# **17-ma’ruza. Funkstiya. Funkstiyani yaratish. Funkstiyaga dinamik murojaat qilish.**

Reja:

## Funkstiyalarga murojaat

## Funkstiya nima?

## Qiymat qaytaruvchi funkstiyalar yaratish

## Funkstiyalarga dinamik murojaat qilish

|  |  |
| --- | --- |
| **importance_of_education_1** | **O`quv modullari**  *PHPda funksiya, qiymat qaytaruvhci funksiya, funksiyaga dinamik murojaat* |

Funkstiya – bu yaxshi ishlangan dasturning eng asosiy komponenti hisoblanadi. Ular yaratilayotgan dasturning tushunarli juda ham yaxshi bo’lishiga va foydalanishga qulay bo’lishiga katta yordam beradi. Ixtiyoriy yaratilayotgan dastur uning yordamisiz biror katta natijaga erishgan emas. Ya’ni, funkstiyalar dasturlash bilan shug’ullanishni boshlaganlar va davom ettirayotgan uchun juda ham zarur va qulay hisoblanadi. Bu ma’ruza davomida funkstiyalar va ularning yordamida ko’p ishlatiluvchi bir xil tipdagi o’zgaruvchilar ustida ko’p martalik hisoblashlarni qanday qilib ixchamlashtirish mumkinligini o’rganildi. Bu ma’ruzada asosan quyidagilarni o’rganiladi:

* Funkstiyalarni qanday yaratish va ularga murojaat qilish;
* Funkstiya parametrlariga qiymatlarni qanday uzatish va qaytarilgan qiymatlarni qabul qilib olish;
* Matnli o’zgaruvchida uning nomini saqlagan holda qanday qilib funkstiyaga dinamik murojaat qilish;
* Global o’zgaruvchilarga funkstiyadan turib qanday murojaat qilish;
* «Eslash» funkstiyalarini qurish;
* Ko’rsatma bo’yicha funkstiya argumentini uzatish;

## Funkstiya nima?

Funkstiyani shunday qurol sifatida tasavvur qilish mumkinki, unga boshlang’ich ma’lumotlarni ya’ni xom ashyolarni berib tayyor mahsulot olish mumkin. Funkstiya boshlang’ich bir yoki bir necha ma’lumotlar, qiymatlar oladi hamda o’zida ularni qo’yilgan maqsadiga qarab qiymatlarni natija sifatida qaytaradi. Shuning uchun ham funkstiya maqsadga bog’liq ravishda ixtiyoriy vazifani bajaruvchi instrument hisoblanadi.

Masalan, oddiy misol sifatida uyda pechene pishirishni olib qaraylik. Agar bir-ikki marta pishirsak uni qo’lda tayyorlash mumkin. Lekin, agar bozor uchun har kuni tayyorlanadigan bo’lsa, u holda unga mos ravishda avtomat qidirib qolinadi. Ushbu misol asosida shunday savol tug’iladi. «Bu funkstiya dasturchi bajarmoqchi bo’lgan ishdagi takrorlanishlar uchun qanday ahamiyatga ega?». Bu esa jarayonni qonuniy va ko’p marta takrorlanishi bilan bog’liq.

U holda funkstiyani dasturida murojaat qilish mumkin bo’lgan alohida nomdagi dastur sifatida tasavvur qilish mumkin. Bu dasturiy funkstiya faqat murojaat qilingandagina ishlaydi. Funkstiya qaytargan qiymatlarni natija sifatida chop qilinishi mumkin yoki dastur davomida foydalanish mumkin.

***Yangi termin***. Funkstiya – bu instrukstiya va operatorlardan iborat qism dastur bo’lib, u asosiy dasturlarda zarur vaqtlarda foydalaniladi. Funkstiyalar tayyor qurilgan yoki dasturchi tomonidan qurilgan bo’lishi mumkin. Ular ba’zi bir boshlang’ich ma’lumotlarni o’zida foydalanish uchun so’rashi va natijaviy qiymatni qaytarishi mumkin.

## Funkstiyalarga murojaat

Funkstiya ikki xil bo’lishi mumkin. Bular tilning kompilyatorida mavjud funkstiyalar hamda dasturchi tomonidan yaratiladigan funkstiyalar. Birinchi ko’rinishdagi funkstiya xaqidagi deyarli ma’lumotga egamiz. Chunki, yuqorida keltirib o’tgan print (“Hello, Web”); funkiya kompilyatorga tegishli. Lekin print() funkstiyasi oddiy funkstiya hisoblanmaydi, chunki bu funkstiya o’zining argumenti uchun qavslarni qo’yish shartini qattiq talab qilmaydi. U holda print (“Hello,Web”) va print “Hello,Web” funkstiyalar ekvivalent konstrukstiyalar hisoblanadi. Lekin faqat shu funkstiyagina bunday ko’rinishga ega bo’la oladi. Qolgan barcha funkstiyalar qavslar qo’yilishini hatto argument talab qilmaganda ham talab qiladi.

Funkstiya albatta o’zining nomiga ega bo’lishi lozim. Yuqorida keltirib o’tilgan funkstiya nomi esa print va uning argumenti qavs ichida beriladi. Funkstiyada bir yoki bir necha argumentlar bo’lishi mumkin. Agar ko’p bo’lsa ular bir-biridan vergul (,) bilan ajratib yoziladi. Ularni umumiy holda quyidagicha keltirish mumkin:

Some-function ($argument-1, $argument-2); ixtiyoriy funkstiya albatta qiymat qaytaradi. Bunga sabab esa, funkstiya bajargan ishning muvaffaqiyatli yakunini aniqlash bo’ladi.

Masalan, abs() funkstiyasi sonning absolyut qiymatini qaytaradi. Buning uchun u manfiy son qabul qilib, musbat qiymat qaytaradi. Bu funkstiyaga esa, quyidagi dasturni misol sifatida keltirish mumkin:

*Misol-17.1. abs() funkstiyasini chaqirish*

<html>

<head>

<title> abs() funksiyasini chaqirish </title>

</head>

<body>

<?php

$num=-321;

$newnum=abs($num);

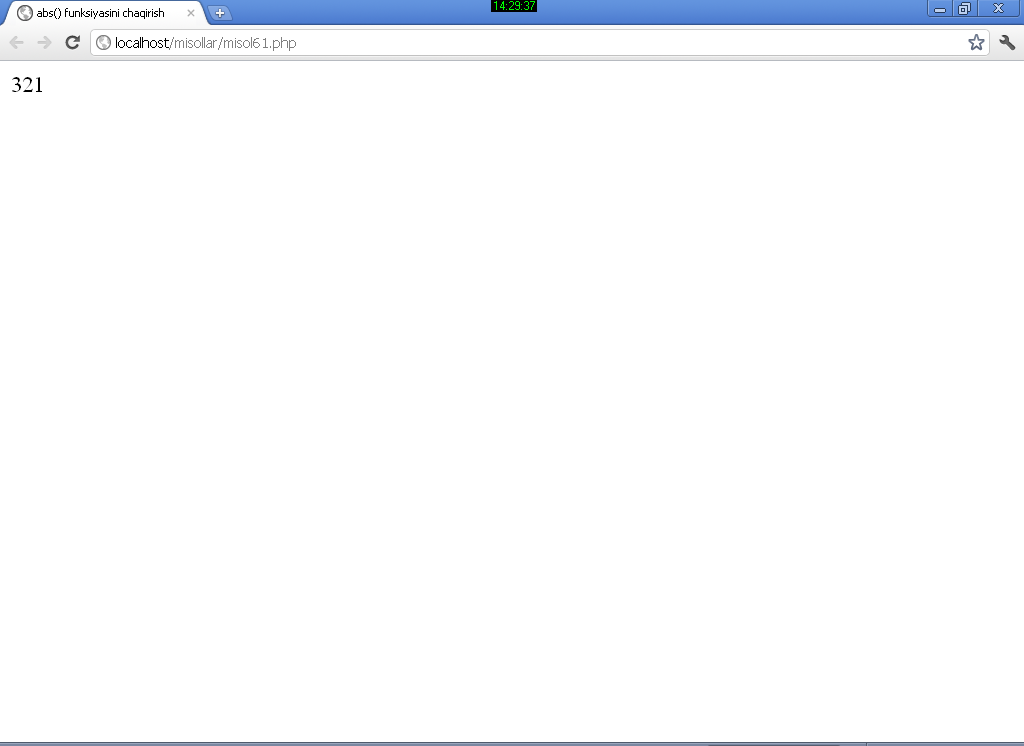
print $newnum; // natija «321»

?>

</body>

</html>

Bu misolda $num num o’zgaruvchisiga – 321 qiymatni o’zlashtiriladi. Keyin esa, abs() funkstiya argumentiga uzatib, shu funkstiya olgan natija $newnum ni o’zgaruvchiga uzatiladi. Bu o’zgaruvchidagi natijani print() orqali chop qilib olinadi. Aslida esa, dasturni yanada ixchamroq bo’lishi uchun $newnum o’zgaruvchisidan foydalanmay print(abs(-321)) berish ham mumkin.



*17.1-rasm.*

***Yangi termin***. Funkstiya argumenti - bu funkstiyaga murojaat qilingandagi beriladigan qiymatlar bo’lib, ular qavs ichida beriladi. Funkstiyani yaratishda argument nomi shartli ravishda beriladi, so’ngra esa, bu argumentdan funkstiya o’zagida foydalanish mumkin.

## Funkstiyalarni yaratish

Funkstiyalarni yaratish uchun function kalit so’zidan foydalaniladi, uning umumiy sxemasi esa quyidagicha bo’ladi:

function some\_func (#argument\_1, #argument\_2);

{

//funkstiya uzagi

}

Funkstiya nomi function so’zidan keyin beriladi va qavslar ichida argumentlar keltiriladi. Agar yaratilayotgan funkstiya hech qanday argument talab qilmasa ham qavslar qo’yilishi shart.

Keltirib o’tgan fikrlarni yanada mustahkamlash uchun quyidagi misolni keltirib o’taylik:

*Misol-17.2. Funkstiyani aniqlash*

*<html>*

*<head>*

*<title> funkstiyani aniqlash </title>*

*</head>*

*<body>*

*<?php*

*function bighello()*

*{*

*print “<h1>HELLO</h>”;*

*}*

*bighello()*

*?>*

*</body>*

*</html>*

Bu dastur brauzer oynasiga faqat «HELLO» yozuvini chiqarib beradi. Dasturda esa, funkstiya nomini bighello() deb nomlandi, hamda bu funkstiya hech qanday argument talab qilmaydi. Misolda keltirib o’tilgan funkstiya barcha talablarga to’g’ri keladi.



*17.2-rasm.*

Endi esa, argument talab qiluvchi quyidagi misolni keltirib o’taylik:

*Misol-17.3. Argument talab kiluvchi funkstiyaga misol*

*<html>*

*<head>*

*<title> Argument talab qiluvchi funksiyaga misol</title>*

*</head>*

*<body>*

*<?php*

*function printBR($txt)*

*{*

*print ("$txt<br>\n");*

*}*

*printBR("Bu qator");*

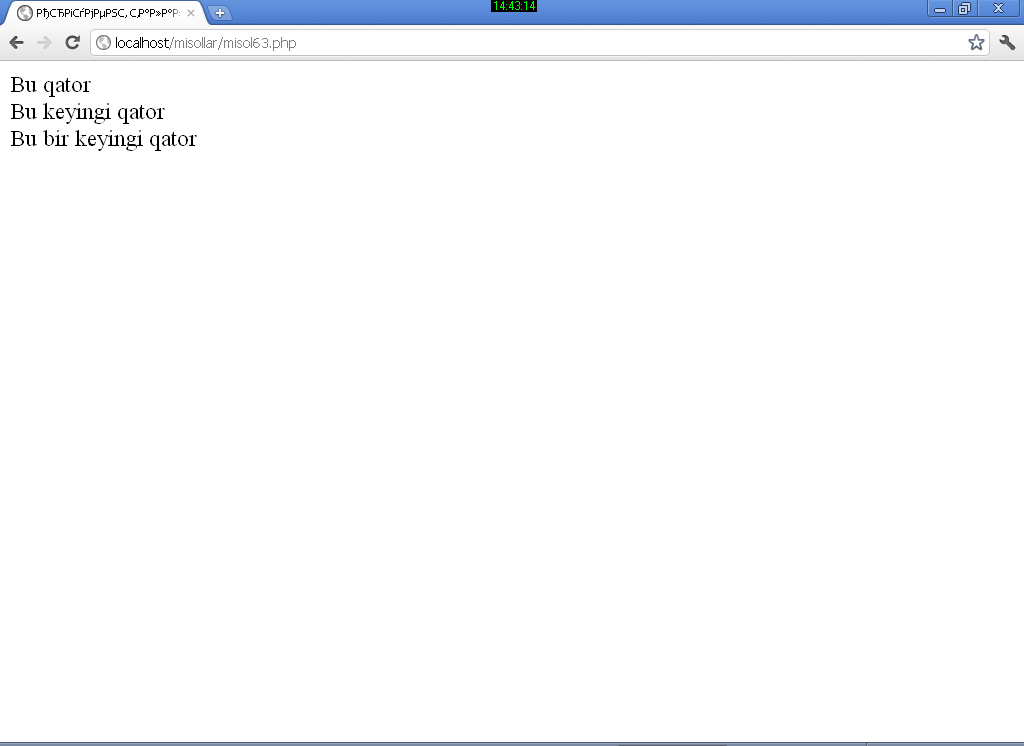
*printBR("Bu keyingi qator");*

*printBR("Bu bir keyingi qator");*

*?>*

*</body>*

*</html>*



*17.3-rasm.*

PrintBR() funkstiyasi argument kiritilishini talab qiladi. Argument tipi esa, matnli bo’ladi va uning nomi $txt deb atalgan. Endi bu funkstiya yordamida ixtiyoriy yozuvni brauzerga chop qilish mumkin.

## Qiymat qaytaruvchi funkstiyalar yaratish

Funkstiya qiymat qaytarishi yoki return operatori yordamida ob’ekt qaytarishi mumkin. Bu operator funkstiya ishini tugatadi va qaytarilishi lozim bo’lgan qiymatni chaqirilgan bosh dasturga qaytaradi.

Bu fikrlarni mustahkamlash maqsadida 2 sonning yig’indisini hisoblashga doir quyidagi misolni keltirib o’taylik.

*Misol-17.4. Qiymat qaytaruvchi funkstiya*

*<html>*

*<head>*

*<title> Qiymat qaytaruvchi funksiya </title>*

*</head>*

*<body>*

*<?php*

*function addNums($firstnum, $secondnum)*

*{*

*return $result = $firstnum + $secondnum;*

*}*

*print addNums(3, 5); // natija 8 chiqadi*

*?>*

*</body>*

*</html>*

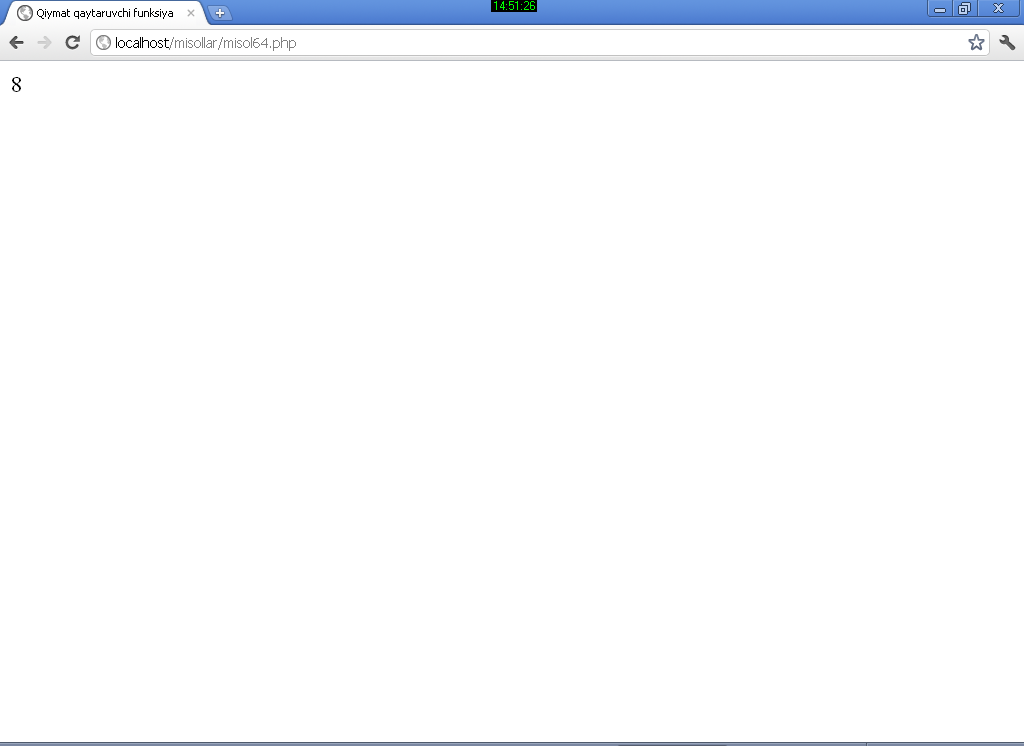
Bu dasturning bajarilishi natijasida brauzerda 8 chop qilinadi. Chunki addNums() funkstiyasining argumentlari 3 va 5 qiymatlarni qabul qiladi. Avvalroq aytib o’tilganidek, dasturni yanada ixchamroq ko’rinishga keltirish mumkin. U holda funkstiyaning o’zagi quyidagi ko’rinishga ega bo’ladi.

*{*

*return ($firstnum+$secondnum);*

*}*

return operatori qiymat, ob’ekt qaytarishi yoki hech narsa qaytarmasligi ham mumkin.



*17.4-rasm.*

Return operatori yordamida qiymat qaytarishning bir necha usullari mavjud bo’lib ular:

return4; - o’zgarmas;

return($a/$b); - ifoda yoki

return(function($argument)) - bo’lishi mumkin.

## Funkstiyalarga dinamik murojaat qilish

Funkstiya nomini biror matnli tipga ega bo’lgan o’zgaruvchiga o’zlashtirish mumkin, hamda xuddi shu o’zgaruvchiga funkstiya murojaat qilgandek ishlash mumkin. Fikrimizni yanada aniqroq tasavvur qilish uchun esa, quyidagi misolni keltirib o’taylik:

*Misol-17.5. Funkstiyalarga dinamik murojaat*

*<html>*

*<head>*

*<title> Funksiyalarga dinamik murojaat</title>*

*</head>*

*<body>*

*<?php*

*function sayHello()*

*{*

*print "Hello <br>";*

*}*

*$function\_holder="sayHello";*

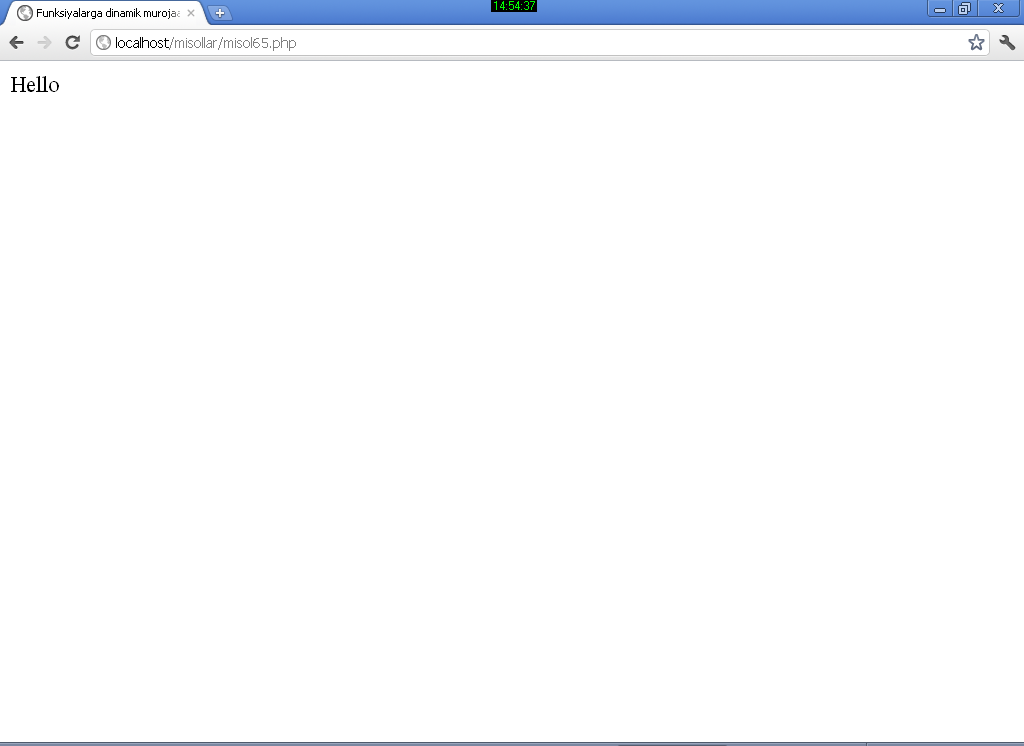
*$function\_holder();*

*?>*

*</body>*

*</html>*

Bu erda $function-holder o’zgaruvchisi matnli tipga ega va u Sayhello() funkstiyasi nomini o’zlashtiradi. Keyin esa, xuddi shu funkstiyaga murojaat qilgandek $function-folder() ga murojaat qilish mumkin.



*17.5-rasm.*

Albatta, tabiiy savol tug’iladi: «Funkstiyaga murojaat qilishning bu usulidan nima naf bor?». Bu misolda ortiqcha ish bajarildi. Lekin boshqa katta dasturlarda bu usulning yutug’i katta bo’ladi. Masalan, foydalanuvchi talabiga asosan dastur holatini o’zgartirishga to’g’ri keladi. Shunda qatorli parametr (o’zgaruvchiga) o’zlashtirilgan funkstiya nomi bo’yicha o’zgartirish mumkin.

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

* 1. *PHPda funksiya qanday yaratiladi?*
  2. *Funksiyaga murojaat qanday bo’ladi?*
  3. *Funsiya argumenti nima?*