

# Sessões em PHP

# O que é uma sessão

- Suporte a sessões no PHP:
  - maneira de preservar dados através de acessos subsequentes.
- Características:
  - permite a criação de aplicações mais personalizadas;
  - permite que certas características do site sejam apresentadas somente aos usuários pertencentes a determinada sessão.
- É determinada através de uma variável auto-global chamada “variável de sessão PHP”.

# Variável de sessão PHP

- É utilizada para
  - armazenar informações sobre a sessão;
  - Modificar configurações da sessão de um usuário.
- Armazenam informação sobre um único usuário;
- Permanecem disponíveis para todas as páginas de um aplicação web.
- Elas trabalham criando uma única identificação (UID) para cada usuário e armazenam variáveis utilizando essa UID.

# Sessões

- Permitem trabalhar como em desktops:
  - Abrir um aplicativo associado a um usuário;
  - Utilizar o aplicativo;
  - Fechar o aplicativo.
- As sessões permitem armazenar informações sobre o usuário no servidor para utilizar depois.
- Porém, são temporárias, ou seja, terminam assim que o site é abandonado.

# Criando uma sessão

```
<?php
session_start();

if( isset( $_SESSION['views'] ) )
    $_SESSION['views']++;
else
    $_SESSION['views']=1;

echo "Session ID=".session_id()."<br/>";
echo "Visualizacoes=".$_SESSION['views'];
?>
```

# Destruindo uma sessão

- Apagando uma variável de sessão:

```
<?php
session_start();
if( isset( $_SESSION[ 'views' ] ) )
    unset( $_SESSION[ 'views' ] );
?>
```

- Destruindo toda a sessão:

```
<?php
session_destroy();
?>
```

# Cookies

# Cookies

- São formas utilizadas por website para identificar um usuário.
- Com isso, consegue salvar informações específicas para que este usuário seja tratado de uma forma diferente na próxima vez que entrar no site:
  - Cor de fundo do site;
  - Pesquisas realizadas;
  - Nick utilizado no chat;
  - etc.
- Assim, na próxima vez que entrar no site, ele abrirá essas variáveis e saberá como tratar sua visita.
  - Geralmente os sites mantêm um campo chamado “Lembre-se de mim” ou “Remember Me” para isso.



# Cuidados com cookies

- Suponha que um cookie possua as seguintes variáveis:
  - Usuario;
  - Nivel\_Acesso;
- Se o usuário modificar o valor dessas variáveis e entrar no site de novo, ele pode se tornar um administrador.

# Criando um cookie

```
setcookie( name, value, expiration, path, domain, secure, httponly );
```

- Name: nome que seu cookie terá no computador do usuário;
- Value: valor que será armazenado no cookie;
- Expiration: define quando o cookie vai expirar (ser apagado);
- Path: O caminho no servidor aonde o cookie estará disponível;
- Domain: O domínio para qual o cookie estará disponível;
- Secure: Indica que o cookie só poderá ser transmitido sob uma conexão segura HTTPS do cliente;
- Httponly: Quando for TRUE o cookie será acessível somente sob o protocolo HTTP.

# Exemplos

```
<?php
```

```
    $value = 'alguma coisa de algum lugar';
```

```
    setcookie("CookieTeste", $value);
```

```
    /* expira em 1 hora */
```

```
    setcookie("CookieTeste", $value, time()+3600);
```

```
    setcookie("CookieTeste", $value, time()+3600,  
              "~/rasmus/", ".example.com", 1);
```

```
?>
```

# Obtendo os valores do cookie

- Os valores são obtidos automaticamente.
- Para acessá-los, basta utilizar a variável `$_COOKIE`:

```
<?php
```

```
// Mostra um cookie individual
```

```
echo $_COOKIE[ "CookieTeste" ];
```

```
echo $HTTP_COOKIE_VARS["CookieTeste"];
```

```
// Outra maneira de testar é vendo todos os cookies
```

```
print_r( $_COOKIE );
```

```
?>
```

# Cookies e Arrays

```
<?php
// envia os cookies
setcookie("cookie[tres]", "cookie tres");
setcookie("cookie[dois]", "cookie dois");
setcookie("cookie[um]", "cookie um");

// Depois que a página recarregar, mostra eles

if ( isset( $_COOKIE["cookie"] ) ) {
    foreach ( $_COOKIE["cookie"] as $nome => $valor ) {
        echo "$nome : $valor <br />\n";
    }
}
?>
```

Qual utilizar: sessão ou cookies?

Qual utilizar: sessão ou cookies?

Os cookies são mais fáceis de serem modificados do que as sessões.

# Exemplo de uma função login

```
function login($username, $password) {  
    $sql = mysql_query(  
        "SELECT id, user_level FROM users WHERE password = '" . $password .  
        "' AND username = '" . $username . "' LIMIT 1");  
  
    if($sql === false) return false;  
    else { $resp = mysql_fetch_array( $sql );  
        session_regenerate_id(true);  
        $session_id = $resp[ id ];  
        $session_username = $username;  
        $session_level = $resp[ user_level ];  
  
        $_SESSION['user_id'] = $session_id;  
        $_SESSION['user_level'] = $session_level;  
        $_SESSION['user_name'] = $session_username;  
        $_SESSION['user_lastactive'] = time();  
        return true;    }  
}
```



# Se for utilizar o “Lembre-se de mim”

```
function login($username, $password, $remember = false)
...
if($remember) {
    // Gera uma nova chave de identificação
    $cookie_auth = rand_string(10) . $username;
    $auth_key = session_encrypt( $cookie_auth );
    $auth_query = mysql_query(
        "UPDATE users SET auth_key = '" . $auth_key .
        "' WHERE username = '" . $username . "'" );
    exp = time()+60*60*24*7; //sete dias após o dia atual
    setcookie( "auth_key", $auth_key, exp, "/",
        "exemplo.com", false, true );
}
```

Mais exemplos no terminal