



VILNIUSCODINGSCHOOL

# PHP 3

# Konstantos



- Konstanta - nekintanti reikšmė. Jos neįmanoma pakeisti vykdant kodą. Konstantas rašome didžiosiomis raidėmis.
- Konstantą deklaruojame su `define( )` funkcija, o panaudojame paprasčiausiai užrašydami jos pavadinimą.

```
<?php  
    define("MINSIZE", 50);  
    echo MINSIZE;  
?>
```

# Skirtumai tarp konstantų ir kintamųjų



- Nereikia rašyti \$ ženklo.
- Konstantų negalima deklaruoti tik priskiriant reikšmę - būtina naudoti **define( )** funkciją.
- Konstantas galima panaudoti bet kur mūsų failuose, kai tuo tarpu kintamųjų galiojimas gali būti ribotas (pvz. tik funkcijos viduje).
- Deklaravus konstantą, jos reikšmės keisti neįmanoma.

# PHP konstantos ir dokumentacija



PHP turi konstantų, kurios jau yra įtrauktos į pačią programinę kalbą. Pvz.:

**PHP\_VERSION** parodo PHP versiją.

**E\_ERROR** parodo klaidas.

**\_\_LINE\_\_** parodo eilučių kiekį faile.

**\_\_FILE\_\_** parodo kelią iki failo ir jo pavadinimą.

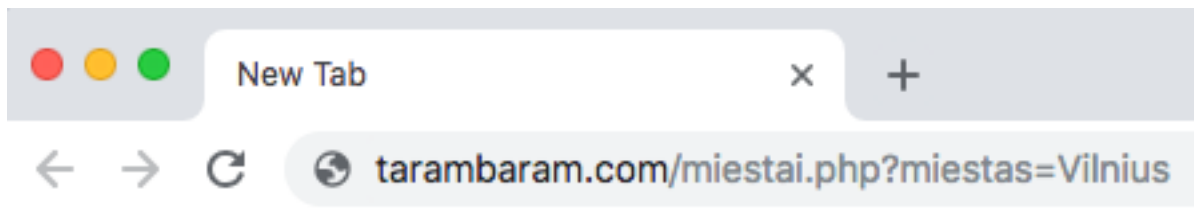
Pilna PHP dokumentacija su aprašymais ir pavyzdžiais:

[php.net/manual/en/](http://php.net/manual/en/)

# Kas yra GET metodas?



- HTTP užklausa naudojanti GET metodą:
  - GET naudojame, kai prašome duomenų iš kokio nors šaltinio
  - Parodo užklausą ir jos turinį URL, turi 1024 simbolių limitą
  - Negali siųsti nuotraukų ar dokumentų
  - GET užklausas galima pasižymėti (bookmark), išsaugoti cache, jos išlieka naršyklės istorijoje, skirtos duomenų gavimui.



# GET pavyzdys



```
<?php
    if($_GET['name'] && $_GET['age']) {
        echo 'Welcome ' . $_GET['name'] . ', <br>' . 'You are ' . $_GET['age'] . ' years old.';
        exit();
    }
?>

<html>
    <body>
        <form action="<?php $_PHP_SELF; ?>" method="GET">
            Name: <input type="text" name="name" />
            Age: <input type="text" name="age" />
            <input type="submit">
        </form>
    </body>
</html>
```

← → ↻ ⓘ localhost:8888/php/index.php?name=Tomas&age=26

Welcome Tomas,  
You are 26 years old.

# Kas yra POST metodas?



- HTTP užklausa naudojanti POST metodą:
  - POST naudojame, kai siunčiame duomenis šaltiniui
  - Neturi duomenų dydžių apribojimų
  - Užklausa ir turinys matomi tik pačioje užklausoje
  - Negalima pasižymėti (bookmark), išsaugoti 'cache', neišlieka naršyklės istorijoje

# POST pavyzdys



```
<?php
    if($_POST['name'] && $_POST['age']) {
        if(preg_match("/[^A-Za-z'-]/", $_POST['name'])) {
            die("Invalid name characters.")
        }
    }

    echo 'Welcome ' . $_POST['name'] . ', <br>' . 'You are ' . $_POST['age'] . ' years old.';
    exit();
?>

<html>
    <body>
        <form action="<?php $_PHP_SELF; ?>" method="POST">
            Name: <input type="text" name="name" />
            Age: <input type="text" name="age" />
            <input type="submit">
        </form>
    </body>
</html>
```



# Kas yra \$\_REQUEST?



- **\$\_REQUEST** kintamasis talpina savyje turinį iš \$\_GET, \$\_POST, \$\_COOKIE. Galime naudoti \$\_REQUEST, kai norime gauti formų duomenis.

```
<?php
    if($_REQUEST['name'] && $_REQUEST['age']) {
        echo 'Welcome ' . $_REQUEST['name'] . ', <br>' . 'You are ' . $_REQUEST['age'] . ' years old.';
        exit();
    }
?>

<html>
    <body>
        <form action="<?php $_PHP_SELF; ?>" method="POST">
            Name: <input type="text" name="name" />
            Age: <input type="text" name="age" />
            <input type="submit">
        </form>
    </body>
</html>
```

# Failų atidarymas, skaitymas, rašymas, uždarymas



- **fopen()** funkcija atidaro failus. Turi du parametrus: pirmas yra failo pavadinimas, antras - atidarymo būdas. Atidarymo būdai:
  - **r** atidaro failą tik skaitymui, kursorius failo pradžioje
  - **r+** atidaro failą ir skaitymui, ir rašymui, kursorius failo pradžioje
  - **w** atidaro failą tik rašymui, jei failo nėra, tuomet sukuria, kursorius failo pradžioje
  - **w+** atidaro failą ir skaitymui, ir rašymui, jei failo nėra, tuomet sukuria, kursorius failo pradžioje
  - **a** atidaro failą tik rašymui, jei failo nėra, tuomet sukuria, kursorius failo gale
  - **a+** atidaro failą ir skaitymui, ir rašymui, jei failo nėra, tuomet sukuria, kursorius failo gale

# failas.php: failo sukūrimas ir skaitymas: w+ ir r



```
<html>
  <head>
    <title>Failo skaitymas su PHP</title>
  </head>
  <body>
    <?php
      $filename = 'text.txt';
      $file = fopen($filename, "w+"); // sukuriame failą

      $filesize = filesize($filename); // filesize() nuskaityto failo dydį
      $filetext = fread($file, $filesize); // fread() skaito failo turinį
      fclose($file); // fclose() uždaro failą, parametras – pointeris į failą

      echo "Failo dydis: $filesize B";
      echo "<pre>$filetext</pre>";
    /*
      Turinio nematysime, kol nepakeisime w+ į r
      Taip pat turime įkelti kokį nors tekstą į naujai sukurtą failą ir išsaugoti.
    */
  ?>
</body>
</html>
```

# Cookies



- **Cookies** - tai tekstiniai failai išsaugomi vartotojo kompiuteryje ir padeda nustatyti vartotoją.
- Nustatant sugrįžtantį vartotoją yra atliekami šie žingsniai:
  - Serverio kodas siunčia cookies į naršyklę, pvz. vardas, amžius, ar ID
  - Naršyklė išsaugo šią informaciją lokaliame kompiuteryje.
  - Kitą kartą, kai naršyklė siunčia užklausą į tą patį web serverį, ji taip pat nusiunčia ir cookies informaciją serveriui, o šis panaudoja šią informaciją vartotojo identifikavimui.

# Cookies



```
HTTP/1.1 200 OK
Date: Fri, 04 Feb 2000 21:03:38 GMT
Server: Apache/1.3.9 (UNIX) PHP/4.0b3
Set-Cookie: name=xyz; expires=Friday, 04-Feb-07 22:03:38 GMT;
           path=/; domain=delfi.lt
Connection: close
Content-Type: text/html
```

- Cookies paprastai nustatomi HTTP header dalyje.
- Nurodytas laikas iki kada galioja
- Šiuo atveju, vardas ir domenas yra informacija, kuri išsaugoma
- Šį cookie galima pasiekti su `$_COOKIE` arba `HTTP_COOKIE_VARS['name']`
- `setcookie()` funkcija sukuria cookie, ir turi būti iškviečiama prieš `<html>`, talpina iki 6 parametrų: `setcookie(name, value, expire, path, domain, security)`

# Cookies



```
<?php
    setcookie("name", "Valdas Adamkus", time()+3600, "/", "", 0);
    setcookie("age", "92", time()+3600, "/", "", 0);
?>

<html>
    <head>
        <title>Kuriame ir pasiekiamo Cookies</title>
    </head>
    <body>
        <?php
            if(isset($_COOKIE["name"])) {
                echo 'Welcome, ' . $_COOKIE["name"] . '<br>';
                echo 'You are ' . $_COOKIE["age"] . ' years old.';
            } else {
                echo "Who are you?";
            }
        ?>
    </body>
</html>
```

- Sukūrėme 2 cookies - name ir age, abu galios 1 val visose direktorijose, nenurodėme domeno, krausis per http, ne per https.

- Pasiekiamo **\$\_COOKIE** dėka

- **isset( )** funkcija patikrina ar toks cookie nustatytas

# SESSION



- Tam, kad duomenys būtų pasiekiami visame tinklalapyje, galime naudoti PHP sesiją. Sesija serveryje sukuria laikiną failą, kuriame talpina registruotos sesijos kintamuosius ir jų reikšmes. Šie duomenys bus prieinami visuose tinklalapio puslapiuose to apsilankymo tinklalapyje metu.
- `session_start()` funkcija patikrina ar sesija jau pradėta ir jei ne, tuomet pradeda naują sesiją. Šią funkciją visuomet rašome failo viršuje.
- Sesijos kintamieji talpinami asociatyviniame masyve `$_SESSION[ ]`
- Sunaikiname sesiją ir visus jos kintamuosius su `session_destroy()` funkcija.
- `unset()` funkcija leidžia sunaikinti konkretų kintamąjį:  
`unset($_SESSION['kintamasis'])`

# Funkcijos



- Funkcija - kodo dalis, kuri priima tam tikrus duomenis (parametrus), atlieka nurodytus veiksmus ir gražina rezultatą. Kiekviena funkcija yra deklaruojama (aprašoma) ir iškviečiama (panaudojama).
- PHP kalboje yra daugybė funkcijų, kurios jau yra sukurtos ir jas galima naudoti be deklaravimo (pvz: fopen( ), unset( ) ir t.t.)
- Funkcijos deklaravimui naudojame raktinį žodį function:

```
<?php
    // Funkcijos deklaravimas
    function labas(){
        echo "Sveikas";
    }

    // Funkcijos iškvietimas
    labas();
?>
```



# Funkcijos su parametrais



```
<html>
  <head>
    <title>Funkcijos su parametrais</title>
  </head>
  <body>
    <?php
      function sudėtis($num1, $num2) {
        $suma = $num1 + $num2;
        echo "Dviejų skaičių suma: $suma";
      }

      sudėtis(10, 20);
    ?>
  </body>
</html>
```

- Funkcijos yra universalesnės, kai atlieka tik vieną veiksmą.
- Funkcijos pavadinimas aiškiai nurodo kokį veiksmą ji atlieka.

# Funkcijos su return



```
<html>
  <head>
    <title>Funkcijos su return</title>
  </head>
  <body>
    <?php
      function sudėtis($num1, $num2) {
        $suma = $num1 + $num2;
        return $suma;
      }

      $atsakymas = sudėtis(10, 20);
      echo "Atsakymas: $atsakymas";
    ?>
  </body>
</html>
```

# Užduotis 1



- Sukurkite funkciją su dviem parametrais  $a$  ir  $b$ , kurie yra stačiakampio kraštinių ilgiai. Funkcija apskaičiuos stačiakampio plotą.
- Rezultatą išveskite į naršyklę.

## Užduotis 2



- Sukurkite formą, kuri leis vartotojui įvesti stačiakampio ilgį ir plotį. Įvestus duomenis panaudokite stačiakampio ploto apskaičiavimui. Rezultatą išveskite į naršyklę.

## Užduotis 3



- **Sukurkite formą, kuri leis vartotojui įvesti savo vardą ir pavardę. Paspaudus mygtuką, šie duomenys turėtų būti išspausdinti naršyklėje.**