

Satrlar



Satrlar

Satrlar ko'rinishi:

- apostrofli
- qo'shtirnoqli (maxsus belgilarni chop etish va o'zgaruvchilarni qayta ishlash imkoni mavjud)
- heredoc sintaksisi (qo'shtirnoqliga mos)
- nowdoc sintaksisi (apostrofliga mos)

```
<?php
```

```
$a = 1;
```

```
echo 'a ning qiymati = $a';
```

```
echo "a ning qiymati = $a";
```

```
echo <<<SATR
```

```
heredoc satr\n
```

```
a = $a
```

```
SATR;
```

```
echo <<<'SATR'
```

```
nowdoc satr\n
```

```
a = $a
```

```
SATR;
```

a ning qiymati = \$a

a ning qiymati = 1

heredoc satr

a = 1

nowdoc satr\n

a = \$a

Nowdoc va heredoc satrlar

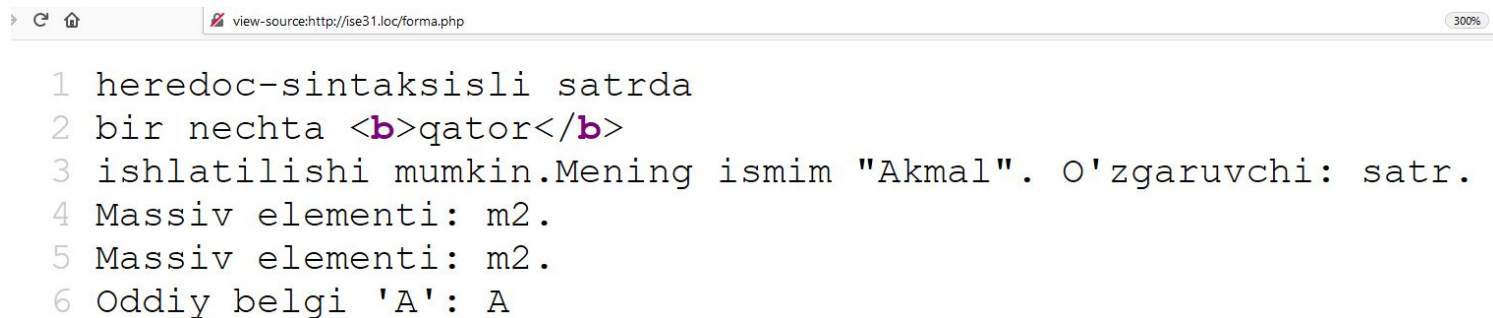
Qoidalar:

1. <<< belgisi va indentifikatordan boshlanadi, keyin yangi qatordan satr(lar) yoziladi va oxirida identifikator bilan yakunlanadi.
2. Yopuvchi identifikator yangi qatordan yozilishi shart.
3. Yopuvchi identifikatorda faqat ; belgisi qatnashishi mumkin.
4. Yopuvchi identifikatordan oldin bo'sh simvollar (probellar yoki tabulyatsiya) bo'lmasligi kerak.

Nowdoc va heredoc satrlar

```
<?php
$satr1 = <<<EOD
heredoc-sintaksisli satrda
bir nechta <b>qator</b>
ishlatilishi mumkin.
EOD;
echo $satr1;

$satr2 = 'satr';
$massiv = array('m1', 'm2', 'm3');
$ism = "Akmal";
echo <<<EOT
Mening ismim "$ism". O'zgaruvchi: $satr2.
Massiv elementi: $massiv[1].
Massiv elementi: {$massiv[1]}.
Oddiy belgi 'A': \x41
EOT;
```



```
1 heredoc-sintaksisli satrda
2 bir nechta <b>qator</b>
3 ishlatilishi mumkin.Mening ismim "Akmal". O'zgaruvchi: satr.
4 Massiv elementi: m2.
5 Massiv elementi: m2.
6 Oddiy belgi 'A': A
```

Nowdoc va heredoc satrlar

```
<?php
class Sinf
{
    var $satr2;
    var $massiv;
    function __construct()
    {
        $this->satr2 = 'satr';
        $this->massiv = array('sm1', 'sm2', 'sm3');
    }
}
$sinf = new Sinf();
echo <<<EOT
O'zgaruvchi: $sinf->satr2.
Massiv elementi: $sinf->massiv[1].
Massiv elementi: {$sinf->massiv[1]}.
EOT;
```



view-source:http://se31.loc/forma.php

```
1 O'zgaruvchi: satr.
2 Massiv elementi: Array[1].
3 Massiv elementi: sm2.
```

Satrlar – belgilardan iborat massivlar

```
$string = 'Salom YTIT';  
echo "Indeksi 4 bo'lgan belgi - $string[4].", PHP_EOL;  
echo "Indeksi -2 bo'lgan belgi - $string[-2].", PHP_EOL;  
$string[1] = 'o';  
echo "1-indeksda joylashgan simvolni o'zgartirildi: $string.", PHP_EOL;
```



view-source:http://ise31.loc/satr.php

```
1 Indeksi 4 bo'lgan belgi - m.  
2 Indeksi -2 bo'lgan belgi - I.  
3 1-indeksda joylashgan simvolni o'zgartirildi: Solom YTIT.  
4
```

Murakkab sintaksis

Murakkab ifodalarni qo'llash imkonini beradi, { } belgilari orasida yoziladi.

```
<?php
$ism = 'Akmal';
echo "Salom {$ism}!";    // ishlaydi
echo "Salom { $ism}\n"; // murakkab sintaksis ishlamaydi,
// Qo'shimchalar qo'shishda
echo "Salom $ismjon\n";  // ishlamaydi
echo "Salom {$ism}jon\n"; // ishlaydi

$uzgaruvchi = 'ism';
function getIsm() {
    return 'ism';
}
echo "O'zgaruvchilar o'zgaruvchisi: ${$uzgaruvchi}}\n";
echo "getIsm() funksiyasi qaytaradigan qiymat: {getIsm()}\n";
echo "getIsm() funksiyasi orqali o'zgaruvchi hosil qilish: ${getIsm()}\n";
echo "Obyekt metodi orqali o'zgaruvchi hosil qilish \${object->getName()}: ${$object->getName()}}\n";
;
echo "Murakkab ifoda namunasi: {$obj->values[3]->name}";
```

Murakkab sintaksis

Murakkab ifodalarni qo'llash imkonini beradi, { } belgilari orasida yoziladi.

```
$massiv = array("salom", "YTIT", "guruh" => "ISE",  
               array("ichki", "massiv", "elementlari"));  
  
echo "Massivga murojaat 1: $massiv[1]\n";           // ishlaydi  
echo "Massivga murojaat 2: $massiv['guruh']\n";     // ishlamaydi, error  
echo "Massivga murojaat 3: {$massiv['guruh']}\n";   // ishlaydi  
  
define('konstanta', 'guruh');  
  
echo "Massivga murojaat 4: {$massiv[konstanta]}\n"; // ishlaydi  
echo "Massivga murojaat 5: $massiv[2][0]\n";       // ishlamaydi  
echo "Massivga murojaat 6: {$massiv[2][0]}\n";     // ishlaydi
```


Satrga keltirish

Quyidagi usullar orqali boshqa turdan satrga keltirish mumkin:

1. (string) turni almashtirish orqali
2. strval() funksiyasi orqali
3. Ifodalarda satr zarur bo'lganda avtomatik keltirish amalga oshiriladi
 - a. Mantiqiy tur qiymatlari: true – "1"ga, false – "", bo'sh satrga aylantiriladi.
 - b. Butun va haqiqiy sonlar raqamlari va belgilaridan iborat satrga aylantiriladi.
 - c. Massivlar "Array" satriga aylantiriladi, satrda elementlari zarur bo'lganda maxsus kod yozish kerak
 - d. "Object" turidagi o'zgaruvchilari uchun __toString sehrli metodi ishlatiladi.
 - e. null qiymati bo'sh satrga ("") aylantiriladi

Foydali funksiyalar

str_split – satrni massivga aylantirish (*str_split(string \$string, int \$length = 1) : array*)

explode – satrni massivga aylantirish (*explode(string \$separator, string \$string, int \$limit = PHP_INT_MAX) : array*)

implode – massiv elementlarini satrga birlashtirish (*implode(string \$separator, array \$array) : string*)

<?php

```
$array = array('Akmal', 'Jamshid', 'Qosim');  
$string = implode(",", $array);  
echo $string, PHP_EOL; // Akmal,Jamshid,Qosim
```

```
$m1 = str_split($string, 4);  
$m2 = explode("a", $string);
```

```
print_r($m1);  
print_r($m2);
```

```
view-source: http://ee31.loc/satr.php  
1 Akmal,Jamshid,Qosim  
2 Array  
3 (  
4     [0] => Akma  
5     [1] => l,Ja  
6     [2] => mshi  
7     [3] => d,Qo  
8     [4] => sim  
9 )  
10 Array  
11 (  
12     [0] => Akm  
13     [1] => l,J  
14     [2] => mshid,Qosim  
15 )  
16
```

Foydali funksiyalar

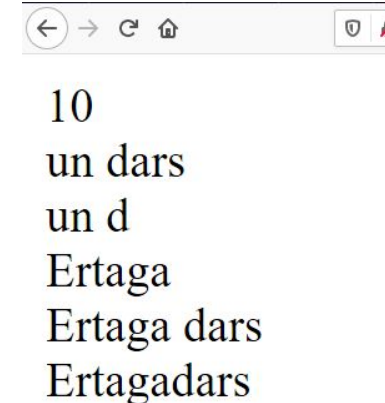
strlen – satr uzunligini aniqlash (*strlen(string \$string) : int*)

substr – qism satrni qaytaradi (*substr(string \$string , int \$offset, int|null \$length = null) : string*)

substr_replace – satr qismini almashtiradi (*substr_replace(array|string \$string, array|string \$replace, array|int \$offset, array|int|null \$length = null) : string|array*)

<?php

```
$string = "Bugun dars";  
echo strlen($string) . "<br />\n";  
echo substr($string, 3) . "<br />\n";  
echo substr($string, 3, 4) . "<br />\n";  
echo substr_replace($string, 'Ertaga', 0) . "<br />\n";  
echo substr_replace($string, 'Ertaga', 0, 5) . "<br />\n";  
echo substr_replace($string, 'Ertaga', 0, strlen($string) - 4) . "<br />\n";
```



10
un dars
un d
Ertaga
Ertaga dars
Ertagadars

Foydali funksiyalar

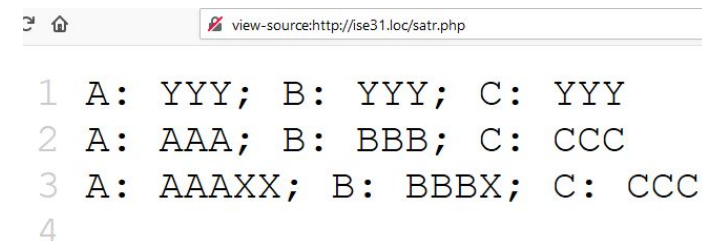
strlen – satr uzunligini aniqlash (*strlen(string \$string) : int*)

substr – qism satrni qaytaradi (*substr(string \$string , int \$offset, int|null \$length = null) : string*)

substr_replace – satr qismini almashtiradi (*substr_replace(array|string \$string, array|string \$replace, array|int \$offset, array|int|null \$length = null) : string|array*)

```
<?php
```

```
$input = array('A: XXX', 'B: XXX', 'C: XXX');  
echo implode('; ', substr_replace($input, 'YYY', 3, 3)) . "\n";  
  
$replace = array('AAA', 'BBB', 'CCC',);  
echo implode('; ', substr_replace($input, $replace, 3, 3)) . "\n";  
  
$length = array(1, 2, 3);  
echo implode('; ', substr_replace($input, $replace, 3, $length)) . "\n";
```



```
1 A: YYY; B: YYY; C: YYY  
2 A: AAA; B: BBB; C: CCC  
3 A: AAAXX; B: BBBX; C: CCC  
4
```

Foydali funksiyalar

number_format – sonni formatlab chop etish, (*number_format(float \$num, int \$decimals = 0, string|null \$decimal_separator = ".", string|null \$thousands_separator = ",") : string*)

```
$number = 1234.5678;  
echo number_format($number);           // 1,235  
echo number_format($number, 2, ',', ' '); // 1 234,56  
echo number_format($number, 2, '.', ''); // 1234.57
```

strcmp – satrlarni solishtirish, (*strcmp(string \$str1, string \$str2) : int*)

```
$str1 = "Salom";  
$str2 = "Alik";  
$str3 = "salom";  
$str4 = "Alik";  
echo strcmp($str1, $str2); // 1  
echo strcmp($str2, $str1); // -1  
echo strcmp($str2, $str3); // -1  
echo strcmp($str2, $str4); // 0
```

Foydali funksiyalar

1. **addslashes**(string \$string) : string – satrdagi maxsus belgilarga sleshtarlar qo'yib chiqadi (' " \ NUL)
2. **stripslashes** (string \$string) : string – satrdagi ekranlashgan simvollarni o'chiradi
3. **htmlentities** (string \$string, int \$flags = ENT_COMPAT, string|null \$encoding = null, bool \$double_encode = true) : string
Belgilarni HTML kod ko'rinishiga o'tkazadi.
4. **html_entity_decode** (string \$string, int \$flags = ENT_COMPAT, string|null \$encoding = null) : string
HTML kodlarni mos belgilarga aylantiradi.
5. **htmlspecialchars** (string \$string, int \$flags = ENT_COMPAT, string|null \$encoding = null, bool \$double_encode = true) : string
Maxsus belgilarni HTML kod ko'rinishiga o'tkazadi.
6. **htmlspecialchars_decode** (string \$string, int \$flags = ENT_COMPAT) : string – HTML kodlarni mos belgilarga aylantiradi.

Foydali funksiyalar

```
<?php
$str1 = "A 'quote' is <b>bold</b>";
$str2 = "A \'quote\' is <b>bold</b>";
echo addslashes($str1), PHP_EOL; // A \'quote\' is <b>bold</b>
echo stripslashes($str2), PHP_EOL; // A 'quote' is <b>bold</b>

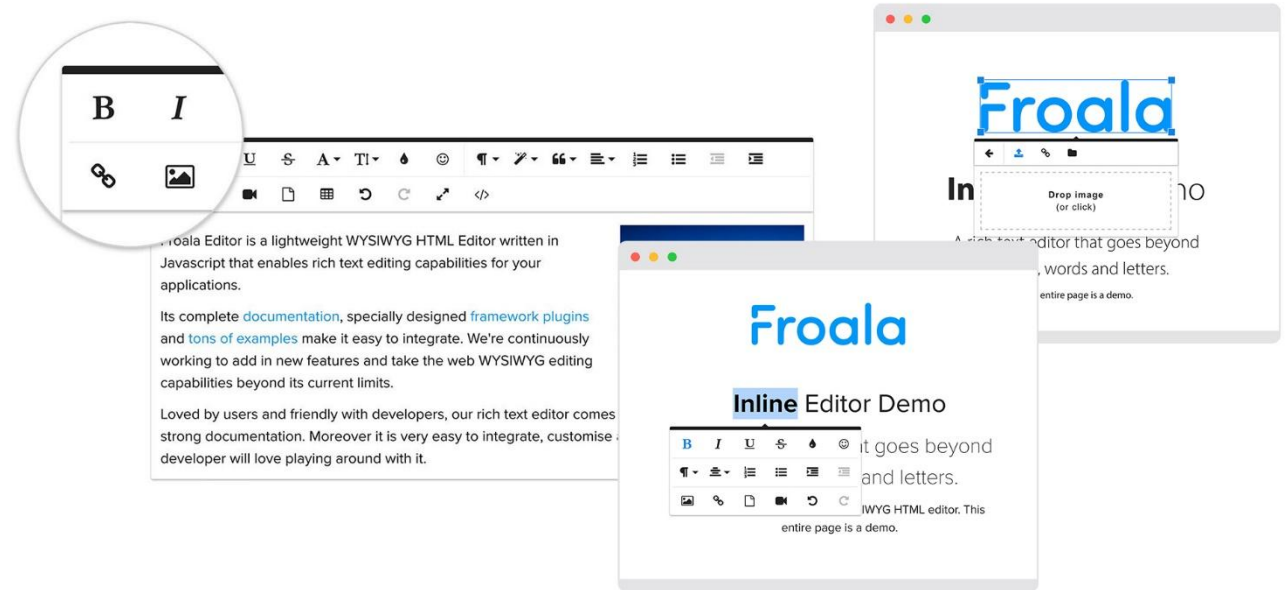
$str = "A 'quote' is <b>bold</b>, éê";
echo htmlentities($str), PHP_EOL; // A 'quote' is &lt;b&gt;bold&lt;/b&gt;, &acute;&ecirc;
echo htmlentities($str, ENT_QUOTES), PHP_EOL; // A &#039;quote&#039; is &lt;b&gt;bold&lt;/b&gt;, &acute;&ecirc;
echo html_entity_decode(htmlentities($str)), PHP_EOL; // A 'quote' is <b>bold</b>, éê

echo htmlspecialchars($str), PHP_EOL; // A 'quote' is &lt;b&gt;bold&lt;/b&gt;, éê
echo htmlspecialchars($str, ENT_QUOTES), PHP_EOL; // A &#039;quote&#039; is &lt;b&gt;bold&lt;/b&gt;, éê
echo htmlspecialchars_decode(htmlspecialchars($str)), PHP_EOL; // A 'quote' is <b>bold</b>, éê
```

WYSIWYG

WYSIWYG - What You See Is What You Get

1. Froala (<https://froala.com/>)
2. TinyMCE (<https://www.tiny.cloud/>)
3. CKEditor (<https://ckeditor.com/>)
4. Setka Editor (<https://setka.io/>)
5. Summernote (<https://summernote.org/>)
6. Vev (<https://www.vev.design/>)
7. Bubble (<https://bubble.io/>)



Regulyar ifoda

Regulyar ifoda – matndagi qism satrlarni qidirish, ularni qayta ishlash uchun formal til. Tilning matematik asosi 1951 yilda matematik Stephen Cole Kleene tomonidan ishlab chiqilgan.

Regular Expression

Regulyar ifoda - ajratuvchilar

PCRE shablonlarini ajratuvchilarda yozish zarur.

Ajratuvchi – harf, raqam, teskari taqsim (\ , backslash) va probelli belgi bo'lmagan ixtiyoriy belgi.

Ko'p hollarda /, #, ~ belgilari ishlatiladi.

Namunalar:

/foo bar/

#^[^0-9]\$#

+php+

%[a-zA-Z0-9_-]%

Regulyar ifoda - metabelgilar

| Metabelgi | Ta'rifi |
|----------------------------------|---|
| \ | umumiy ekranlashtirish belgisi |
| ^ | berilganlarning boshlanishi |
| \$ | berilganlarning tugashi |
| . | yangi qatordan tashqari ixtiyoriy belgi |
| [] | simvolli sinf boshlanishi va tugashi |
| | tanlash qismining boshlanishi |
| () | qism maska boshlanishi va tugashi |
| ? | metabelgi ma'nosini kengaytirish ((?=...) – Positive lookahead, (?!...) – Negative lookahead, (?<=...) – Positive lookbehind, (?<!...) – Negative lookbehind) |
| * | kvantifikator, 0 va undan ko'p |
| + | kvantifikator, 1 va undan ko'p |
| { } | miqdorli kvantifikatorning boshlanishi va tugashi |
| [] ichidagi metabelgilar | |
| ^ | sinfning inkori, faqat boshida berilishi mumkin |
| - | simvolli interval |

Regulyar ifoda - ekranlashtirish

\n – keying qator (0A)

\r – karetkani qaytarish (0D)

\R – \r, \n, \r\n ga mos

\t – tabulyatsiya

\xhh – 16 s.s.dagi hh kodli simvol

\d – ixtiyoriy raqam

\D – raqamdan boshqa ixtiyoriy belgi

\s – ixtiyoriy probelli belgi

\S - ixtiyoriy probelli bo'lmagan belgi

Regulyar ifoda – simvolli sinflar

| | |
|--------|---|
| alnum | harflar va raqamlar |
| alpha | harflar |
| ascii | 0 – 127 kodli simvollar |
| blank | faqat probel va tabulyatsiya belgisi |
| cntrl | boshqaruv belgilari |
| digit | o'nlik s.s. raqamlari (\d) |
| graph | pechatli simvollar (probelsiz) |
| lower | kichik registrli harflar |
| print | pechatli simvollar (probel bilan) |
| punct | pechatli simvollar, harflar va raqamlardan tashqari |
| space | probelli simvollar |
| upper | katta registrli harflar |
| word | so'z simvollar (\w) |
| xdigit | 16 s.s. raqamlar |

Regulyar ifoda – simvolli sinflar

[aeiou] – a, e, i, o, u harflari

[^aeiou] – a, e, i, o, u bo'lmagan harflar

[d-m] – d va m oralig'idagi harflar

[\dABCDEF] – 16 s.s.dagi sonlar

[01[:alpha:]] – 0, 1, alfavit belgilari, % belgisi

[12[:^digit:]] – 1, 2, va raqam bo'lmagan ixtiyoriy belgi

Regulyar ifoda – qism maskalar va qaytarilishlar miqdori

Qism maskalar - () orqali aniqlanadi.

cat(aract|erpillar|) - "cat","cataract" yoki "caterpillar" so'zlaridan biriga mos keladi

the ((red|white) (king|queen)) – "the red king" matni uchun "the red king", "red king", "red", "king" so'zlarini topadi.

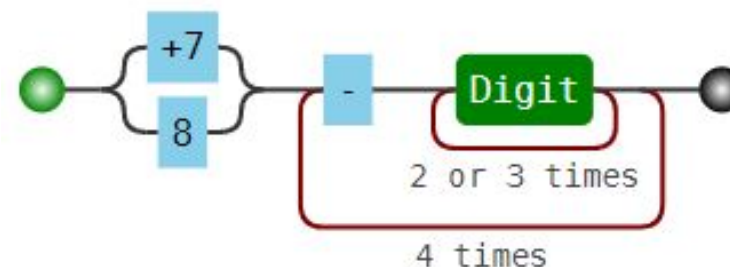
Qaytarilishlar miqdori - { } orqali aniqlanadi.

{2, 4} – 2 tadan 4 tagacha

{4, } – 4 tadan cheksizgacha

{5} – 5ta

RegExp: `/(?:\+7|8)(?:-\d{2,3}){4}/`



Regulyar ifoda bilan ishlovchi funksiyalar

preg_filter — regulyar ifoda bo'yicha qidirish va almashtirishni amalga oshirish

preg_grep — massivdan shablonga mos keluvchi massiv elementlarini qaytaradi

preg_last_error_msg — PCRE oxirgi funksiyasidan kelgan xato xabarini qaytaradi

preg_last_error — PCRE oxirgi regulyar ifoda ishlashidan qaytgan xato kodini qaytaradi

preg_match_all — Satrdan shablon bo'yicha qidiruvni amalga oshiradi, mosliklar sonini qaytaradi

preg_match — Regulyar ifoda bo'yicha mavjudlikni tekshiradi

preg_quote — Ifodadagi simvollarni ekranlashtiradi

preg_replace_callback_array — Regulyar ifoda bo'yicha qidirish va callback funksiya orqali almashtirishni amalga oshiradi

preg_replace_callback — Regulyar ifoda bo'yicha qidirish va callback funksiya orqali almashtirishni amalga oshiradi

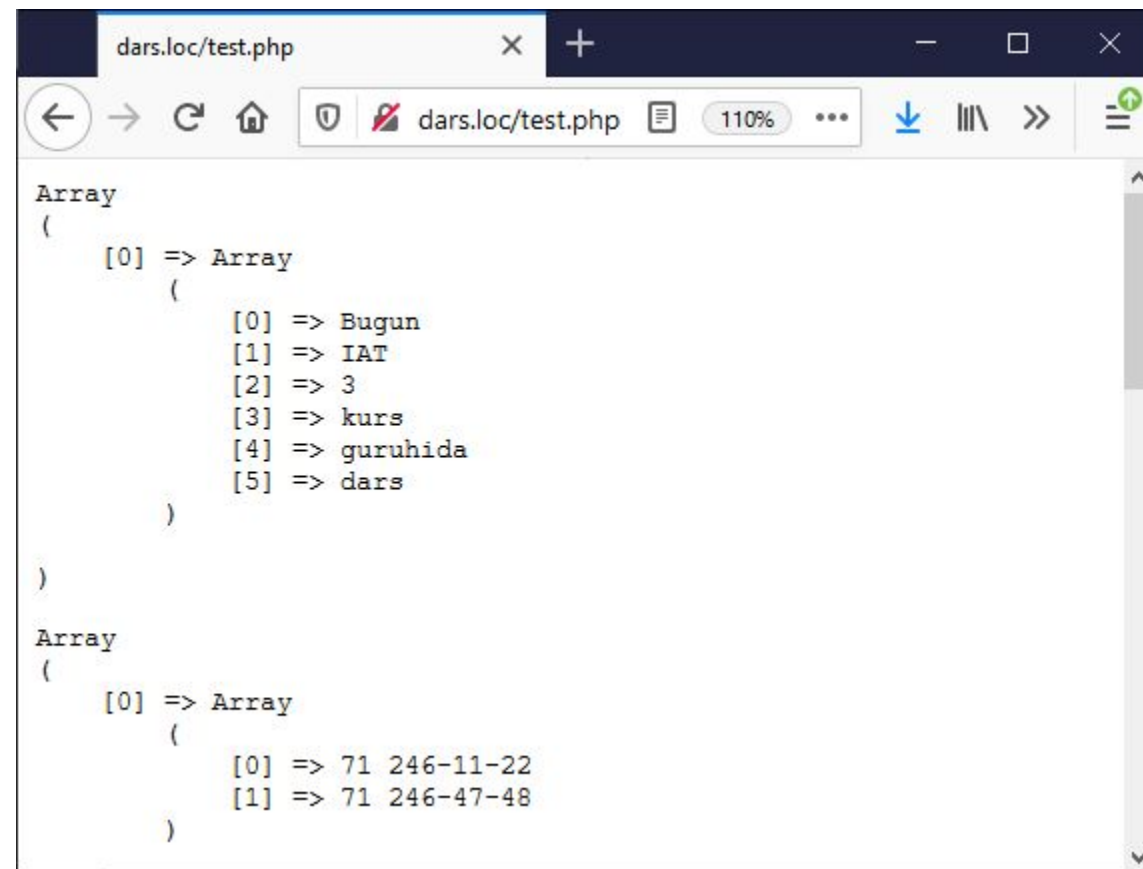
preg_replace — Regulyar ifoda bo'yicha qidirish va almashtirishni amalga oshiradi

preg_split — Regulyar ifoda bo'yicha satrni qismlarga ajratadi

Regulyar ifoda, namuna, qidirish

```
<?php
$text = "Bugun IAT 3-kurs guruhida dars";
$pattern = '/\w+/';
preg_match_all($pattern, $text, $matches);
print "<pre>";
print_r($matches);

preg_match_all(
    "/\d{2} \d{3}-\d{2}-\d{2}/",
    "Telefon qiling: 71 246-11-22, +99871 246-47-48",
    $phones
);
print "<pre>";
print_r($phones);
```

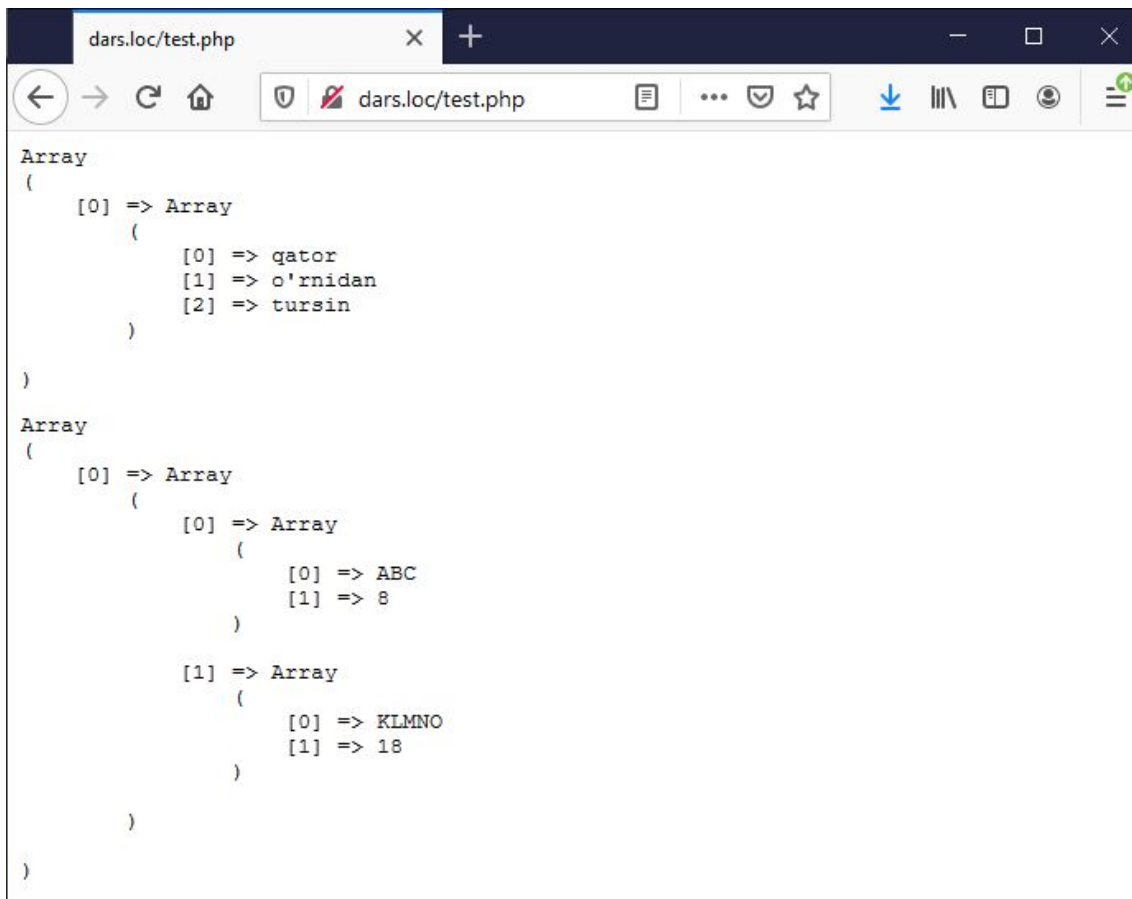


```
Array
(
    [0] => Array
        (
            [0] => Bugun
            [1] => IAT
            [2] => 3
            [3] => kurs
            [4] => guruhida
            [5] => dars
        )
)
Array
(
    [0] => Array
        (
            [0] => 71 246-11-22
            [1] => 71 246-47-48
        )
)
```

Regulyar ifoda, namuna, sinflar bilan qidirish

```
<?php
$text = "2-qator tez o'rnidan tursin";
$pattern = '/[a-z\']{4,}/';
preg_match_all($pattern, $text, $matches);
print "<pre>";
print_r($matches);

$text = "2-qator ABCdefghijKLMNOPqrstuvwxyz";
$pattern = '/[\-[:upper:]]{3,5}/';
preg_match_all($pattern, $text, $matches,
    PREG_OFFSET_CAPTURE);
print "<pre>";
print_r($matches);
```

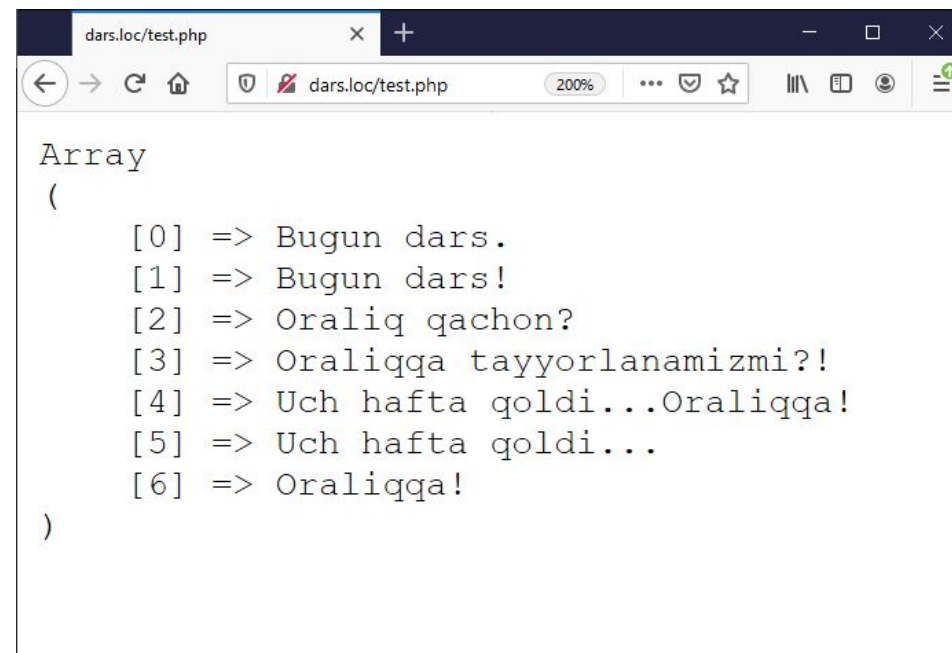


```
Array
(
    [0] => Array
        (
            [0] => qator
            [1] => o'rnidan
            [2] => tursin
        )
)
Array
(
    [0] => Array
        (
            [0] => Array
                (
                    [0] => ABC
                    [1] => 8
                )
            [1] => Array
                (
                    [0] => KLMNO
                    [1] => 18
                )
        )
)
```

Regulyar ifoda, namuna, ajratish (bo'lish)

<?php

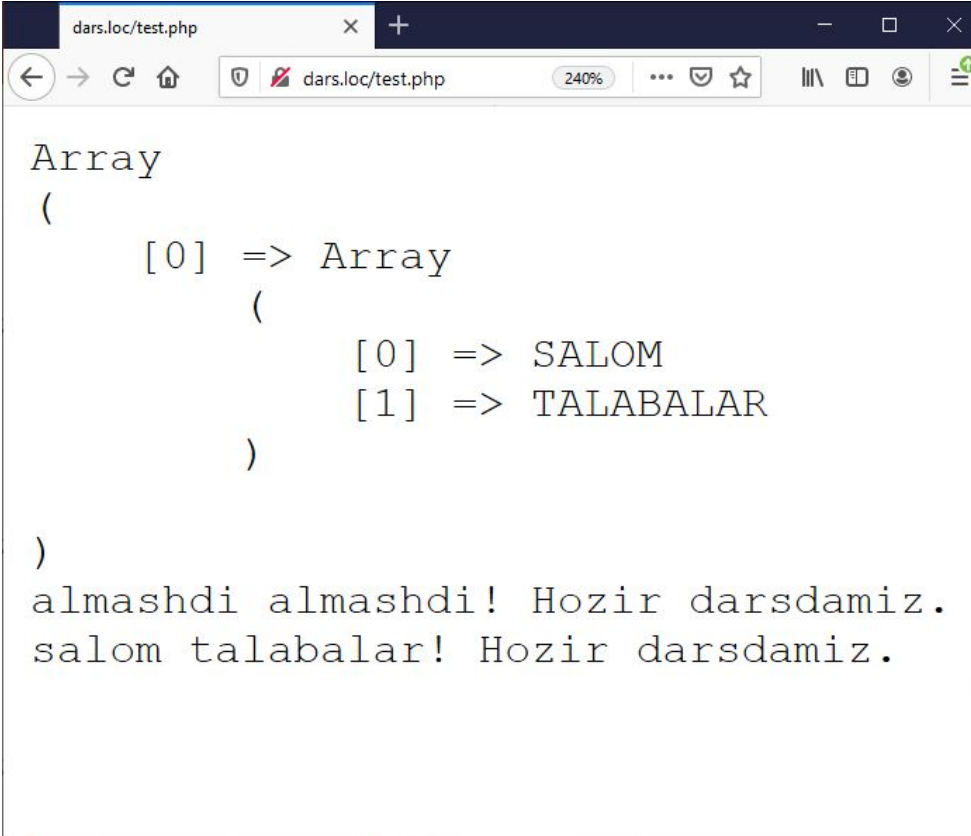
```
$str = "Bugun dars. Bugun dars! Oraliq qachon? Oraliqqa tayyorlanamizmi?!  
Uch hafta qoldi...Oraliqqa! Uch hafta qoldi... Oraliqqa!";  
$sentences = preg_split('/(?<=[.?!])\s+(?=[a-z])/i', $str);  
echo "<pre>";  
print_r($sentences);  
echo "</pre>";
```



```
Array  
(  
    [0] => Bugun dars.  
    [1] => Bugun dars!  
    [2] => Oraliq qachon?  
    [3] => Oraliqqa tayyorlanamizmi?!  
    [4] => Uch hafta qoldi...Oraliqqa!  
    [5] => Uch hafta qoldi...  
    [6] => Oraliqqa!  
)
```

Regulyar ifoda, namuna, almashtirish

```
<?php
$text = "SALOM TALABALAR! Hozir darsdamiz.";
$pattern = "/\b[A-Z]+\b/";
preg_match_all($pattern, $text, $matches);
echo "<pre>";
print_r($matches);
echo preg_replace($pattern, "almashdi", $text) . "<br>";
echo preg_replace_callback($pattern, function ($word) {
    return strtolower($word[0]);
}, $text);
```

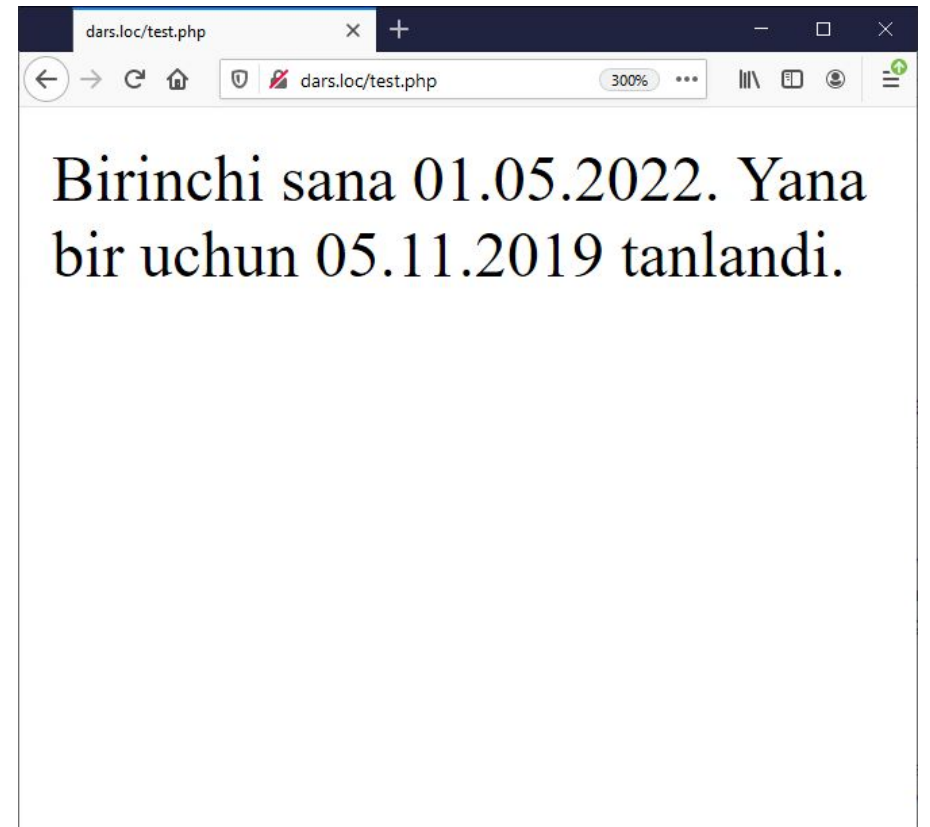


```
Array
(
    [0] => Array
        (
            [0] => SALOM
            [1] => TALABALAR
        )
)
almashdi almashdi! Hozir darsdamiz.
salom talabalar! Hozir darsdamiz.
```

Regulyar ifoda

```
$text = "Birinchi sana 01.05.2021.\n";
$text .= "Yana bir uchun 05.11.2018 tanlandi.\n";
// callback funksiya
function next_year($matches)
{
    // guruhlash orqali qidirilganda
    // qism guruhlar qidirish natijasida
    // massivda hosil bo'ladi
    return $matches[1] . ($matches[2] + 1);
}

echo preg_replace_callback(
    "|(\d{2}.\d{2}).(\d{4})|",
    "next_year",
    $text
);
```

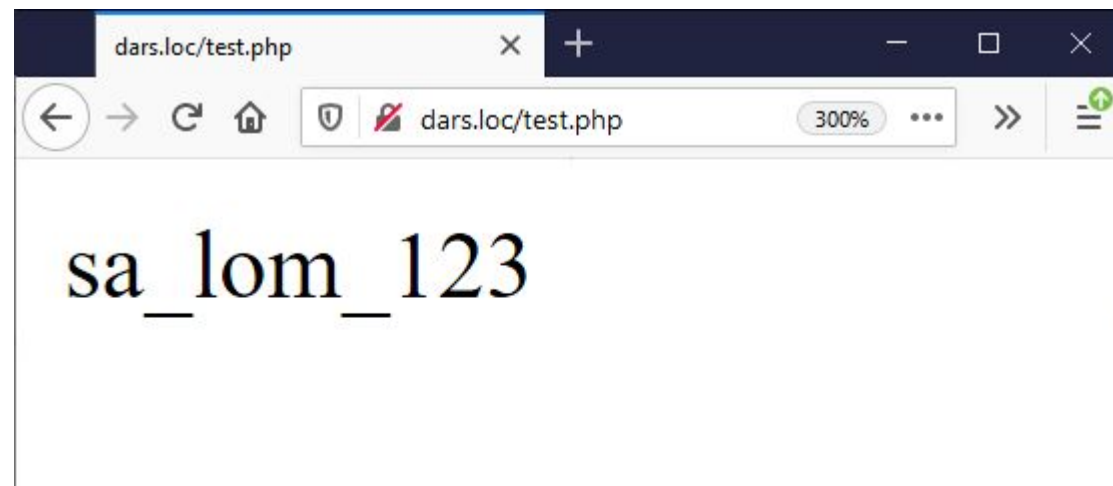


Regulyar ifoda, namuna, kengaytirish

```
<?php
```

```
function camelToUnderscore($string, $delimiter = "_")  
{  
    $pattern = '/(?<=\d)(?=[A-Za-z])|(?<=[A-Za-z])(?=\d)|(?<=[a-z])(?=[A-Z])/';  
    return strtolower(preg_replace($pattern, $delimiter, $string));  
}
```

```
echo camelToUnderscore('saLom123');
```



Savollar?
