

## I. HTML

### 1. Struktura dokumentu HTML:

```
<html>
<head>
    nagłówek strony - informacje o stronie
</head>
<body>
    główna treść strony - tekst, obrazki itd
</body>
</html>
```

### 2. Link tekstowy:

```
<a href="index.php">treść linka</a>
<a href="index.php#nazwa_etykiety">treść linka</a> ---> <a name="nazwa_etykiety"></a>
<a href="http://www.jakisadres.pl">treść linka</a>
```

### 3. Przycisk jako link:

```
<button onclick="window.location.href='/index.php'">link</button>
<a href="index.php"><button>link</button></a>
```

### 4. Obraz jako link:

```
<a href="index.php"></a>
```

### 5. Tabela:

1
2 3

```
<table>
  <tr>
    <td colspan="2">1</td>
  </tr>
  <tr>
    <td>2</td>
    <td>3</td>
  </tr>
</table>
```

<tr> - wiersz

<td> - komórka

colspan - łączenie kolumn

rowspan - łączenie wierszy

### 6. Formularze:

```
<form action="index.php" method="post">
  Login: <input type="text" name="login"><br>
  Hasło: <input type="password" name="password"><br>
  <input type="submit" value="ZALOGUJ">
</form>
```

action - zdefiniowanie strony która zostanie załadowana po naciśnięciu przycisku "ZALOGUJ"

method - sposób wysyłania danych: GET - jawnie w URL'u, POST - niejawnie

input - pole wprowadzania typu pola

name - określenie nazwy indexu zmiennej pod jaką będą się kryć wartości pola w PHP:

\$ GET["login"]

\$ POST["login"]

### 7. Przekazywanie zmiennych z wartościami w linku tekstowym:

```
<a href="index.php?login=abc&password=haslo&a=1&b=2">treść linka</a>
```

## II. PHP

1. Wymagania: SERWER WWW + moduł PHP np. APACHE + PHP (XAMPP - zestaw serwerów)

2. Skrypty PHP wykonywane są po stronie serwera.

Ścieżka do katalogu strony głównej na serwerze XAMPP:

**C:\xampp\htdocs (<-- TO NIE JEST ŚCIEŻKA PULPITU WINDOWS)**  
**TU UMIESZCZAMY SKRYPTY PHP np. index.php, mojastrona.php**  
**mojadluganazwabezspacjiwielkichliter.php**

2. **Tagi** w których umieszcza się skrypty PHP:

<?php ----kod php---- ?> - na egzaminie tylko ta forma

<? ----kod php---- ?> - skrócona notacja standardowo wyłączona (w php.ini ---> short\_open\_tag=On)

### 3. **Komentarze:**

//wiersz wyłączony

#wiersz wyłączony

/\*

wyłączona sekcja

\*/

### 4. **Instrukcja warunkowa if:**

if(\$a>\$b) echo("\$a jest większe od \$b");

if(\$a>\$b) echo("\$a jest większe od \$b");

else echo("\$a jest mniejsze lub równe \$b");

if(\$a>\$b)

{

echo("\$a jest większe od \$b");

echo("123");

}

else

{

echo("\$a jest mniejsze lub równe \$b");

echo("321");

}

Złożony warunek

if(\$a>\$b||\$a<2)

echo("\$a jest większe od \$b, lub mniejsze od 2");

|| - alternatywa (lub)

&& - koniunkcja (i)

! - negacja (zaprzeczenie)

Typy porównań:

\$a==\$b - spełnione gdy \$a i \$b są równe. Należy pamiętać że \$a=\$b jest operacją przypisanie - zmiennej \$a jest przypisywana wartość \$b, i całe wyrażenie ma wartość \$b.

\$a>\$b - spełnione gdy \$a jest większe od \$b

\$a>=\$b - spełnione gdy \$a jest większe lub równe \$b

\$a<\$b - spełnione gdy \$a jest mniejsze od \$b

\$a<=\$b - spełnione gdy \$a jest mniejsze lub równe \$b

isset(), empty() and is\_null()

if(isset(\$ GET["x"])) \$x=\$ GET["x"]; - czy jest ustawiona zmienna i NOT NULL

### 5. **Pętla WHILE:**

\$a=0; //nadanie wartości początkowej

while(\$a<5) //pętla wykonuje się dopóki warunek jest spełniony

{

echo("\$a"); //wyświetlenie zawartości zmiennej \$a

\$a++; //inkrementacja zmiennej

}

## 6. Pętla for:

for(\$a=0;\$a<5;\$a++) //for (inicjalizacja zmiennych ; warunek ; modyfikacja zmiennych)

```
{  
    echo("$a");  
}
```

## III. Połączenie z bazą danych MySQL

### 1. Mechanizmy:

- MySQLi extension (litera "i" jest tu bardzo ważna) - zorientowane proceduralnie i obiektowo
- PDO (PHP Data Objects) - zorientowane obiektowo

Driver name	Supported databases
<a href="#">PDO_CUBRID</a>	Cubrid
<a href="#">PDO_DBLIB</a>	FreeTDS / Microsoft SQL Server / Sybase
<a href="#">PDO_FIREBIRD</a>	Firebird
<a href="#">PDO_IBM</a>	IBM DB2
<a href="#">PDO_INFORMIX</a>	IBM Informix Dynamic Server
<a href="#">PDO_MYSQL</a>	MySQL 3.x/4.x/5.x
<a href="#">PDO_OCI</a>	Oracle Call Interface
<a href="#">PDO_ODBC</a>	ODBC v3 (IBM DB2, unixODBC and win32 ODBC)
<a href="#">PDO_PGSQL</a>	PostgreSQL
<a href="#">PDO_SQLITE</a>	SQLite 3 and SQLite 2
<a href="#">PDO_SQLSRV</a>	Microsoft SQL Server / SQL Azure

<https://www.php.net/manual/en/pdo.drivers.php>

### 2a. MySQLi proceduralnie (a).

Połączenie:

```
$conn = mysqli_connect(adres serwera, login, hasło,nazwa bazy danych);
```

Rozłączenie:

```
mysqli_close($conn);
```

### 2b. MySQLi obiektowo (b).

Połączenie:

```
$conn = new mysqli(adres serwera, login, hasło,nazwa bazy danych);
```

Rozłączenie:

```
$conn->close();
```

### 2c. PDO (c).

Połączenie:

```
$conn = new PDO("mysql:host=adres serwera;dbname=nazwa bazy danych", login, hasło);
```

Rozłączenie:

```
$conn = null;
```

### 3a. Pobranie danych z BD i wyświetlenie w tabeli.

```
$sql = "SELECT * FROM a";
```

```
$result = mysqli_query($conn, $sql);
```

```
if (mysqli_num_rows($result) > 0) {  
    echo"<table border=1>";  
    echo"<tr><td>id</td><td>nazwa</td></tr>";  
    while($row = mysqli_fetch_assoc($result))  
    {  
        echo"<tr>";  
        echo "<td>".$row["id"]."</td>";  
        echo "<td>".$row["nazwa"]."</td>";  
        echo"</tr>";  
    }  
    echo"</table>";  
} else {  
    echo "0 results";  
}
```

### 3b. Pobranie danych z BD i wyświetlenie w tabeli.

```
$sql = "SELECT * FROM a";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
    echo"<table border=1>";
    echo"<tr><td>id</td><td>nazwa</td></tr>";
    while($row = $result->fetch_assoc())
    {
        echo"<tr>";
        echo "<td>".$row["id"]."</td>";
        echo "<td>".$row["nazwa"]."</td>";
        echo"</tr>";
    }
    echo"</table>";
} else {
    echo "0 results";
}
```

### 3c. Pobranie danych z BD i wyświetlenie w tabeli.

```
$sql = "SELECT * FROM a";
echo"<table border=1>";
echo"<tr><td>id</td><td>nazwa</td></tr>";
foreach ($conn->query($sql) as $row)
{
    echo"<tr>";
    echo "<td>".$row["id"]."</td>";
    echo "<td>".$row["nazwa"]."</td>";
    echo"</tr>";
}
echo"</table>";

lub

$wykonaj=$db->query("SELECT * FROM magazyn ORDER BY nazwa");
if($wykonaj -> rowCount() > 0)
{
    while($wiersz = $wykonaj->fetch(PDO::FETCH_ASSOC))
    {
        ?>
        <b>kod html <?php echo $wiersz['id'] ?></b>
        <?
    }
}
```

### 4a. Formularz i dodawanie rekordu.

```
if(isset($_POST["nazwa"])&&isset($_POST["nazwa2"]))
{
    $sql = "INSERT INTO a (nazwa, nazwa2) VALUES ('".$_POST["nazwa"]."', '".$_POST["nazwa2"]."')";
    mysqli_query($conn, $sql);
}

<FORM ACTION=07.php METHOD=POST>
Nazwa: <INPUT TYPE=TEXT NAME=nazwa><BR>
Nazwa2: <INPUT TYPE=TEXT NAME=nazwa2><BR>
<INPUT TYPE=SUBMIT value=\"ZAPISZ\">
</FORM>
```

### 5a. Przesyłanie wartości zmiennej w linku tekstowym przez URL.

```
if(!empty($_GET["del"]))
{
    $sql = "DELETE FROM a WHERE id=".$_GET["del"];
    mysqli_query($conn, $sql);
}

<a href=\"08.php?del=".$_row["id"]."\"><button>USUN</button></a>
```

**IV. Funkcje szyfrujące.**

Name	Description
<a href="#">AES_DECRYPT()</a>	Decrypt using AES
<a href="#">AES_ENCRYPT()</a>	Encrypt using AES
<a href="#">DECODE()</a> Decodes a string encrypted using ENCODE()	<a href="#">DECODE()</a> Decodes a string encrypted using ENCODE()
<a href="#">ENCODE()</a> Encode a string	<a href="#">ENCODE()</a> Encode a string
<a href="#">ENCRYPT()</a> Encrypt a string	<a href="#">ENCRYPT()</a> Encrypt a string