

# сесии

**PHP web development 2020/2021**

Milena Tomova  
Vratsa Software

<https://vratsasoftware.com/>

1. PHP сесии
2. Стартиране на сесия
3. Запазване на информация в сесията
4. Унищожаване на сесията и променливите, записани в нея
5. Сесии и регистрирани потребители



PHP сесии



- Създайте игра - викторина.
- Потребителят ви започва играта, като въвежда своето име.
- Отговаря на серия въпроси като избира от няколко отговора и натиска бутон ЗАПАЗИ ОТГОВОР. Един от тези отговори е верния.
- Всеки въпрос е разположен на отделна страница - преминаването между въпросите и съответно страниците става след натискане на бутона ЗАПАЗИ ОТГОВОР.
- За всеки правилен отговор, потребителят получава 1 точка.
- Във всяка страница се вижда името на потребителя, резултатът му към момента и бутон прекъсни игратата.
- След отговор на последния въпрос или натискане на бутона за прекъсване на играта потребителят преминава на страница, на която вижда своето име, резултата си към момента и бутон ИЗЛЕЗ
- бутонът ИЗЛЕЗ изтрива, запазената до момента информация и връща потребителя към началната страница на играта.

# PHP сесия

сесия

- Всяка **сесия** разполага с уникален идентификатор - **session ID** и може да бъде използвана за съхраняване на информация. Тази информация може да бъде достъпвана и използвана.
- Информацията, асоциирана с текущата сесия е достъпна чрез суперглобалната PHP променлива

**`$_SESSION`**

сесия

Асоциираните с текущата сесия данни /**session state**/ се запазват обикновено във временен файл.

Може да се използва база данни или друг метод, чрез функцията извикване на функцията

**session\_set\_save\_handler()**



The background of the slide features a light gray, stylized illustration of a mountain range with sharp peaks and a dense forest of evergreen trees in the foreground. A large, solid dark blue circle is centered on the slide, containing the text.

**стартиране на сесия**

- За да работите със сесия трябва **изрично да я стартирате** или да промените настройките на сесиите в **php.ini**.  
**По подразбиране сесиите не започват автоматично.**
- За да започват автоматично, трябва да промените следните настройки в **php.ini** -

от `session.auto_start = 0`

на `session.auto_start = 1`



# Стартиране на сесия

При **session.auto\_start = 1**

Сесията стартира за всеки документ зареден от уеб сървъра

При **session.auto\_start = 0**

в началото на всеки скрипт трябва да поставите

**session\_start()**

за да стартира сесията за текущия документ.

# Стартиране на сесия

След стартиране на сесията може да достъпите нейното ID чрез `session_id()`

Пример Стартиране на сесия.

```
<?php
session_start();

echo "<p>Your session ID is ".session_id()."</p>";
```

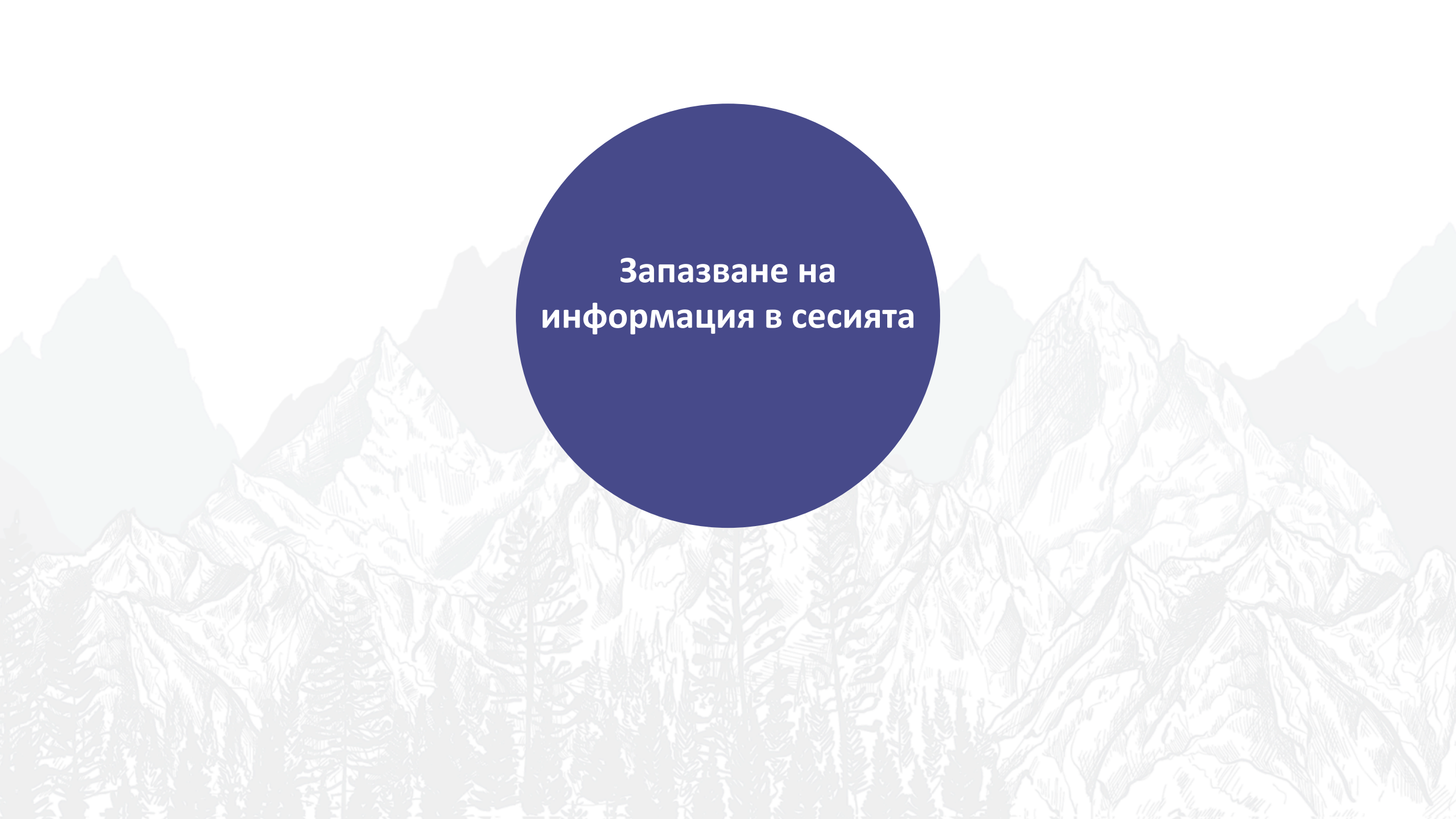


По **подразбиране** сесията продължава докато се затвори браузъра.

Продължителността на сесията може да се промени в **php.ini**

Конфигуриране на сесията в php.ini - метод по подразбиране





## Запазване на информация в сесията

# Запазване на данни в сесията

- След стартиране на сесия, може да запазите произволен брой променливи в **\$\_SESSION** и да ги достъпвате в страници, в които също е стартирана сесия

*Пример* Добавяне на две променливи към \$\_SESSION \$product1 и \$product2

```
<?php
    session_start();

    $product1 = "Sonic Screwdriver";
    $product2 = "HAL 2000";
    $_SESSION['product1'] = $product1;
    $_SESSION['product2'] = $product2;
    echo "The products have been registered.";
```



# Запазване на данни в сесията

- При преминаване в друг файл, в който е стартирана сесия, данните са достъпна в `$_SESSION` масива.

*Пример* Достъпване на данни, записани в `$_SESSION`

```
<?php
    session_start();

    echo $_SESSION['product1'];
    echo $_SESSION['product2'];
```



# Запазване на данни в сесията

- Какво се случва в действителност?
- Зад кадър данните се записват във временен файл на сървъра.
- Може да откриете този файл чрез извикване на функцията


`session_save_path()`

# Запазване на данни в сесията

- След като добавите данни към `$_SESSION`, може да ги промените в последствие и да използвате променената стойност.
- Ако искате да запазите променената стойност в сесията, трябва да презапишете стойността на същия елемент в глобалния масив `$_SESSION` /същия индекс/

```
<?php
$a = 10;
$_SESSION['a'] = $a;
echo $_SESSION['a']; //10
$a = 20;
echo $_SESSION['a']; //10
$_SESSION['a'] = $a;
echo $_SESSION['a']; //20
```





**Унищожаване на  
сесията и  
променливите,  
записани в нея**



# Унищожаване на сесията и ....

За да прекратите сесия използвайте функцията

**session\_destroy()**

Тя ще изтрие всички променливи, запазени на сървъра.

В скрипта се използва в следната последователност -

```
session_start();  
    //стартираме сесия  
session_destroy();  
    //унищожаваме сесията
```

# Унищожаване на сесия и ...

След унищожаване на сесията, преминаването към страница, **стартираща сесия** - ще предизвика **стартиране на нова сесия**.

Всички данни от предходната сесия вече не са достъпни, започва запаметяването на нови данни - тези на новостартираната сесия.





# Унищожаване на сесия и ...

Извикването на **session\_destroy()** не унищожават веднага всички, регистрирани в сесията променливи.

Те все още са достъпни в скрипт /файла/, в който е извикана **session\_destroy()**, **докато не презаредите страницата!**

## Пример

```
<?php
    session_start();
    $_SESSION['test'] = 5;

    session_destroy();
    echo $_SESSION['test']; // 5
```






# Унищожаване на сесията и ....

Използвайте функцията **unset()** ако сценария, под който работи вашето приложение, изисква унищожаване на запаметени в сесията данни /елементи в масива `$_SESSION/`.

```
<?php
session_start();

$_SESSION['test'] = 5;

unset($_SESSION['test']);
echo $_SESSION['test']; // prints nothing (or a notice about an undefined index)
```



**Унищожаване на  
сесията и  
променливите,  
записани в нея**



# Сесии и регистрирани потребители

Да предположим, че сте изградили онлайн платформа - блог, портал или друго приложение, в което потребителите могат да се присъединяват като се регистрират, например с **потребителско име и парола**.

След регистриране, при всяко влизане на потребителя може да запазите потребителското му име /и друга информация/ в сесията.

Данните за потребителя, които решите да запазите в сесията, трябва да бъдат, тези, които смятате, че ще използвате доста често.

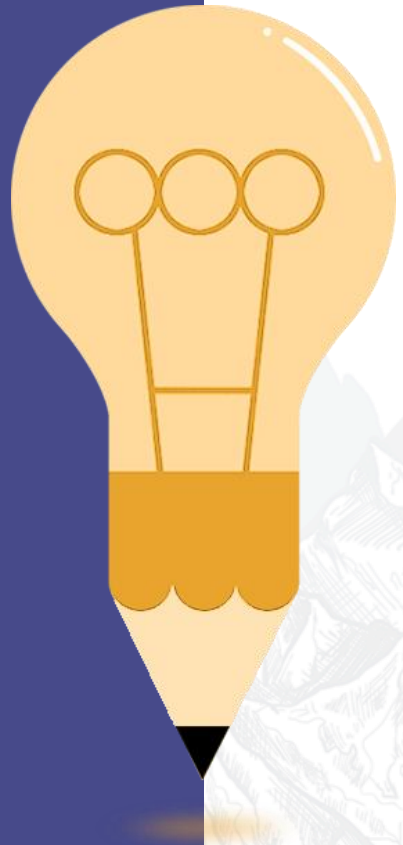
Например - ако сте разработили портал, в който всеки потребител има определени **права за достъп** - администратор, регистриран потребител, анонимен гост и т. н.

В отделните модули на приложението, ще проверявате правата на потребителя, за да сте сигурни, че той има право да достъпва съответния модул.

**Правата на достъп на потребителя**, са пример за стойност, която бихте запазили в сесията на съответния потребител, така че скрипта проверяващ дали може да визуализира даден модул по заявка на потребителя да проверява само стойността в сесията, без да е необходимо да се свързва, селектира и търси в базата данни.



# Обобщение



- Сесиите ни дават възможност **да съхраним състояние**, при използване на HTTP заявки /които са stateless - не могат да запамятват състояние - стойности на променливи/.
- Един от вариантите за съхраняване на данните от сесията е да се записват във **временен файл**, който се съхранява на сървъра.
- **Стартиране/Възстановяване/Продължаване** сесия с функцията **session\_start()**
- При стартиране на сесия, запазваме данни в сесията като елементи на масива **\$\_SESSION** /**добавяме уникални индекси**/
- Данните записани в \$\_SESSION се **унищожават с unset()** или при унищожаването на самата сесия със **session\_destroy()**.
  - Във втория случай променливите са **достъпни до края на скрипта** или **докато се презареди страницата**.

# Learn to Search in Internet

- The course assignments require to search in Internet
  - This is an important part of the learning process
  - Some exercises intentionally have no hints
- Learn to find solutions!
  - Software development includes everyday searching and learning
  - No excuses, just learn to study!
  - Developers learn new technologies, tools, languages every day!





# Questions?



Гнездото  
Coworking

Цялостен  
курс по  
програми  
ране

Дизайн  
курс

Курс по  
дигит.  
маркетинг

MindHub





# Partners



**Telerik  
Academy**



**MindHub**

**ПРОМЯНАТА**

# Trainings @ Vratsa Software



- Vratsa Software – High-Quality Education, Profession and Jobs
  - [www.vratsasoftware.com](http://www.vratsasoftware.com)
- The Nest Coworking
  - [www.nest.bg](http://www.nest.bg)
- Vratsa Software @ Facebook
  - [www.fb.com/VratsaSoftware](http://www.fb.com/VratsaSoftware)
- Slack Channel
  - [www.vso.slack.com](http://www.vso.slack.com)

