



**WEB** разработка

**Въведение в PHP**

# Съдържание

1. рнр - ?????
2. php - как работи
3. php.ini
4. хатрр - настройки
5. най-чести грешки на начинаещите
6. Как да настроим показването на грешките
7. Ресурси - php.net ...
8. Синтаксис
9. Променливи - синтаксис, подреждане на кода, коментари, имена на файловете, документиране, TO DO
10. Променливи – задаване на стойност escaping ' , ""
11. Оператори и изрази
12. php&html
13. php функции

# Ресурси



# Ресурси

W3school

Обща информация за PHP - <http://php.net/>, отчасти на български

Добри практики за писане на PHP код -

<http://www.phptherightway.com>

<https://phpbestpractices.org>

....

# PHP факти и основни понятия

Victorious warriors win first  
and then go to war  
while defeated warriors go to war  
and then seek to win.

Sun Tzu

# PHP факти и основни понятия

PHP - Hypertext Preprocessor

Скриптов език /server-side/

Създава динамично съдържание в сайт

Работи с бази данни (БД)

Трябва ни сървър

Може да се влага в HTML

Може да генерира HTML код

С отворен код - <http://php.net> всичко, свързано с PHP

Може да бъде използван и за десктоп приложения.



# PHP факти и основни понятия

## Малко предварителна информация

### HTTP - Hyper Text Transfer Protocol (HTTP)

Client-server протокол за прехвърляне на Web-ресурси, които могат да са - HTML файлове, CSS стилове, изображения, скриптове и др.

### Web Server

Web server е софтуер, който служи за предаване на Web – съдържание чрез HTTP заявки. Хоства Web – сайтове, Web – приложения и др. услуги.

/Каква е разликата м/у Web – сайтове и Web – приложения?/

Обработка HTTP заявките и връща статично или динамично Web – съдържание (HTML, CSS, изображения, JSON, JS)

Стартира **server - side скриптове като PHP**, Python и Ruby.

**Обикновено клиента - (Web browser) заявява Web - страница, Web – сървър я генерира и връща/подава на клиента.**

**CGI (Common Gateway Interface)** - [https://bg.wikipedia.org/wiki/Common\\_Gateway\\_Interface](https://bg.wikipedia.org/wiki/Common_Gateway_Interface)

CGI програмите приемат данни от сървър и обикновено връщат HTML, картинки, видео и др. съдържание, разбираемо от уеб браузърите.

**Кодът на CGI програмите не е видим - само неговия резултат.**

Предназначение – генериране на динамично Web – съдържание.

Съществуват и др. технологии, свързващи Web – сървърите със сървър-скриптове - CGI, FastCGI, Apache modules, ISAPI (IIS), WSGI, Ruby Rack, ...

# PHP факти и основни понятия

Как работи PHP?

1. .php - файлове
2. CGI

Детайли - <http://www.phpknowhow.com/basics/running-php-files/#sthash.c47MMjSV.dpuf>

За да инсталираме PHP сървър ще използваме *xampp*

Създаваме файлове с разширение **.php - example1.php**

Стартираме в браузъра файловете – написвайки **localhost/example1.php**

*!файловете, които разработваме трябва да бъдат в web папката на сървъра /htdocs по подразбиране/ за да са достъпни в браузъра!.*



# PHP факти и основни понятия

Къде се намират -

PHP файловете /проектите, върху които ще работим оттук нататък/ - **xampp/htdocs**

Apache web – сървър **xampp/apache**

PHP **xampp/php**

MySQL database **xampp/mysql**

## **XAMPP** конфигурационни файлове

Apache конфигурационен файл (**httpd.conf**) –

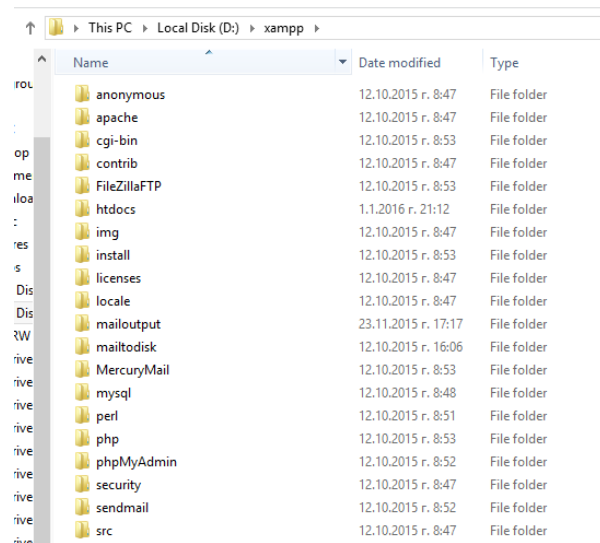
**\\xampp\\apache\\conf\\httpd.conf**

PHP конфигурационен файл (**php.ini**): -

**\\xampp\\apache\\bin\\php.ini**

MySQL конфигурационен файл (**my.ini**):-

**\\xampp\\mysql\\bin\\my.ini**



↑ This PC > Local Disk (D:) > xampp >		
Name	Date modified	Type
anonymous	12.10.2015 r. 8:47	File folder
apache	12.10.2015 r. 8:47	File folder
cgi-bin	12.10.2015 r. 8:53	File folder
contrib	12.10.2015 r. 8:47	File folder
FileZillaFTP	12.10.2015 r. 8:53	File folder
htdocs	1.1.2016 r. 21:12	File folder
img	12.10.2015 r. 8:47	File folder
install	12.10.2015 r. 8:53	File folder
licenses	12.10.2015 r. 8:47	File folder
locale	12.10.2015 r. 8:47	File folder
mailoutput	23.11.2015 r. 17:17	File folder
mailtodisk	12.10.2015 r. 16:06	File folder
MercuryMail	12.10.2015 r. 8:53	File folder
mysql	12.10.2015 r. 8:48	File folder
perl	12.10.2015 r. 8:51	File folder
php	12.10.2015 r. 8:53	File folder
phpMyAdmin	12.10.2015 r. 8:52	File folder
security	12.10.2015 r. 8:47	File folder
sendmail	12.10.2015 r. 8:52	File folder
src	12.10.2015 r. 8:47	File folder

# PHP факти и основни понятия

## PHP настройки

Съхраняват се в **php.ini**, съдържа

Определен брой **name = value** декларации.

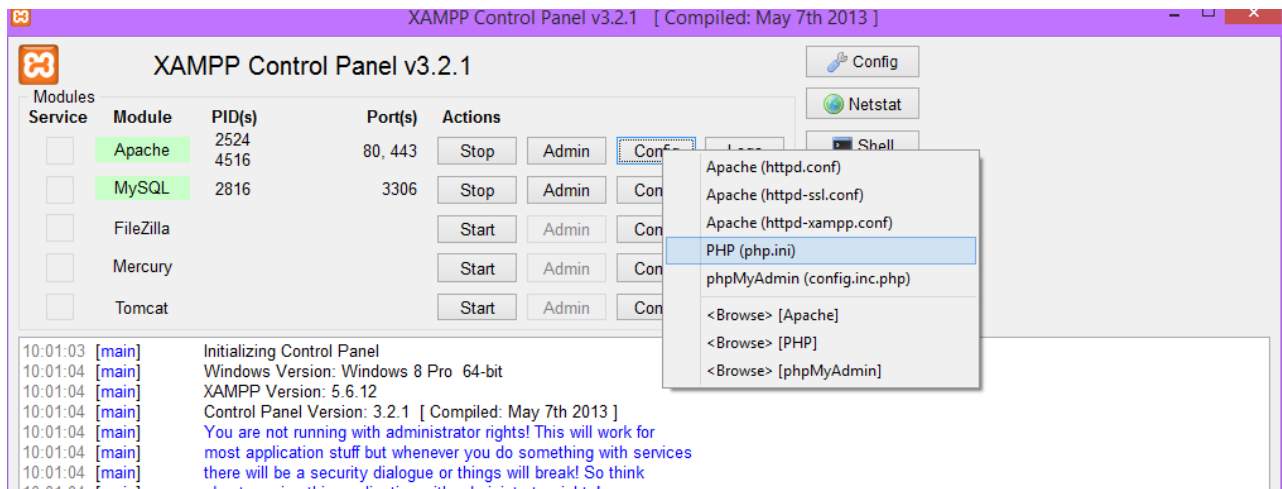
Ако няма стойност декларацията изглежда като **name =**

Местонахождението на файла е различно в зависимост от операционната система и версиите.

**!** Може да проверите кой **php.ini** е зареден с **phpinfo()**, както и текущите стойности на настройките на PHP или ->

# PHP факти и основни понятия

От контрол панела на XAMPP



# PHP факти и основни понятия

PHP Настройки да виждаме съобщенията за грешки –

`display_errors = on /php.ini/`

<http://stackoverflow.com/questions/1053424/how-do-i-get-php-errors-to-display>

# Най-често повтаряни грешки

# Най-често повтаряни грешки

## Най-чести грешки

1. Липсва ;
2. Липсва затваряща скоба
3. Липсва echo



# Най-често повтаряни грешки

PHP настройки за да са видими съобщенията за грешки – `display_errors = on` /php.ini/

Повече →

<http://stackoverflow.com/questions/1053424/how-do-i-get-php-errors-to-display>

# PHP синтаксис



# Синтаксис

PHP файлът започва с `<?php`

Всяка команда завършва с ;

```
<?php echo "Hello PHP!";
```

Вградени функции - print, echo

Всички функции, описание и примери <http://php.net/>

# Синтаксис 2

Когато файла ви съдържа само php код е препоръчително да не се поставя затварящия `?>` таг на php кода!

<http://php.net/manual/en/language.basic-syntax.phptags.php>

**променливи**

# Променливи - синтаксис

Променливи - контейнер за съхраняване на информация

1. Всички променливи в PHP започват с \$.
2. Имената им могат да съдържат – букви, цифри [0-9], долно тире '\_'
3. Не могат да започват с цифра
4. Без запазени за езика думи като – new, function ...
5. \$this – има специално място в php, не може да го използвате за име на променлива
6. Имената на променливите в PHP са case-sensitive

# Променливи – синтаксис 2

7. Името на променливата, трябва да е описателно - \$counter /явно е брояч/, \$name\_length /дължина на име/, \$date\_deleted /дата на изтриване/ и т.н.

8. Без еднобуквени променливи – \$a, \$b, .....\$aa, \$bb, .....

9. Без твърде описателни - \$the\_name\_of\_the\_game\_of\_the\_man  
- \$theNameOfTheGameOfTheMan

10. Препоръчва се имената да не започват с долно тире, тъй като има няколко важни глобални променливи, с този синтаксис например \$\_POST.

11. Съдържат само латински букви и думи на АНГЛИЙСКИ!

# Променливи – синтаксис 3

12. Променливите използват т.нар. **camelCase** или **\_** за разделяне на думите една от друга. **Без интервали в имената!**

13. **\$length&width** не е валидно име на променлива, тъй като съдържа - &

**!само латински букви и цифри!**

# Променливи - деклариране

Оператор за присвояване =

задава стойност на променливата

PHP е **typeless** език – т.е. не декларираме типа на променливата изрично. Типът ѝ може да бъде променян /но не се препоръчва!/

# Променливи – деклариране 2

Променливата има – име,  
тип (на съхраняваните данни),  
стойност

Пример:

**\$counter = 5;**

Име: **\$counter**

Тип/на данните/: **integer** /цяло число/

Value: **5**



# Променливи – деклариране 3

Когато декларираме променлива даваме ѝ уникално /за проекта/ име, (identifier)

Даваме и първоначална стойност, която ще определи типа на променливата.

Например:       <?php

```
$name = "John";
```

```
$height = 200;
```

```
$str = 'Hello';
```

```
$isPositive = true;
```

# Променливи – gettype

> Variable handling Functions

Change language:

[Edit](#)

## gettype

(PHP 4, PHP 5, PHP 7)

gettype — Get the type of a variable

### Description

от какъв тип е резултатът от функцията

параметър

```
string gettype ( mixed $var )
```

име на функцията

тип на параметъра

Returns the type of the PHP variable **var**. For type checking, use **is\_\*** functions.

описание на функцията

### Parameters

**var**

The variable being type checked.

# Оператори

# Оператори

Оператори – използваме ги за действия с променливите и техните стойности.

Видове –

Аритметични оператори

Оператори за присвояване

Оператори за сравняване

Детайли - [http://www.w3schools.com/php/php\\_operators.asp](http://www.w3schools.com/php/php_operators.asp)

Операторите имат приоритет/ред/ на действие.

Скобите винаги имат най-висок приоритет.

!Винаги използвайте скоби, дори да ви се струва излишно!

# Оператори 2

Чрез операторите построяваме изрази.

Най-простия пример за израз е

```
$a = 5 //присвояване стойност на променлива
```

Можем да го усложним –

```
$b = 2*$a; ... всеки математически израз можем да изразим с код,
```

Например да намерим изминатия път –

```
$travel = 100 //изминатия път
```

```
$time = 10// времето, за което е изминат
```

```
$velocity = $travel/$time //намираме скоростта
```

Възможно е да се опишат и доста по-сложни формули – изрази.

# Оператори 3

Конкатениране или “събиране на стрингове” - **.** /точката/

```
$firstName = "Ivan";  
$lastName = "Petrov";  
$name = $firstName." ".$lastName; //обърнете внимание на добавянето на празния стринг  
echo $name; // Ivan Petrov
```

```
$age = 33;
```

```
echo "Age - ".$age." years"; //Age – 33 years
```

//Конкатениране използваме и за отпечатване на текст и стойности на променливи.

# Оператори 4

Кавички – единични ' и двойни "" ?!

Използваме ги за ограждане на стринговете.

```
$str = "Some string";
```

```
$strAnother = 'Another string';
```

! ' ще отпечата **променливата** като стринг, а не нейната стойност ->

```
echo 'My string is $str.'; // My string is $str.
```

```
$number = 1025;
```

```
echo 'My favorite number is $number.'; //My favorite number is $number.
```

! "" ще отпечата **стойността на променливата** ->

```
echo "My string is $str."; //My string is Some string.
```

```
echo "My favorite number is $number."; //My favorite number is 1025.
```

# Оператори 4

!! Важно !! Когато в текста трябва да се отпечата кавички  
echo "Name"//Name';

echo "Name"//"Name";

\$name = "Ivan";  
echo "Name" - \$name;// 'Name' – Ivan /!!!/

Или

Echo "Name" - ".\$name;// 'Name' – Ivan



# PHP & HTML

# PHP & HTML

```
<?php
$name = "My Name";
echo "<div>".$name."</div>";
$name = 'Your Name';
$id = "first";
echo "<div id=$id>".$name."</div>";
$name = 'His Name';
$id = "second";
echo "<div id=$id>".$name."</div>";
?>
```

Ако **вградим** този код в HTML можем да използваме променливата \$id и задавайки различни стойности можем да задаваме различни стилове на елементите.

My Name

Your Name

His Name