elementlar bo’laklab olinadi va massivga yoziladi:

*$start\_date="2015-01-10";*

*$date\_array=explode("-", $start\_date);*

*echo $date\_array[0];*

*//$$date\_array[0]= “2015”*

*//$$date\_array[1]= “01”*

*//$$date\_array[2]= “10”*

# Massivlar. Bir va ko’p o’lchovli massivlar.

Massiv va uning mexanizmidan foydalanish PHP dasturlash tilining unumdorligi va samaradorligini yanada oshiradi. Massiv bilan ishlashda murakkab strukturaga ega ma’lumotlarni saqlash va qayta ishlash imkoniyatiga ega bo’lish mumkin.

Asosan quyidagilar o’rganiladi:

1. Massiv nima va ularni qanday yaratiladi?
2. Massiv elementiga qanday murojat kilinadi?
3. Massivdagi ma’lumotlar qanday saralanadi?

## Massiv nima?

O’zgaruvchi nima ekanligi bizga ma’lum. O’zgaruvchi asosida ma’lumotni saqlash, qayta ishlash va chop qilish mumkin. Lekin o’zgaruvchida bir vaqtning o’zida faqat bitta ma’lumot saqlash mumkin. Shunday o’zgaruvchilar mavjudki, dasturchini bunday chegaralanishlardan ozod qila oladi. Massiv xuddi shunday imkoniyatga ega bulib, u bir nomda bir necha ma’lumotni, xattoki katta hajmdagi ma’lumotlarni o’zida saqlab tura oladi. Massivdagi har bir qiymat esa alohida raqam yoki matnni qator yordamida ko’rsatiladi.

***Yangi termin.*** Massiv – bu bitta nomda saqlanuvchi qiymatlar to’plami. Bu qiymatlarning har biriga uning indeks nomeri orqali yoki matnni qatori orqali murojaat qilish mumkin.

Agar 5 ta qiymatdan foydalanmoqchi bo’linsa, unga mos yana 5 ta o’zgaruvchi ham yarata olinadiku. Unda massivni nima zarurati bor? Massiv birinchi o’rinda ixchamlik uchun zarur. Ikkinchidan, dasturchini xohishiga qarab ikkita yoki minglab qiymatlarni yangi o’zgaruvchilarni yaratmay saqlay olish mumkin. Uchinchidan esa, ularni stikl asosida qurish, ixtiyoriy qiymatni tanlash, alfavit bo’yicha saralash hatto dasturchi xohlaganday saralab olib foydalana olish mumkin.

Massivning har bir elementiga uning indeksi asosida murojaat qilinadi. Indeks massiv nomeri yoki matnli qator bo’lishi mumkin.

Massiv elementi raqam bo’yicha ko’rsatiladi va u noldan boshlanadi. Shuning uchun unutmaslik lozimki, massivning oxirgi element nomeri massivdagi elementlar (qiymatlar) sonidan bitta kam bo’ladi. Masalan, quyidagi jadvalda users massivining 4 ta elementi keltirilgan. Bunda uchinchi element indeksi 2 ga teng bo’ladi .

Users massivi

|  |  |
| --- | --- |
| indeks | qiymati |
| 0  1  2  3 | Bort  Sharon  Batty  Harry |

Matnli qator yordamida massiv elementini ko’rsatish asosan, bir vaqtning o’zida qiymat va kattalik nomi saqlanayotganda qulay bo’ladi.

PHP tilida nom va nomeri bo’yicha elementlarni ko’rsatish va ishlash uchun massivni murakkabrok shakllari mavjud. Bular haqida keyingi ma’ruzalarda batafsil to’xtalib o’tiladi.

## Massivlarni yaratish

Oddiy holatda massiv – bu nomerlangan tartib bo’yicha joylashgan qiymatlar ro’yxati. Qiymatlar massiv elementlariga to’g’ridan-to’g’ri yoki array() funkstiyasi yordamida o’zlashtirilishi mumkin. Bu usullarning ikkalasi bilan ham tanishib chikaylik.

## array() funkstiyasi yordamida massivni aniqlash

array() funkstiyasidan massiv elementiga birdaniga bir necha qiymatlarni o’zlashtirishda foydalanish mumkin. Masalan,

*$users=array (“Xamid”, “Sobir”, “Botir”, “Nosir”);*

Endi uning nomeri bo’yicha massiv elementiga murojaat qilamiz:

*print “$users[2]”;*

Natijada esa, brauzer oynasiga “Botir” yozuvi beriladi. Massivlar bilan ishlashda massiv indeksi noldan boshlanishini esdan chiqarmaslik lozim.

## Identifikator yordamida massiv elementlarini yaratish

Massiv identifikatori yordamida yangi massiv yaratish yoki mavjud massivga yangi element qo’shish imkoniyati mavjud. Buning uchun massiv nomi va [] qavslarni berish lozim. Mustahkamlash uchun yana bir marta shu usul bilan $users massivini yaratamiz:

*$users[] =“Akmal”;*

*$users[] = “Salimn”;*

*$users[] = “Nodir”;*

*$users[] = “Komil”;*

Bu erda [] ichida element indekslari beriladi. PHP esa, avtomatik ravishda ularni sanab keyingi element nomeri bo’lishini aniqlab qo’yadi.

Dasturchi elementlar nomerini ko’rsatishi ham mumkin, faqat bu juda ham ehtiyotkorlikni talab qiladi. Quyidagi misolni ko’raylik:

*$users[0]= “Olim”; $users[200]= “Ikrom”;*

massivda faqat ikkita element mavjud, lekin oxirgi nomer 200 ga teng. Massivni oradagi elementlar hozircha yo’q. Massiv yaratilgandan so’ng esa, unga yangi elementlarni qo’shish mumkin.

Quyidagi misolda esa, array() funkstiyasi yordamida massiv yaratiladi va unga yangi element qo’shiladi:

## $users = array("Halil", "Kolil", "Sohib", "Toxir");

## $users[]= "Sobit";

## echo $users[4]; // Sobit chop etiladi

## Strukturali (assostirovannыy) massiv

Massiv elementiga nomeri bo’yicha murojaat qilish ya’ni, qanday yaratilgan bo’lsa shundayligicha foydalanish oson bo’ladi. Ba’zan massiv elementiga uning nomi bo’yicha murojaat qilishga ham to’g’ri keladi. Strukturali massivlarda element nomeri bo’yicha emas, balki nomi bo’yicha ko’rsatiladi.

*Yangi termin.* Strukturali massiv – bu massivni, uning elementiga nomi bo’yicha murojaat qilish mumkin. Boshqa dasturlash tillarida ham bu massivni struktura deb ataladi.

Strukturali massivni ham to’g’ridan-to’g’ri yoki array() funkstiyasi yordamida yaratish imkoni mavjud. PHP tilida bu massivlarni (strukturali va oddiy massivlar) alohida katta farqqa ega emas. Faqat ularga murojaat qilish boshqacharoq va ular bir-biridan farqli ularoq usul talab qiladi.

## array() funkstiyai yordamida strukturali massivni yaratish

array() funkstiyasi yordamida strukturali massivni yaratish uchun har bir element nomi va qiymatini berish lozim. Buni quyidagi misolda ko’rib o’tamiz. Unda 4 ta elementdan iborat $character nomli strukturali massiv yaratilsin:

*$character=array(Name=> “Komil”, occupation=> “Aqilli”, age=>30, “special power”=> “x-ray vision”)*

Endi massivning ixtiyoriy elementiga murojaat qilish mumkin;

*print $character[age]; // 30 chop etiladi*

Massiv elementi nomi - bu qatorlar bo’lib, agar bir nechta so’zdan iborat bo’lmasa, ularni qo’shtirnoq ichiga olish shart emas.

## Strukturali massivni to’g’ridan-to’g’ri yaratish

Yangi massiv yaratish yoki massivga nom va qiymat qo’shish uchun nomi bo’yicha massivni ko’rsatib, shu elementga qiymatni o’zlashtirish lozim. Masalan, yana bir $character massiv yarataylik

$character[name]= “Komil”,

$character[occupation]= “Aqlli”,

$character[age]=30,

$character[“special power”]= “x-ray vision”

## Ko’p o’lchovli massivlar

Ma’lumki, massiv elementlari biror qiymatga ega bo’lishi mumkinligini avvalgi ma’ruzada qarab o’tildi. Masalan, yuqoridagi misolida uchta element qator va bitta element butun sondan iborat. Massivlar ba’zan murakkabroq ko’rinishga ega bo’lishi ham mumkin. Massivlar qiymatlardan, ob’ektlardan yoki hatto massivlarni o’zidan ham tashkil topgan bo’lishi mumkin. Ko’p o’lchovli massivlar aslida massivlardan tashkil topgan bo’ladi. Ularning har bir elementi massivdan iborat bo’ladi. Masalan, birinchi massivning ikkinchi elementiga murojaat qilish uchun uni quyidagicha yozish lozim bo’ladi:

*$array[1][2]*

***Yangi termin.*** Ko’p o’lchovli massiv bu - har bir elementi massiv hisoblangan massivdir.

Ko’p o’lchovli massivlardan foydalanish malumotlarni murakkab strukturasini yaratish imkonini beradi. Masalan, har bir elementi strukturali massiv hisoblangan massiv yarataylik.

*Misol 16.6. Ko’p o’lchovli massiv yaratish*

*<html>*

*<head>*

*<title>Ko'p o'lchovli massiv yaratish </title>*

*</head>*

*<body>*

*<?php*

*$characters=array(array(name=>"Komil",*

*occupation=>"Superqahramon",*

*age=>30,*

*speciality=>"x-ray vision"),*

*array(name=>"Avaz",*

*occupation=>"Superqahramon",*

*age=>24,*

*speciality=>"superhuman strangth"),*

*array(name=>"Malika",*

*occupation=>"arch villiain",*

*age=>63,*

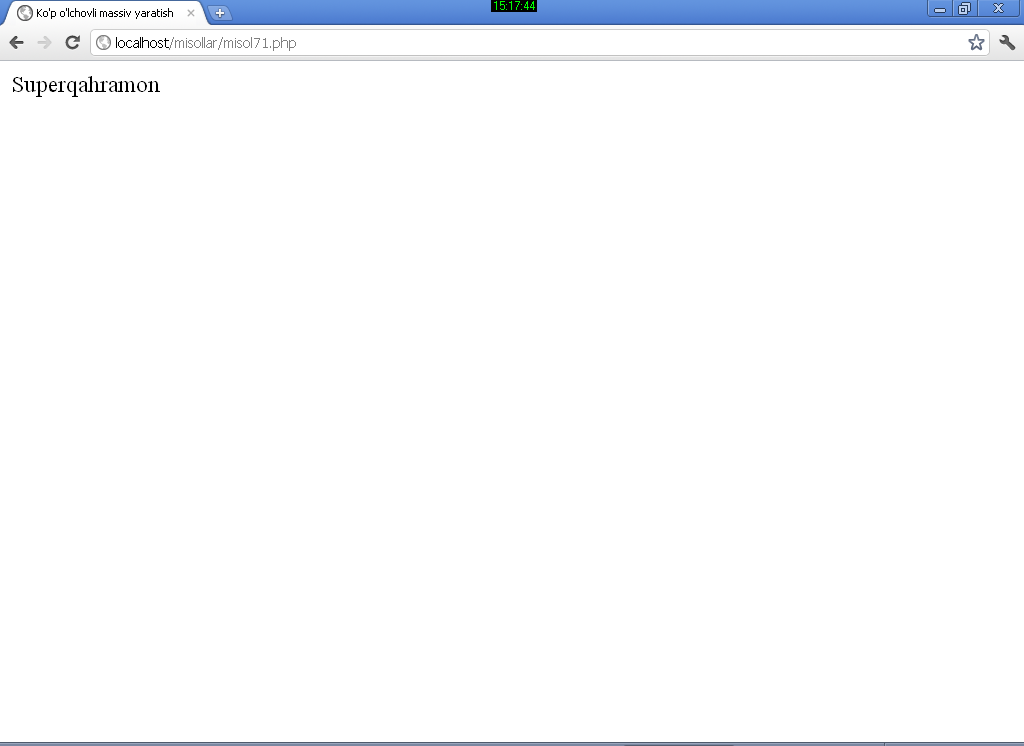
*speciality=>"nanotexnologiya"));*

*print $characters[0][occupation]; // Superqahramon chop etiladi*

*?>*

*</body>*

*</html>*

**

*16.6.-rasm.*

Agar misolga ahamiyat bersangiz, array() massivi ichma –ich ishlatilmoqda. Chunki, birinchi massiv yaratildi, keyin esa, xuddi shu massiv ichidagi element uchun ham ichki massiv yaratildi, so’ng xuddi shu massiv ichidagi element uchun ham ichki massiv yaratiladi.

U holda $character[2][name] massiv foydalanuvchiga «mary» natijani, $character[2][age] esa, 63 natijani kaytaradi.

Bu dasturni mantiqiy tahlil qilib, aytish mumkinki, oddiy, strukturali yoki murakkab kombinastiyali massivlarni yaratish imkoni mavjud.

# Massivlar bilan ishlash Massivlar ustida amallar.

Avvalgi ma’ruzada massivlarni yaratish usullari va mavjud massivga element qo’yishni ko’rib chiqildi. Endi esa, massiv xaqida ma’lumot olish va uning elementiga murojaat qilish uchun PHPning qo’shimcha imkoniyatlari bilan tanishamiz.

## Massiv hajmini aniqlash

Massivning ixtiyoriy elementiga uning nomeri asosida murojaat qilish mumkin: print $user[4]. Lekin massivlar bilan ishlash mexanizmi o’ta ixcham bo’lib, ba’zan massiv elementi nechtaligini ham unutib qo’yish mumkin. Xuddi shunday vaqtda count() funkstiyasi yordam beradi. Bu funkstiya massiv elementlar soni nechtaligini aniqlab beradi. Quyidagi misolda massivning oxirgi elementiga murojaat qilish uchun shu funkstiyadan foydalaniladi:

*$user=array(“Bert”,”Sharon”, “Betty”, “Harry”); print $users[cout($users)-1];*

Agar keltirib o’tilgan buyruqqa ahamiyat bersangiz, element nomeridan birini ayirilyapti, chunki massiv elementlari noldan boshlab sanaladi. Ya’ni indeks nomeri elementlar soniga teng emas.

## Sikl yordamida massivni ko’rish

Цiklda massivning barcha elementlarini ko’rishning juda ham ko’plab usullari mavjud. Ma’ruzada bu usullardan kengroq tarqalgan foreach instrukstiyasini ko’rib o’tamiz. Bu instrukstiya PHP tilini 4-versiyasida paydo bo’lgan.

Agar indeksli massivga ega bo’lsak, u holda foreach instrukstiyasidan quyidagicha foydalanish mumkin:

*Foreach ($array as $ tmp)*

*{*

*//…*

*}*

bu erda $array - massiv nomi, $tmp - har bir elementning qiymati vaqtincha saqlanadigan o’zgaruvchi.

*Misol 16.7. Massiv elemenlarini qo’shish dasturida quyidagicha bajariladi:*

*<?php*

*$users=array("Komil", "Salim", "Umid", "Murod");*

*foreach ($users as $val)*

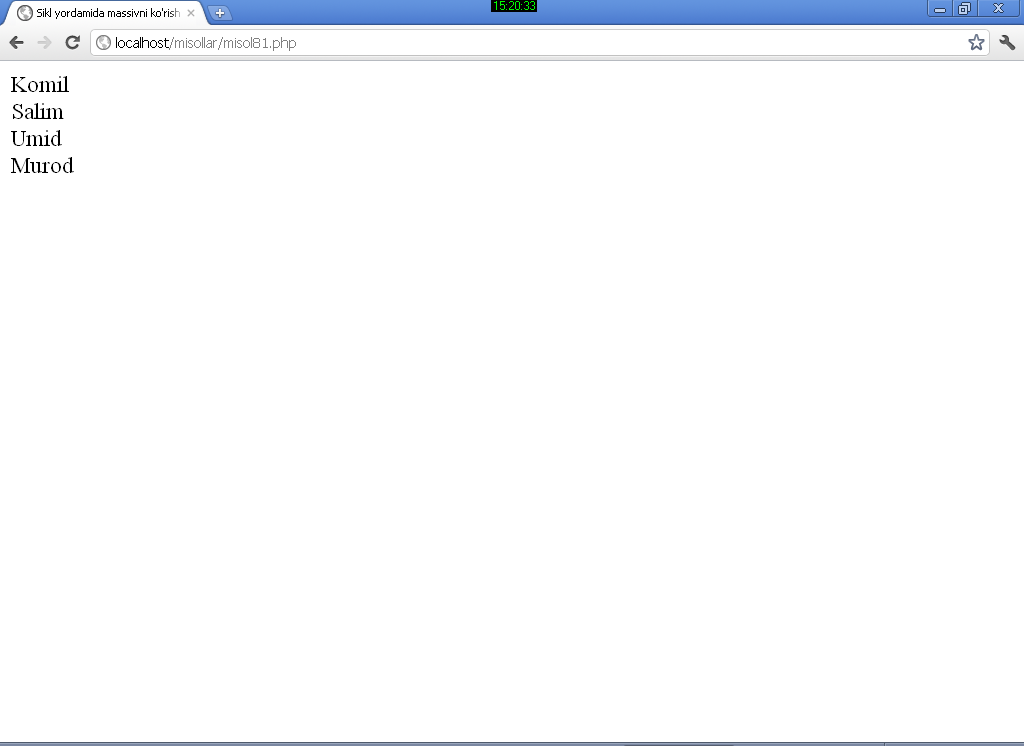
*{*

*print "$val<br>";*

*}*

*?>*

Bu dasturda massiv elementlari vaqtincha $ tmp o’zgaruvchisiga o’zlashtiriladi va brauzer oynasiga chop qilinadi.



*16.7.-rasm.*

## Strukturali massivni stiklda ko’rish

Strukturali massivni stiklda ko’rish uchun , foreach instrukstiyasini biroz boshqacharoq usulda keltiriladi. Bu hol instrukstiya quyidagicha beriladi:

*Foreach($ array as $ key⇒$ value)*

*{*

*//…*

*}*

bu erda $array – massiv nomi, $key- massivning har bir elementi nomi saqlanadigan o’zgaruvchi, $value - har bir element qiymati vaqtincha saqlanadigan o’zgaruvchi. Fikrlarni yanada mustahkamlash maqsadida quyidagi strukturali massivni ko’rish dasturini ko’rib o’taylik.

*Misol 16.8. Strukturali massivni ko’rish.*

*<html>*

*<head>*

*<title> Strukturali massivni ko'rish </title>*

*</head>*

*<body>*

*<?php*

*$character=array(name=>"Akmal",occupation=>"superqahramon", age=>30, "special power"=>"x-ray vision");*

*foreach($character as $key=>$val)*

*{*

*print $key="$val<br>";*

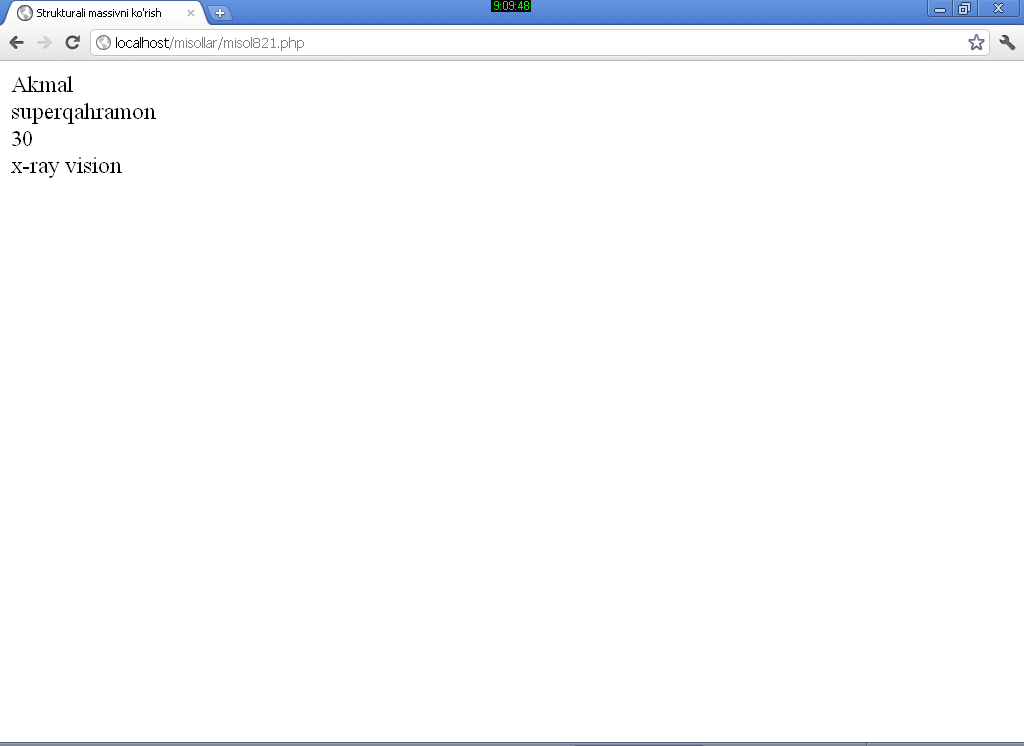
*}*

*?>*

*</body>*

*</html>*

Dasturni ishlatish natijasida esa quyidagi natijani olamiz:



*Rasm 16.8. Dasturdan olingan natija.*

## Ko’p o’lchovli massivlarni bruzerga chop qilish.

Endi ko’p o’lchovli massivlarni yuqorida keltirib o’tilgan usullar bilan brauzer oynasiga chop qilish mumkin. 7.1-misolda keltirilgan dasturdagi massiv natijasini quyidagi misol yordamida chop qilaylik:

*Misol 16.8. Ko’p ulchovli massivni stiklda kurish*

*<html>*

*<head>*

*<title> Ko’p o’lchovli massivni stiklda ko’rish </title>*

*</head>*

*<body>*

*<?php*

*$characters=array(array(name=>”bob”,*

*occupation=>”superhero”,*

*age=>30,*

*speciality=>”x-ray vision”),*

*array(name=>”sally”,*

*occupation=>”superhero”,*

*age=>24,*

*speciality=>”superhuman strangth”),*

*array(name=>”mary”,*

*occupation=>”arch villiain”,*

*age=>63,*

*speciality=>”nanotechnology”)*

*);*

*foreach($characters as $val)*

*{*

*foreach($val as $key=>final\_val)*

*{*

*print “$key:final\_val)*

*}*

*print “<br>”;*

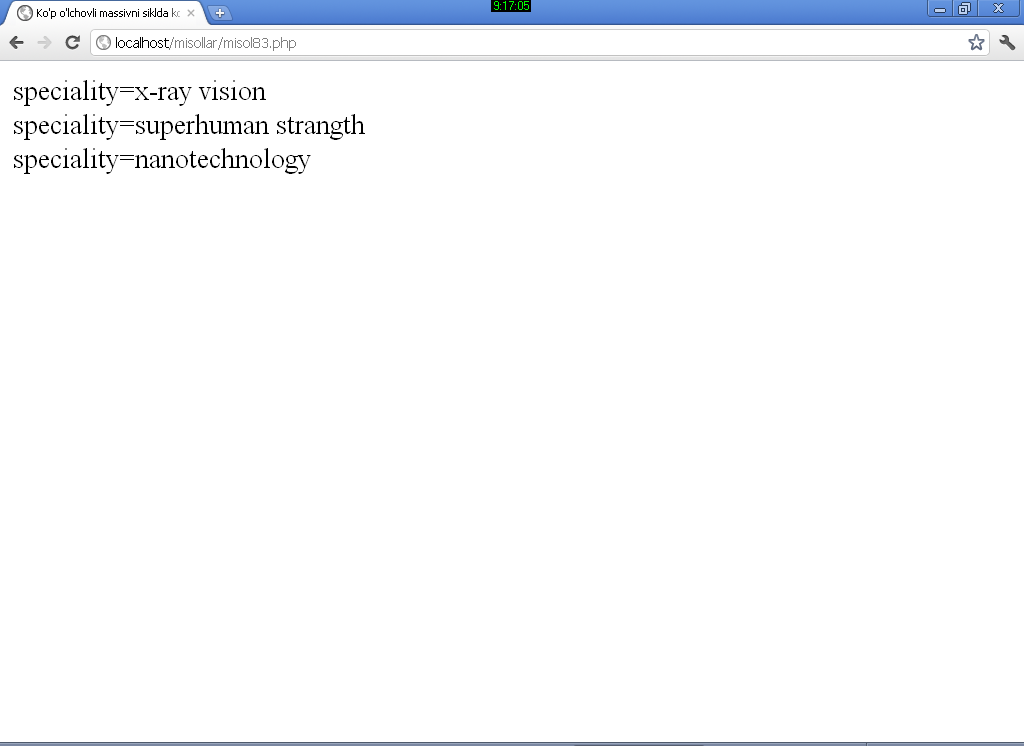
*}*

*?>*

*</body>*

*</html>*

Bu dasturdan 2 ta foreach stiklini tashkil qilinadi. Tashqi stikl nomeri bo’yicha saralangan $users massiv elementlarini ajratadi. Bunda har bir element qiymati $val o’zgaruvchisiga olinadi. Bu o’zgaruvchiga olingan har bir elementning o’zi ham massiv hisoblanadi va uni stiklda ko’rish mumkin. Ichki tashkil etilgan foreach stikli esa har bir element nomi va qiymatini $key va $final\_val o’zgaruvchilarga oladi.



*Rasm 16.9. Dasturdan olingan natija.*

Dastur biz hohlagandek ishlashi uchun $val o’zgaruvchisi doimo massiv bo’lishi lozim. Dasturini yanada yaxshiroq bo’lishi uchun argument massiv bo’lsa true aks holda false qiymat qaytaruvchi is\_array() funkstiyasidan foydalanish lozim.

Endi massiv qiymatlarini saqlash va uning elementiga murojaat qilishni bilamiz, lekin PHP tilida bundan ham yaxshiroq imkoniyatlar mavjud.

## array\_merge() funkstiyasi bilan massivlarni bog’lash

Array\_merge() funkstiyasi bir necha massiv argumentlarini qabul qiladi va ularni bog’langanligi haqidagi natijani qaytaradi. Quyidagi misolda 2 ta massiv tashkil etib, 2-massivni birinchi massivga bog’laymiz va bu bog’lanish natijasini stiklda ko’ramiz.

*<?php*

*$first=array ("a", "b", "c");*

*$second=array("1", "2", "3");*

*$third=array\_merge($first,$second);*

*foreach ($third as $val)*

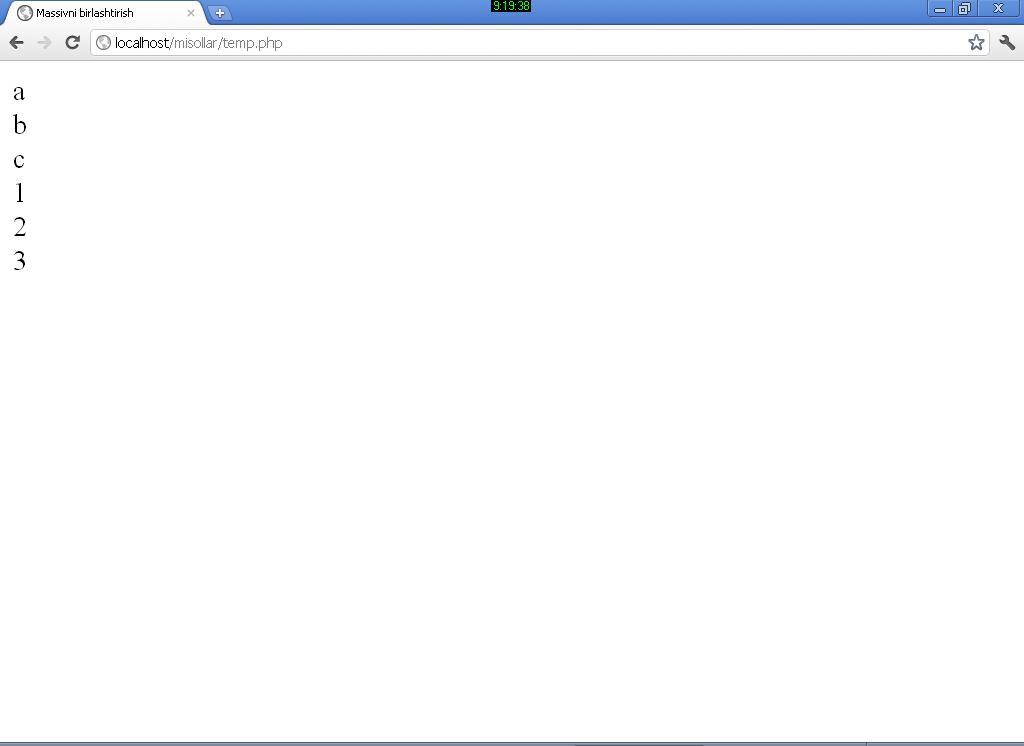
*{*

*print "$val<br>";*

*}*

*?>*

$third massivda $first va $second massivdagi barcha elementlarning nusxasi olinadi. Foreach stikl bu elementlarning hammasini <br> tegi bilan ajratib chop qilib beradi. Bunda berilgan massivdagi ma’lumotlar o’zgarib ketmaydi.



*Rasm 16.10. Dasturdan olingan natija.*

Lekin array\_merge() funkstiyasi PHP tilini 4-versiyasidan boshlab ishlatilishi mumkin.

## Array\_push() funkstiyasi yordamida massivga element qo’shish

Array\_push() funkstiyasi massiv argumenti va shu massivga bog’lanayotgan argumentlar uchun bir nechta parametrlarni oladi. Bu funkstiyaning boshqa funkstiyalardan farqi, birinchi argumentni massiv sifatida oladi. Array\_push() funkstiyasi natijaviy massivdagi elementlar sonini qaytaradi. Keltirilgan fikrlarni mustahkamlash maqsadida quyidagi misol yordamida massiv yaratamiz va unga bir necha element qo’shamiz.

*$first=array(“a”, “b” ,”c”);*

*$total= array\_push ($first,1,2,3);*

Bu erda array\_push() funkstiyasi natijaviy massivdagi elementlar sonini qaytaradi. Endi $first massividagi elementlarga array\_push() funkstiyasi yordamida berilgan 3 ta butun sonlar ham qo’shiladi. Bu barcha elementlarni foreach stikli yordamida brauzer oynasiga chop qilinadi.

Agar dasturga ahamiyat berilsa, $first o’zgaruvchisidan oldin “\” belgisi qo’yilgan. Bu belgini qo’yilmasa PHP qator emas, balki $first o’zgaruvchi qiymatini qaytarishi mumkin. Bu belgini ishlatishni maksimallashtirish deb ataladi.

## Array\_shift() funkstiyasi yordamida birinchi elementni o’chirish

Array\_shift() funkstiyasi berilgan massivning birinchi elementini o’chiradi va shu o’chirilgan elementni qaytaradi. Quyidagi misolda ushbu funkstiyani stiklda qo’llaniladi. Bunda massivda biror element qolganligini count() funkstiyasi yordamida tekshirib turamiz.

*<?php*

*$an\_array=array ("a" ,"b", "c");*

*while (count($an\_array))*

*{*

*$val=array\_shift($an\_array);*

*print "$val<br>";*

*print "$an\_array massivda ".Count ($an\_array)." element bor<br>";*

*}*

*?>*

Bu dasturning natijasi quyidagi rasmda keltirilgan. Bu array\_shift() funkstiyasi ham PHP tilining 4-versiyasida paydo bo’lgan.

## array\_slice() funkstiyasi yordamida massiv qismini ajratish

array\_slice() funkstiyasi yordamida massiv qismini ajratish mumkin. Buning uchun massiv nomi, ajratilayotgan qismning birinchi (boshlang’ich) pozistiyasi va shart bo’lmagan argumentning oxirgi pozistiya beriladi. Agar ikkinchi argument berilmasa, u holda ajratilayotgan qism massiv oxirigacha deb tushuniladi. Array\_slice() funkstiyasi mavjud massivni o’zgartirmaydi va ajratilgan qismdan iborat yangi massiv qaytaradi.

Quyidagi misolda massiv yaratamiz va undan 3 ta elementdan iborat yangi massiv ajratamiz.

*<?php*

*$first=array ("a", "b", "c", "d", "e", "f");*

*$second=array\_slice($first,3,4);*

*foreach ($second as $val)*

*{*

*print "$val <br>";*

*}*

*?>*

Natija sifatida esa, “s”, “d”, “e” dan iborat 3 ta element beriladi. Agar array\_slice() funkstiyaning ikkinchi argumentiga qiymat berilmasa, elementlar soni berilgan massivning oxirigacha deb tushuniladi. array\_slice() funkstiyasi ham PHP tilining 4-versiyasida paydo bo’lgan.

## Massivlarni saralash

Massivlarni saralash - bu massivlar bilan ishlashning eng yaxshi usuli hisoblanadi. Buning uchun PHP tilida bir necha funkstiyalar mavjud bo’lib, ular oddiy va strukturali massivlarni saralash imkonini beradi.

## sort() funkstiyasi yordamida oddiy massivni saralash.

*sort()* funkstiyasi 1 ta argument qabul qiladi va uni alfavit bo’yicha saralaydi. Agar uning elementlari sonlardan iborat bo’lsa, tartib bo’yicha saralaydi. Bu funkstiya berilgan massivni qayta ishlaydi va hech narsa qaytarmaydi. Masalan, quyidagi misol yordamida 1 ta simvoldan iborat qatorli massiv yaratamiz hamda uni saralaymiz va natijani brauzerga chop qilamiz.

*<?php*

*$an\_array=array("x", "a", "f", "c");*

*sort ($an\_array);*

*foreach ($an\_array as $var)*

*{*

*print "$var<br>";*

*}*

*?>*

*sort()* funkstiyasiga strukturali massiv berilmasligi lozim. Aks holda massiv to’g’ri saralansada, massivdagi hamma nomlar yo’qolib o’rniga sonlar yozilib qoladi.

Oddiy massivlarni teskari tartibda saralash ham mumkin. Buning uchun sort() funkstiyasi kabi foydalanuvchi rsort() funkstiyasi qo’llaniladi.

## asort() funkstiyasi yordamida strukturali massivni saralash

asort() funkstiyasi strukturali massiv argumentinioladi va sort() funkstiyasi kabi uni saralaydi lekin maydon nomini o’chirmaydi

*<?php*

*$first=array ("birinchi"=>5, "ikkinchi"=>2, "uchinchi"=>1);*

*asort($first);*

*foreach($first as $key=>$val)*

*{*

*print "$key=$val<br>";*

*}*

*?>*

Bunday massivni teskari tartibda saralash uchun huddi shunday tartibda ishlovchi *arsort()* funkstiyasidan foydalanish lozim.

## ksort() funkstiyasi yordamida strukturali massivni nomi bo’yicha saralash

*ksort()* funkstiyasi strukturali massiv argumentini oladi va uni maydon nomi bo’yicha saralaydi. U saralanayotgan massivni oladi lekin qiymat qaytarmaydi.

Masalan:

*<?php*

*$first=array ("x"=>5, "a"=>2, "f"=>1) ;*

*ksort($first);*

*foreach ($first as $key=>$val)*

*{*

*print "$key=$val <br>";*

*}*

*?>*

Massivni teskari tartibda saralash uchun shunday tartibda ishlovchi *krsort()* funkstiyasidan foydalanish lozim.

# 

# Mаssivlаr

# *Mаssivlаr initsiаlizаtsiyasi*

PHP dа mаssivlаrni initsiаlizаtsiya qilishning 2 usuli mаvjud. Birinchisi mаssiv elementlаrigа qiymаt berishdаn ibоrаt:

**<?**

**$car**[] **=** "passenger car"**;**

**$car**[] **=** "land-rover"**;**

**echo($car**[1]**);** // chiqаrаdi "land-rover"

**?>**

Mаssiv indeksini оchiq ko‘rsаtish mumkin:

**<?**

**$car**[0] **=** "passenger car"**;**

**$car**[1] **=** "land-rover"**;**

**echo($car**[1]**);** // chiqаrаdi "land-rover"

**?>**

Аgаr mаssiv elementlаrini e’lоn qilishdа оshkоrа indeksаtsiyali vа indeksаtsiyasiz o‘zgаruvchilаr аrаlаshib kelsа indeksi berilmаgаn elementgа ishlаtilgаn indekslаr ichidа eng kаttаsidаn keyin keluvchi ruхsаt berilgаn indeksni berаdi. Mаsаlаn аgаr biz yarаtgаn mаssiv elementldаr indekslаri 10, 20 vа 30 bo‘lsа vа indeks ko‘rsаtmаsdаn Yangi element yarаtsаk, uning indeksi аvtоmаtik rаvishdа 31 bo‘lаdi:

**<?**

**$car**[10] **=** "passenger car"**;**

**$car**[20] **=** "land-rover"**;**

**$car**[30] **=** "station-wagon"**;**

**$car**[] **=** "victoria"**;**

**echo($car**[31]**);**

**?>**

Аlternаtiv usul **array**() kоnstruktsiyasidаn fоydаlаnishdаn ibоrаt:

**<?**

**$car** **=** array**(**"passenger car","land-rover"**);**

**echo($car**[1]**);** // chiqаrаdi "land-rover"

**?>**

Indekslаrni оshkоrа ko‘rsаtish uchun **=>** оperаtоr qo‘llаnаdi:

**<?**

**$car** **=** array**(**"passenger car", 5 => "land-rover",

"station-wagon","victoria"**);**

**echo($car**[0]**); echo(**"<**br**>"**);** // chiqаrаdi "passenger car"

**echo($car**[5]**); echo(**"<**br**>"**);** // chiqаrаdi "land-rover"

**echo($car**[6]**); echo(**"<**br**>"**);** // chiqаrаdi "station-wagon"

**echo($car**[7]**);** **// chiqаrаdi "victoria"**

**?>**

Mаssiv indekslаri sаtrlаr ham bo‘lishi mumkin:

**<?**

**$car** **=** array**(**"pc" => "passenger car", "lr" => "land-rover"**);**

**echo($car**["lr"]**); echo(**"<**br**>"**);** // chiqаrаdi "land-rover"

**echo($car**["pc"]**);** // chiqаrаdi "passenger car"

**?>**

# *Mаssivlаrni ko‘rib chiqish uchun foreach sikli*

PHP4 dа mаssiv elementlаrini ko‘rib chiqish uchun **foreach** оperаtоridаn fоydаlаnish mumkin. Bu оperаtоr sintаksisi:

foreach **(**array as [**$key** =>] **$value)**

**{**

statements**;**

**}**

Busikl mа’nоsi sоddа: har bir element ko‘rilgаndа uning indeksi **$key** o‘zgаruvchigа, qiymаti bo‘lsа **$value** o‘zgаruvchigа jоylаshtirilаdi. Bu ikki o‘zgаruvchilаrning nоmlаri iхtiyoriydir.

*Misоl:*

**<?**

**$car** **=** array**(**"passenger car", "land-rover",

"station-wagon","victoria"**);**

foreach**($car** as **$index** => **$val)**

**{**

**echo(**"**$index** -> **$val** <**br**>"**);**

**}**

**?>**

Sintаksisdаn ko‘rinib turibdiki, **$key** o‘zgаruvchidаn fоydаlаnish shаrt emаs, shuning uchun tаshlаb yubоrilishi mumkin:

**<?**

**echo(**

"available cars: <**br**> <**ul**>"

**);**

**$car** **=** array**(**"passenger car", "land-rover",

"station-wagon","victoria"**);**

foreach**($car** as **$val)**

**{**

**echo(**"<**li**>**$val**<**/li**>\n"**);**

**}**

**echo(**"<**/ul**>"**);**

**?>**

# *Ko‘p o‘lchоvli mаssivlаr*

Ko‘p o‘lchоvli mаssivlаrni ko‘rib chiqish uchun ichki jоylаshgаn **array**() kоnstruktsiyasidаn fоydаlаnilаdi. Ko‘p o‘lchovli massivlarni o‘qib chiqish joylangan sikllar yordamida amalgam oshiriladi. Quyidagi scriptda ko‘p o‘lchovli massiv yaratish va ko‘rib chiqish ko‘rsatilgan.

Misоl:

**<?**

**$ship** **=** array**(**

"Passenger ship" => array("Yacht","Liner","Ferry"),

"War ship" => array("Battle-wagon","Submarine","Cruiser"),

"Freight ship" => array("Tank vessel","Dry-cargo ship","Container

cargo ship")

**);**

foreach**($ship** as **$key** => **$type)**

**{**

**echo(**

"<**h2**>**$key**<**/h2**>\n"."<**ul**>\n"**);**

foreach**($type** as **$ship)**

**{**

**echo(**"\t<**li**>**$ship**<**/li**>\n"**);**

**}**

**}**

**echo(**"<**/ul**>\n"**);**

**?>**

*Bu scriptbаjаrilish nаtijаsi:*

## Passenger ship

* Yacht
* Liner
* Ferry

## War ship

* + Battle-wagon
  + Submarine
  + Cruiser

## Freight ship

* + - Tank vessel
    - Dry-cargo ship
    - Container cargo ship

Endi PHP dа mаvjud mаssivlаr Bilаn ishlаsh funktsiyalаrini ko‘rаmiz. Biz mаssivlаrni tаrtiblаsh funktsiyalаrdаn bоshlаymiz. Lekin аvvаl misоllаrimizdа ko‘p fоydаlаnаdigаn uchtа funktsiyani ko‘rib chiqаmiz.

# Funktsiya count()

*Sintаksis:*

int **count(**mixed **var)**

Bu funktsiya аrgument sifаtidа mаssivni qаbul qilib, undаgi elementlаr sоnini qаytаrаdi.

# Funktsiya in\_array()

*Sintаksis:*

boolean **in\_array(**mixed **needle**, array **haystack** [, bool **strict**]**)**

Bu funktsiya **haystack** mаssivdа **needle** qiymаtni qidirаdi vа аgаr u mаvjud bo‘lsа **true** qаytаrаdi, аks хоldа **false** qаytаrаdi.

# Funktsiya reset()

*Sintаksis:*

mixed **reset(**array **array)**

Funktsiya **reset**()mаssiv ko‘rsаtkichini birinchi elementgа o‘rnаtаdi vа mаssiv birinchi elementi qiymаtini qаytаrаdi.

*Kichkinа izох. PHP dа har bir mаssiv jоrielemengа ko‘rsаtkichnа egа. Quyidаgi* ***foreach*** *kаbi kоnstruktsiyalаr Bilаn ishlаshdа ko‘rsаtkich haqida o‘ylаsh kerаk emаs, chunki foreach uni mаssiv bоshigа o‘rnаtаdi.Lekin bоshqа mаssivlаr funktsiyalаri mаsаlаn* ***prev****(),* ***next****(), mаssiv ko‘rsаtkichlаrini surаdi, bu esа* ***array\_walk****(), kаbi qаytа ishlаshni ko‘rsаtkichni turgаn jоyidаn bоshlоvchi funktsiyalаr uchun katta аhamiyatgа egа.*

Endi tаrtiblаsh Bilаn shug‘ullаnаmiz.

# *Mаssivlаrni tаrtiblаsh funktsiyalаri*

# sort()

Mаssivni o‘sish bo‘yichа tаrtiblаsh funktsiyasi.

*Sintаksis:*

void **sort(**array **array** [, int **sort\_flags**]**)**

Funktsiya **array** mаssivini o‘sish bo‘yichа tаrtiblаydi. Mаjburiy bo‘lmаgаn element **sort\_flags** elementlаr qаndаy tаrtiblаnishi kerаkligini ko‘rsаtаdi(tаrtiblаsh bаyrоqlаrini belgilаydi). Аrgumentning mumkin bo‘lgаn qiymаtlаri quyidаgilаr:

* SORT\_REGULAR – elementlаr nоrmаl sоldishtirаdi nоrmаlnоe srаvnenie elementоv (elementlаrni "bоrichа" sоlishtirаdi)
* SORT\_NUMERIC – elementlаrni sоnlаr sifаtidа sоlishtirаdi
* SORT\_STRING - elementlаrni sаtrlаr sifаtidа аlmаshtirаdi

Umumаn оlgаndа bu funktsiya ro‘yхаtlаrni tаrtiblаsh uchun mo‘ljаllаngаn. Ro‘yхаt degаndа kаlitlаri nuldаn bоshlаngаn vа bo‘shliklаrgа egа bo‘lmаgаn mаssiv tushunilаdi. Funktsiya **sort**() iхtiyoriy mаssivni ro‘yхаt deb qаrаydi.

Misоl:

**<?**

**$arr** **=** array**(**"2", "1", "4", "3","5"**);**

sort**($arr);**

for**($i**=0**;** **$i** < count**($arr);** **$i**++**)**

**{**

**echo (**"**$i**:**$arr**[**$i**] "**);**

**}**

// chiqаrаdi "0:1 1:2 2:3 3:4 4:5"

**?>**

*Nаtijа:*

0:1 1:2 2:3 3:4 4:5

Аgаr siz sаtrlаrni tаrtilаyotgаn bo‘lsаngiz, misоl uchun mаssiv quyidаgi ko‘rinishgа egа bo‘lsа

array**(**"one", "two", "abs", "three", "uic", "for","five"**);**

bu аjоyib funktsiya quyidаgi nаtijаni qаytаrаdi:

*Nаtijа:*

0:abs 1:five 2:for 3:one 4:three 5:two 6:uic

Ya’ni sаtrlаrni u аlfа-betа tаrtibdа, sоddаrоq аytgаndа birinchi harflаri аlfаvitdа kelishi bo‘yichа tаrtiblаydi.

# rsort()

Mаssivlаrni kаmаyish bo‘yichа tаrtiblаsh.

*Sintаksis:*

void **rsort(**array **arr** [, int **sort\_flags**]**)**

Shungа o‘хshаsh **sort**() funktsiyasi fаqаt kаmаyish bo‘yichа tаrtiblаydi. Оldingi **sort**() funktsiyasi uchun ko‘rilgаn scriptni оlаmiz, fаqаt sort(**$arr**) o‘rnigа rsort(**$arr**) qo‘yamiz.

*Nаtijа:*

0:5 1:4 2:3 3:2 4:1

# asort()

Аssоtsiаtiv mаssivni o‘sish bo‘yichа tаrtiblаsh.

*Sintаksis:*

void **asort(**array **arr** [, int **sort\_flags**]**)**

Funktsiya **asort**() berilgаn **arr** mаssivni shundаy tаrtiblаydiki uning qiymаtlаri аlfаvit tаrtibidа (аgаr sаtr bo‘lsа) yoki o‘sish tаrtibdа (sоnlаr uchunl) tаrtibdа jоylаshаdi. Bu funktsiyaning **sort**() funktsiyasidаn muхim fаrqi shundаki **asort**() funktsiyasi qo‘llаnilgаndа kаlitlаr vа ulаrgа mоs qiymаtlаr оrаsidа bоg‘liklik sаqlаnаdi, **sort**() funktsiyasidа bo‘lsа bu bоg‘liqlik uzilаdi.

*Misоl:*

**<?**

**$arr** **=** array**(**"a" =>"one","b" => "two","c" => "three","d" => "four"**);**

asort**($arr);**

foreach**($arr** **as** **$key** **=>** **$val)**

**{**

**echo (**" **$key** => **$val** "**);**

**}**

**?>**

*Nаtijа:*

d => four a => one c => three b => two

Ko‘rinib turibdiki "kаlit-qiymаt" bоg‘lаnishlаri sаqlаnib qоlgаn.

Ko‘zdа tutilgаn bo‘yichа **asort**() funktsiyasi mаssivni аlfаvit bo‘yichа tаrtiblаydi. Tаrtiblаshbаyrоqlаri **sort\_flags** qiymаtlаri **sort**() funktsiyasi tа’rifidа keltirilgаn.

# arsort()

Аssоtsiаtiv mаssivlаrni kаmаyish bo‘yichа tаrtiblаsh.

*Sintаksis:*

void **arsort(**array **arr** [, int **sort\_flags**]**)**

Bu funktsiya **asort**() funktsiyasigа o‘хshаsh, fаqаt u mаsivni o‘sish bo‘yichа emаs kаmаyish bo‘yichа tаrtiblаydi.

# ksort()

Mаssivlаrni kаlit o‘sishi bo‘yichа tаrtiblаsh.

*Sintаksis:*

int **ksort(**array **arr** [, int **sort\_flags**]**)**

Bu funktsiyadа tаrtiblаsh qiymаtlаr bo‘yichа emаs, bаlkim kаlitlаr bo‘yichа o‘sish tаrtibidа аmаlgа оshirilаdi.

**<?**

**$arr** **=** array**(**"a" =>"one","b" => "two","c" => "three","d" => "four"**);**

ksort**($arr);**

foreach**($arr** **as** **$key** **=>** **$val)**

**{**

**echo (**" **$key** **=>** **$val** "**);**

**}**

**?>**

*Nаtijа:*

a => one b => two c => three d => four

# krsort()

Indekslаr kаmаyishi bo‘yichа mаssivlаrni tаrtiblаsh.

*Sintаksis:*

int **krsort(**array **arr** [, int **sort\_flags**]**)**

Хuddi **ksort**() funktsiyagа o‘хshаsh, fаqаt mаssivni kаlitlаr bo‘yichа teskаri tаrtibdа (kаmаyish bo‘yichа) tаrtiblаydi.

# array\_reverse()

Mаssiv elementlаrini teskаri jоylаshtirish.

*Sintаksis:*

array **array\_reverse(**array **arr** [, bool **preserve\_keys**]**)**

Funktsiya **array\_reverse**()elementlаri pаrаmetrdа berilgаn **arr** mаssivi elementlаrigа nisbаtаn teskаri jоylаshtirilgаn mаssivni qаytаrаdi. Kаlitlаr vа qiymаtlаr оrаsidаgi bоg‘lаnish sаqlаnib qоlаdi. Аgаr mаjburiy bo‘lmаgаn pаrаmetr **preserve\_keys** gа **true** berilsа, kаlitlаr ham teskаri tаrtibdа jоylаshаdi.

Misоl:

**<?**

**$arr** **=** array **(**"php", 4.0, array **(**"green", "red"**));**

**$result** **=** array\_reverse **($arr);**

**echo** "Mаssiv: <**br**>"**;**

foreach**($result** **as** **$key** **=>** **$val)**

**{**

**echo (**"**$key** **=>** **$val** <**br**>"**);**

**}**

**echo(**"<**br**>"**);**

**echo** "Tаrtiblаngаn mаssiv: <**br**>"**;**

**$result\_keed** **=** array\_reverse **($arr**, **false);**

**foreach($result\_keed as $key => $val)**

**{**

**echo (**"**$key** **=>** **$val**<**br**> "**);**

**}**

**?>**

Birinchi хоldа:

*Nаtijа:*

Mаssiv:   
0 =>Array   
1 =>4   
2 =>php   
  
Tаrtiblаngаn mаssiv:   
0 =>Array   
1 =>4   
2 =>php

Аgаr ikkinchi pаrаmetrgа **true** qiymаt berilsа:

*Nаtijа:*

Mаssiv:   
0 =>Array   
1 =>4   
2 =>php   
  
Tаrtiblаngаn mаssiv:   
2 =>Array   
1 =>4   
0 =>php

# shuffle()

Mаssiv elementlаrini tаsоdifiy jоylаshtirish.

*Sintаksis:*

void **shuffle(**array **arr)**

**shuffle**() funktsiyasi **arr** mаssivi elementlаrini tаsоdifiy аrаlаshtirаdi.

# natsort()

Tаbiiy tаrtiblаshni bаjаrаdi.

*Sintаksis:*

void **natsort(**array **arr)**

Bundаy tаrtiblаshni sаtrlаrni tаrtiblаshdа uchrаtgаn edik. Tаbiiy tаrtiblаsh deb elementlаr tushunаrli tаrtibdа jоylаshgа аytilаdi.

*Misоl:*

**<?**

**$array1** **=** **$array2** **=** array**(**"pict10.gif", "pict2.gif", "pict20.gif", "pict1.gif"**);**

**echo (**"оddiy tаrtiblаsh:"**); echo (**"<**br**>"**);**

sort**($array1);**

print\_r**($array1);**

**echo (**"<**br**>"**); echo (**"tаbiiy tаrtiblаsh:"**); echo (**"<**br**>"**);**

natsort**($array2);**

print\_r**($array2);**

**?>**

*Nаtijа:*

оddiy tаrtiblаsh:  
Array ( [0] => pict1.gif [1] => pict10.gif [2] => pict2.gif [3] => pict20.gif )   
estestvennаya sоrtirоvkа:  
Array ( [3] => pict1.gif [1] => pict2.gif [0] => pict10.gif [2] => pict20.gif )

# *Mаssivs kursоri (ko‘rsаtkichi) Bilаn ishlаsh*

Yuqоridа **reset**() funktsiyasi Bilаn tаnishgаn edik. Bu funktsiya mаssiv kursоrini bo‘shаtаdi, Ya’ni mаssiv ichki kurssоrini mаssiv bоshigа keltirаdi vа birinchi element qiymаtini qаytаrаdi.

Funktsiya **end**() ko‘rilgаn **reset**() funktsiyagа teskаri vаzifа bаjаrаdi –kursоrni mаssivа охirigа keltirаdi. Funktsiya sintаksis **reset**() funktsiyasi sintаksisigа o‘хshаsh:

# end()

*Sintаksis:*

mixed **end(**array array arr**)**

# next()

Funktsiya next() mаssiv kursоrini bitа pоzitsiya оldingа surаdi.

*Sintаksis:*

mixed **next(**array array arr**)**

Ya’ni bu funktsiya mаssiv kursоrini keyingi elementgа o‘rnаtаdi vа siljitishdаn оldin kursоr jоylаshgаn element qiymаtini qаytаrаdi. Аgаr mаssivdа qоlmаgаn bo‘lsа **false** qаytаrаdi. Bu funktsiyadаn fоydаlаnilgаndа bir nаrsаni unutmаslik kerаk: kursоrgа bo‘sh element uchrаgаn хоldа ham **false** qаytаrilаdi. Shuning uchun bo‘sh elementlаri yuo‘lgаn mаssivlаr bilаn ishlаmоqchi bo‘lsаngiz **each**() funktsiyasidаn fоydаlаngаn yaхshirоqdir.

# prev()

**prev**() funktsiyasi kursоrni bittа pоzitsiyagа оrqаgа surаdi. Funktsiya Sintаksisi vа ishi **next**() funktsiyasi Bilаn bir хil.

*Sintаksis:*

mixed **prev(**array array arr**)**

# current()

Mаssivning jоriy elementini, kursоr хоlаtini o‘zgаrtirmаsdаn аniqlаsh uchun **current**() funktsiyasi qo‘llаnаdi.

*Sintаksis:*

mixed **current(**array array arr**)**

Funktsiya **current**() mаssiv kursоri jоylаshgаn elementni fаytаrаdi vа shu bilаn birgа kursоrni siljitmаydi. Аgаr kursоr mаssiv tаshqаrisidа bo‘lsа yoki mаssiv bo‘sh elementlаrdаn ibоrаt bo‘lsа funktsiya **false** qаytаrаdi.

**pos**()funktsiyasi **current**()funktsiyasining to‘lа sinоnimidir.

# key()

Funktsiya **key**() mаssiv jоriy elementi indeksini qаytаrаdi.

*Sintаksis:*

mixed **key(**array array arr**)**

Endi оldin аytilgаn **each**() funktsiyasini ko‘rаmiz.

# each()

*Sintаksis:*

array **each(**array array arr**)**

Funktsiya **each**() mаssiv jоriy elementi "indeks – qiymаt juftligini qаytаrаdi vа mаssiv kursоrini keyingi elementgа surаdi. Funktsiya mаssiv qаytаrib, to‘rttа elementgа egаdir:

1. [1] => "znаchenie"
2. [value] => "znаchenie"
3. [0] => indeks
4. [key] => indeks

Аgаr kursоr mаssiv охirigа etgаn bo‘lsа, funktsiya **false** qаytаrаdi.

Funktsiya ishini ko‘rib chiqаmiz:

*Misоl:*

**<?**

**$name** **=** array**(**"maks", "igor", "sergey"**);**

**$each\_name** **=** each**($name);**

print\_r**($each\_name);**

**echo(**"<**br**>"**);**

**$each\_name** **=** each**($name);**

print\_r**($each\_name);**

**echo(**"<**br**>"**);**

**$each\_name** **=** each**($name);**

print\_r**($each\_name);**

**?>**

Biz хоsil qilgаn:

*Nаtijа:*

Array ( [1] => maks [value] => maks [0] => 0 [key] => 0 )   
Array ( [1] => igor [value] => igor [0] => 1 [key] => 1 )   
Array ( [1] => sergey [value] => sergey [0] => 2 [key] => 2 )

Funktsiya **each**() vа funktsiya **list**() mаssiv elementlаrini tekshirib chiqish uchun birgаlikdа qo‘llаnishi mumkin. Mаsаlаn quyidаgichа:

*Misоl:*

**<?**

**$name** **=** array**(**"maks", "igor", "sergey"**);**

reset**($name);**

while**(**list**($key**, **$val) =** each**($name))**

**{**

**echo (**"**$key** **=** **$val**<**br**>"**);**

**}**

**?>**

Chiqishdаgi:

*Nаtijа:*

0 = maks  
1 = igor  
2 = sergey

# Funktsiya array\_walk()

# array\_walk()

Muхim funktsiya bo‘lib, fоydаlаnuvchi funktsiyasini mаssiv harbir elementigа qo‘llаshgа imkоn berаdi.

*Sintаksis:*

bool **array\_walk(**array **arr**, callback **func** [, mixed **userdata**]**)**

Bu funktsiya sintаksisidаn ko‘rinib turibdiki u **func** fоydаlаnuvchi funktsiyasini **arr** mаssivning har bir elementigа qo‘llаydi. Fоydаlаnuvchi funktsiyasigа ikki yoki uch аrgument uzаtilаdi: jоriy element qiymаti, indeksi vа аrgument **userdata**. Охirgi аrgument mаjburiy emаs. Аgаr **func** uchtаdаn оrtiq аrgument tаlаb qilsа, vа har gаl chаqirgаndа оgохlаntirish chiqаrilsа, bu оgохlаntirish chiqаrilmаsligi uchun **array\_walk**() funktsiyasi оldidаn "**@**" belgisini qo‘yish kerаk. Yanа shuni аytish kerаkki **func** funktsiyasi **arr** mаssiv qiymаtlаri vа indekslаrini shundаy оlаdiki, ulаrni o‘zgаrtirа оlmаydi. Аgаr shundаy zаrurаt tug‘ilsа **arr** аrgumentni ilоvа bo‘yichа uzаtish lоzim. Bu хоldа hamma o‘zgаrtishlаr mаssivdа аks etаdi.

Funktsiya qаndаy qo‘llаnishini ko‘rib chiqаmiz. Bizgа mаssiv hamma elementlаrini chiqаrish kerаk bo‘lsin. Buning uchun оldin ulаrni chiqаrаdigаn funktsiyani yozib, **array\_walk**() funktsiyasi yordаmidа uni chаqirаmiz:

*Misоl:*

**<?**

**$name** **=** array **(**"m"=>"maks", "i"=>"igor", "s"=>"sergey"**);**

function print\_array **($item**, **$key)**

**{**

**echo** "**$key=>$item**<**br**>\n"**;**

**}**

array\_walk **($name**, 'print\_array'**);**

**?>**

Nima chiqdi:

*Nаtijа:*

m=>maks  
i=>igor  
s=>sergey

Muхim qo‘shimchа. Yuqоridа keltirilgаn kоddа nоаniqlik mаvjuddir. Ya’ni biz mаssiv kursоrini mаssiv bоshigа o‘rnаtmаdik vа shuning uchun **array\_walk**() funktsiyasini chаqirishdаn оldin shu mаqsаddа **reset**() funktsiyasini chаqirish lоzim, chunki **array\_walk**() mаssiv kursоri turgаn elementdаn ishni bоshlаydi.

Endi mаssivni оlib, uning har bir elementini birgа оshirаmiz.

*Misоl:*

**<?**

**$number** **=** array **(**"1"=>"15", "2"=>"20", "3"=>"25"**);**

function printarray **($item**, **$key)**

**{**

**echo** "**$key**=>**$item**<**br**>\n"**;**

**}**

function add\_array **(**&**$item**, **$key)**

// $item илоа буйича узатилади

// уни узгарtириш керак

{

**$item** **=** **$item** + 1**;**

**}**

**echo(**"Before:<**br**>"**);**

array\_walk **($number**, 'printarray'**);**

**echo(**"After:<**br**>"**);**

array\_walk **($number**, 'add\_array'**);**

array\_walk **($number**, 'printarray'**);**

**?>**

*Nаtijа:*

Before:  
1=>15  
2=>20  
3=>25  
After:  
1=>16  
2=>21  
3=>26

|  |  |
| --- | --- |
|  | ***MUHOKAMA UCHUN SAVOLLAR VA TOPSHIRIQLAR!*** |

1. *Butun qiymatlar haqiqiy qiymat kabi ekranga chop qilish uchun qaysi simvoldan foydalaniladi?*
2. *Qator uzunligini aniqlash funkstiyasini keltiring.*
3. *Satrdan satr pozistiyasini aniqlash uchun qaysi funkstiyadan foydalaniladi?*
4. *Satrdan satrni ajratish (bo’lib olish) uchun qaysi funkstiyadan foydalaniladi?*
5. *Qator boshidan qaysi funkstiya yordamda bo’shliqni olib tashlanadi?*
6. *Satrdagi hamma harflarni katta harflarga o’tkazish qaysi funkstiya orqali bajariladi?*
7. *Satrdagi elementlarni massivga olish funkstiyasini keltiring?*
8. *Massiv yaratish uchun qanday funkstiyalardan foydalanish mumkin?*
9. *Quyidagi massivda oxirgi elementning indeksi nechaga teng? $users=array(“H”, “B”, “S”);*
10. *Massivdagi elementlar sonini qanday aniqlash mumkin?*
11. *Massivning barcha elementlarini qanday ko’rish mumkin?*
12. *Ikkita massivni qo’shish uchun qanday funkstiyadan foydalaniladi?*
13. *Strukturali massivni nomi bo’yicha qanday saralash mumkin*

## Qaytarish uchun mashqlar

1. *Forma yarating va shu formada foydalanuvchi ismi, familiyasi va pochtasini kiriting. Kiritilgan nomlarni hammasini bosh harflarini katta harfga aylantiring. Natijani brauzer oynasiga chop qiling.*
2. *Butun va haqiqiy sonlardan iborat qiymatlarga ega bo’lgan massiv yarating. Bu massiv elementlarini hammasini haqiqiy sonlar ko’rinishida verguldan keyin 2-tagacha aniqlikda son qoldirib ekranga chop qiling.*