

# MAINĪGO DEFINĒŠANA

PHP

# MAINĪGO DEFINĒŠANA

- Mainīgos izmanto, lai saglabātu **skaitliskas** un **tekstuālas** vērtības.
- Nosaukuma pirmais simbols ir latīņu alfabēta burts vai apakšsvītra.
- Ja mainīgais ir definēts, to var pielietot šī faila iekšienē **nepārtraukti**.
- Visi mainīgie **sākas** ar simbolu **\$**, tad seko mainīgā nosaukums.

Mainīgie var saturēt burtus un ciparus	0-9 A-Z a-z un _
Mainīgais nedrīkst saturēt atstarpes	\$pirmais mainīgais \$pirmais_mainīgais
Mainīgais nedrīkst sākties ar ciparu	\$1 mainīgais \$mainīgais1 \$mainīgais_1
Mainīgos nevajag definēt pirms tam tiek pievienota vērtība	
Ja mainīgā vērtība satur tekstu, tad šis teksts ir jāliek pēdiņās, bet, ja ciparu – tad nav jāliek pēdiņās.	Protams, ka to var darīt, bet tas nav obligāti!
Mainīgie ir reģistrjutīgi	\$mainīgais      \$MAINīgais \$MAINIGAIS      \$mainIGAIS

## PIEMĒRS I

```
<?php  
$teksts ="Globālā  
tīmekļa  
tehnoloģijas";  
echo "Man patīk  
$teksts!";  
?>
```

Man patīk Globālā tīmekļa tehnoloģijas

## PIEMĒRS II

```
<?php  
$x = 10;  
$y = 6;  
echo $x + $y;  
?>
```

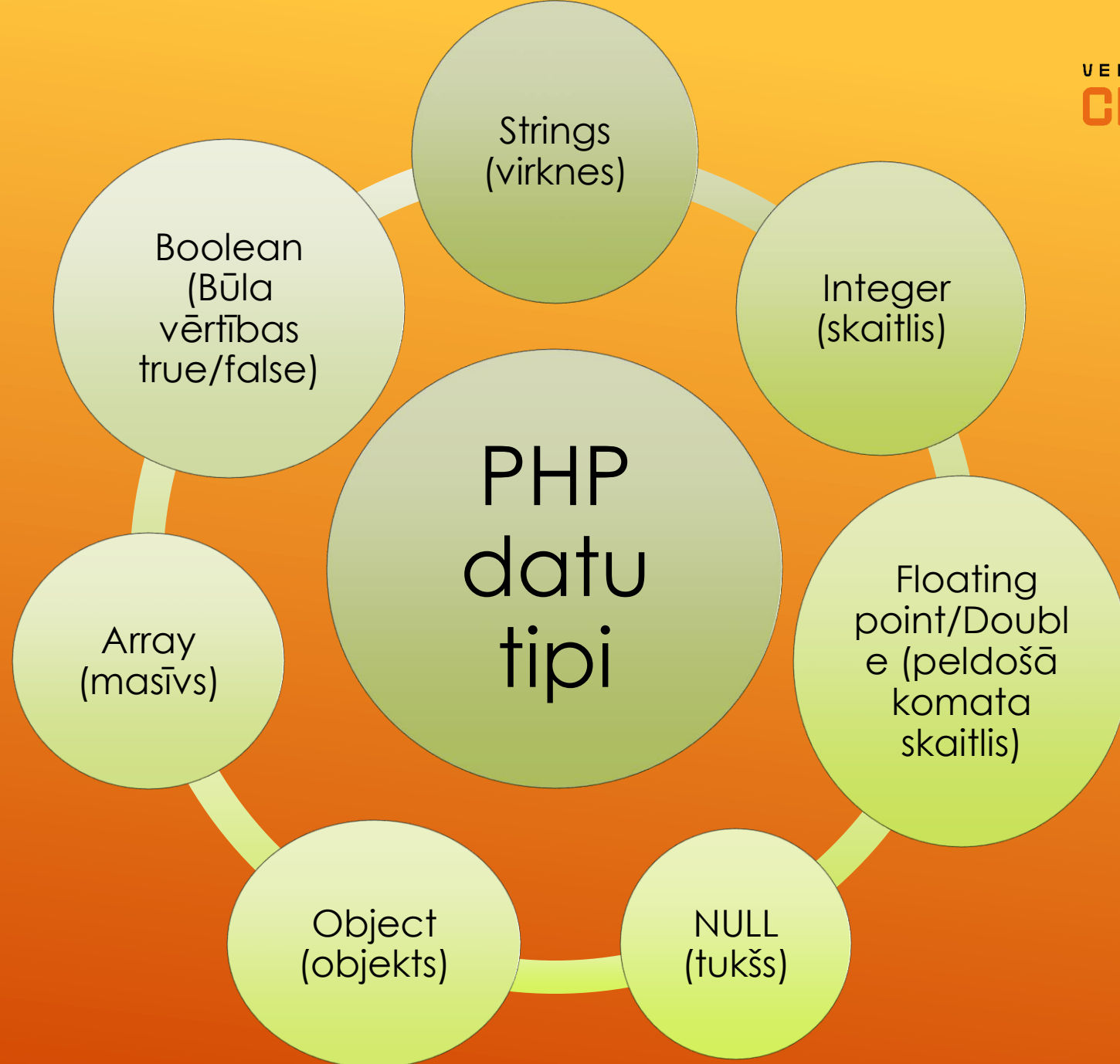
16

# MAINĪGO TIPI

- Atšķirībā no daudzām programmēšanas valodām PHP **nav jānorāda mainīgā tips!**
- PHP automātiski maina un piešķir mainīgā tipu.
- Bet ir jānorāda mainīgā datu tips!

- Mainīgie var tikt definēti jebkurā koda/sripta vietā.
- Var būt trīs veida mainīgie:

Lokāls (local);	Dokumenta ietvaros
Globāls (global);	Ārpus dokumenta
Statisks (static)	Mainīgā vērtība funkcijas iekšienē nemainās





# PHP STRINGS

- ❑ Strings ir simbolu virkne, piemēram, Sveika pasaule!
- ❑ Virkni vienmēr ir jāliek pēdiņās, tās var būt ordināras vai dubultas.

```
<?php  
$x = "Hello world!";  
$y = 'Hello world!';  
  
echo $x;  
echo "<br>";  
echo $y;  
?>
```

Atgriež  
simbolu  
virkni!

```
Hello world!  
Hello world!
```

# PHP INTEGERS

Integer ir vesels skaitlis bez decimāldaļām no -2,147,483,648 līdz 2,147,483,647.

Jābūt vismaz vienam ciparam	0-9
Nedrīkst saturēt komatu vai atstarpes	
Var būt pozitīvs vai negatīvs	
Veselos skaitļus var norādīt trīs formātos	decimāli (10 bāzes), astotnieku jeb oktālā (8 bāzes – prefiksu ar 0), heksadecimāli (16 bāzes - prefiksu ar 0x)

# PIEMĒRS

```
<?php  
$x = 5985;  
var_dump($x);  
?>
```

Atgriež  
datu tipu  
un vērtību

int(5985)

# PHP FLOAT/DOUBLE

Float ir reāli skaitļi, kas satur arī zīmes aiz komata vai skaitlis eksponenciālā formā.

```
<?php  
$x = 10.365;  
var_dump($x);  
?>
```

Atgriež  
datu tipu  
un vērtību

float(10.365)

# BŪLA ALGEBRA

Matemātikā un matemātiskajā loģikā Būla algebra ir algebras **apakšnozare**, kurā mainīgo vērtības ir patiesumvērtības *patiess un nepatiess*, ko parasti atzīmē attiecīgi ar 1 un 0. Atšķirībā no elementārās algebras, kur mainīgo vērtības ir skaitļi un galvenās operācijas ir saskaitīšana un reizināšana, Būla algebrā galvenās operācijas ir **konjunkcija** un, ko apzīmē ar  $\wedge$ , **disjunkcija** vai, ko apzīmē ar  $\vee$ , un **negācija**, ko apzīmē ar  $\neg$ .

Būla algebrai ir bijusi ārkārtīgi liela nozīme datorzinātnes un digitālās loģikas attīstībā. To izmanto arī kopu teorijā un statistikā.



un (konjunkcija), apzīmēta  $x\wedge y$  (reizēm  $x$  AND  $y$  vai  $Kxy$ ), apmierina  $x\wedge y = 1$ , ja  $x = y = 1$  un  $x\wedge y = 0$  citos gadījumos.

vai (disjunkcija), apzīmēta  $x\vee y$  (reizēm  $x$  OR  $y$  vai  $Axy$ ), apmierina  $x\vee y = 0$ , ja  $x = y = 0$  un  $x\vee y = 1$  citos gadījumos.

ne (negācija), apzīmēta  $\neg x$  (reizēm NOT  $x$ ,  $Nx$  vai  $!x$ ), apmierina  $\neg x = 0$ , ja  $x = 1$  un  $\neg x = 1$ , ja  $x = 0$ .

Ja patiesības vērtības 0 un 1 interpretē kā veselus skaitļus, šīs darbības var izteikt ar parastajiem aritmētikas darbību operatoriem:

- $x \wedge y = xy$ , AND/UN
- $x \vee y = x + y - xy$ , OR/VAI
- $\neg x = 1 - x$ . NOT/NE

ALTERNATĪVI  $x \wedge y$ ,  $x \vee y$  UN  $\neg x$   
VĒRTĪBAS VAR IZTEIKT AR PATIESUMVĒRTĪBU TABULU  
PALĪDZĪBU:

$x$	$y$	$x \wedge y$	$x \vee y$
0	0	0	0
1	0	0	1
0	1	0	1
1	1	1	1

$x$	$\neg x$
0	1
1	0

Var pieņemt, ka tikai negācija un viena no divām operācijām ir pamatoperācija, jo šīs vienādības ļauj definēt konjunktiju kā atkarību no negācijas un disjunktijas un otrādi:

$$x \wedge y = \neg(\neg x \vee \neg y)$$

$$x \vee y = \neg(\neg x \wedge \neg y)$$

# PHP BOOLEAN

Būla vērtībām PHP ir iespējamās divas vērtības:  
TRUE (paties) vai FALSE (nepaties).

```
$x = true;  
$y = false;
```

# PHP ARRAY

Masīvs saglabā vairākas vērtības vienā mainīgajā.

```
<?php
$cars =
array("Volvo","BMW","Toyota"
);
var_dump($cars);
?>
```

```
array(3) { [0]=> string(5) "Volvo" [1]=> string(3) "BMW"
           [2]=> string(6) "Toyota" }
```

# PHP OBJECT

- ❑ Objekts ir datu tips, kas uzglabā datus un informāciju par to, kā apstrādāt šos datus.
- ❑ PHP, objektam ir jābūt skaidri deklarētam.
- ❑ Vispirms jādefinē objekta klasi. Lai to izdarītu, jāizmanto **class** atslēgvārdu.
- ❑ Klase ir struktūra, kas var saturēt īpašības un metodes (properties and methods).

```
<?php
class Car {
    function Car() {
        $this->model = "VW";
    }
}
// create an object
$herbie = new Car();

// show object properties
echo $herbie->model;
?>
```

Kāds būs  
rezultāts?



# PHP NULL VĒRTĪBAS

- NULL vērtība apliecina, ka mainīgajam nav nekādas vērtības.
- NULL ir vienīga iespējama vērtība datu tipam NULL.
- NULL vērtība nosaka, ka mainīgais ir tukšs vai neeksistē vispār.

# MAINĪGO VAR IZTUKŠOT, NOSAKOT VĒRTĪBU NULL:

```
<?php  
$x = "Hello world!";  
$x = null;  
var_dump($x);  
?>
```

NULL

# Jautājumi ????