Materiał dla zdających egzamin EE09

Jak zdać egzamin?

EGZAMIN POTWIERDZAJĄCY KWALIFIKACJE W ZAWODZIE

TECHNIK INFORMATYK

Nazwa kwalifikacji: **Tworzenie aplikacji internetowych i baz danych oraz administrowanie bazami**

Opracował: Piotr Lata

SPIS TREŚCI

1.Zaczynamy egzamin

Od czego zaczynamy - otwieramy arkusz 😊

KROK 1

Koniecznie czytamy **Instrukcję dla zdającego:**

1. Na pierwszej stronie arkusza egzaminacyjnego wpisz w oznaczonym miejscu swój numer PESEL i naklej naklejkę z numerem PESEL i z kodem ośrodka.
2. Na KARCIE OCENY w oznaczonym miejscu przyklej naklejkę z numerem PESEL oraz wpisz:
   a) swój numer PESEL*,
   b) oznaczenie kwalifikacji,
   c) numer zadania,
   d) numer stanowiska.
3. Sprawdź, czy arkusz egzaminacyjny zawiera 5 stron i nie zawiera błędów. Ewentualny brak stron lub inne usterki zgłoś przez podniesienie ręki przewodniczącemu zespołu nadzorującego.
4. Zapoznaj się z treścią zadania oraz stanowiskiem egzaminacyjnym. Masz na to 10 minut. **Czas ten nie jest wliczany do czasu trwania egzaminu.**
5. Czas rozpoczęcia i zakończenia pracy zapisze w widocznym miejscu przewodniczący zespołu nadzorującego.
6. Wykonaj samodzielnie zadanie egzaminacyjne. Przestrzegaj zasad bezpieczeństwa i organizacji pracy.
7. Po zakończeniu wykonania zadania pozostaw arkusz egzaminacyjny z rezultatami oraz KARTĘ OCENY na swoim stanowisku lub w miejscu wskazanym przez przewodniczącego zespołu nadzorującego.
8. Po uzyskaniu zgody zespołu nadzorującego możesz opuścić salę/miejsce przeprowadzania egzaminu.

** w przypadku braku numeru PESEL – seria i numer paszportu lub innego dokumentu potwierdzającego tożsamość*

KROK 2

Zadanie egzaminacyjne
Wykonaj aplikację internetową portalu (stronę internetową), wykorzystując pakiet XAMPP oraz edytor (np. Brackets) zaznaczający składnię.
Aby wykonać zadanie, zaloguj się na konto Egzamin bez hasła. Na pulpicie znajdziesz archiwum ZIP o nazwie ……..zip zabezpieczone hasłem: (tu będzie podane hasło)
Archiwum należy rozpakować. (**konieczne programem 7zip**)
Na pulpicie konta Egzamin utwórz folder. Jako nazwy folderu użyj swojego numeru PESEL. Rozpakowane pliki umieść w tym folderze. Wyniki swojej pracy również zapisz w tym folderze. **(pamiętaj aby podczas pracy umieścić ten folder na dysku c: w katalogu xampp – podkatalogu htdocs – na koniec pracy skopiuj go na pulpit)**

KROK 3

Operacje na bazie danych

Archiwum zip przeznaczone do naszego arkusza zawiera plik z bazą danych przeznaczony do importu.

Uruchom usługi MySQL i Apache za pomocą XAMPP Control Panel. Za pomocą narzędzia phpMyAdmin wykonaj podane operacje na bazie danych:
a) Utwórz bazę danych o podanej nazwie
b) Do bazy *(podanej w punkcie a)* zaimportuj tabele z pliku *nazwa.sql* z rozpakowanego archiwum
c) Wykonaj zrzut ekranu po imporcie. Zrzut zapisz w folderze z numerem PESEL, w formacie PNG i nazwij *import (lub nazwie podanej w arkuszu)*. Nie kadruj zrzutu. Powinien on obejmować **cały ekran monitora**, z widocznym paskiem zadań. Na zrzucie powinny być widoczne elementy wskazujące na poprawnie wykonany import tabel
d) Zapisz i wykonaj podane niżej zapytania SQL działające na bazie *(nazwa bazy)*. Zapytania zapisz w pliku ***kwerendy.txt***, w folderze z numerem PESEL. Wykonaj zrzuty ekranu przedstawiające wyniki działania kwerend. Zrzuty zapisz w formacie JPG i nadaj im nazwy *kw1*, *kw2*, *kw3*, *kw4 (lub inne nazwy podane w arkuszu)*. Zrzuty powinny obejmować cały ekran monitora z widocznym paskiem zadań

Przykładowe zapytania SQL:

a) Dodawanie rekordów do bazy:

INSERT INTO *table_name* (*column1*, *column2*, *column3*, ...)
VALUES (*value1*, *value2*, *value3*, ...);

Legenda:
table_name – nazwa tabeli w bazie danych
(column1, column2, column3, ...) – nazwy kolumn w bazie danych
(value1, value2, value3, ...); - wartości do wpisania do ww. kolumn

Przykład z innych arkuszy:

INSERT INTO zgloszenia
(ratownicy_id,dyspozytorzy_id,adres,pilne,czas_zgloszenia) VALUES
(1,4,'Warszawa, Staszica 34/5',1,'11:13:00');

Lub

INSERT INTO filmy VALUES (NULL,5,9,'Suburbicon',2017,5);

b) Wyświetlanie rekordów:

Instrukcja SELECT służy do wybierania danych z bazy danych.

SELECT *column1*, *column2, ...*
FROM *table_name*;

Jeśli chcesz zaznaczyć wszystkie pola dostępne w tabeli, użyj następującej składni:

SELECT * FROM *table_name*;

Klauzula SQL WHERE

Klauzula WHERE służy do filtrowania rekordów.

Służy do wyodrębniania tylko tych rekordów, które spełniają określony warunek.

SELECT *column1*, *column2, ...*
FROM *table_name*
WHERE *condition*;

Przykład:
SELECT adres,dyspozytorzy_id FROM zgloszenia WHERE ratownicy_id=3;

SELECT tytul,rok FROM filmy WHERE gatunki_id=3 AND ocena=5;

Klauzula JOIN służy do łączenia wierszy z co najmniej dwóch tabel na podstawie powiązanej między nimi kolumny.

Przykład:
SELECT filmy.tytul,gatunki.nazwa FROM filmy JOIN gatunki ON filmy.gatunki_id=gatunki.id;

c) Instrukcja CREATE TABLE służy do tworzenia nowej tabeli w bazie danych.

```
CREATE TABLE table_name (
   column1 datatype,
   column2 datatype,
   column3 datatype,
  ....
);
```

Przykład:

```
CREATE TABLE Persons (
   PersonID int,
   LastName varchar(255),
   FirstName varchar(255),
   Address varchar(255),
   City varchar(255)
);
```

CREATE TABLE aktorzy (id int NOT NULL,imie text, nazwisko text);

d) Oświadczenie GRANT

Nadaniu użytkownikowi przywileju (praw) po raz pierwszy, tworzony jest kolejny grant, który wygląda tak:

*GRANT USAGE on *.* TO user IDENTIFIED BY PASSWORD password*

USAGE to sposób na poinformowanie MySQL, że konto istnieje bez nadawania mu żadnych prawdziwych uprawnień. Mają jedynie uprawnienia do korzystania z serwera MySQL, stąd USAGE. Odpowiada wierszowi w `mysql`.`user` tabeli bez ustawionych uprawnień.

IDENTIFIED BY Klauzula wskazuje, że ustawione jest hasło dla tego użytkownika. Skąd wiemy, że użytkownik jest tym, za kogo się podaje? Oni identyfikują się poprzez wysłanie poprawne hasło do swojego konta.

Hasło użytkownika jest jednym z tych atrybutów konta na poziomie globalnym, które nie są powiązane z określoną bazą danych lub tabelą. Żyje również w `mysql`.`user`tabeli. Jeśli użytkownik nie ma żadnych innych uprawnień ON *.*, są one nadawane USAGE ON *.*i wyświetlany jest tam skrót jego hasła. Często

jest to efekt uboczny CREATE USER wypowiedzi. Gdy użytkownik jest tworzony w ten sposób, początkowo nie ma żadnych uprawnień, więc są one jedynie przyznawane USAGE.

Przykład:
(potwierdzenie użytkownika i hasła)

*GRANT USAGE ON *.* TO anna IDENTIFIED BY 'Anna4!';*

(nadawanie uprawnień INSERT,UPDATE,SELECT do tabeli zgloszenia dla użytkownika anna)

*GRANT INSERT,UPDATE,SELECT ON zgloszenia.*0) TO anna;*

e) Słowo kluczowe ORDER BY służy do sortowania zestawu wyników w kolejności rosnącej lub malejącej.

Słowo kluczowe ORDER BY domyślnie sortuje rekordy w kolejności rosnącej. Aby posortować rekordy w kolejności malejącej, użyj słowa kluczowego DESC.

SELECT *column1*, *column2, ...*
FROM *table_name*
ORDER BY *column1, column2, ...* ASC|DESC;

Przykłady:
Poniższa instrukcja SQL wybiera wszystkich klientów z tabeli „Customers", posortowanych malejąco według kolumny „Country":
SELECT * FROM Customers
ORDER BY Country DESC;

Poniższa instrukcja SQL wybiera wszystkich klientów z tabeli „Customers", posortowanych rosnąco według kolumny „Country" i malejąco według kolumny „CustomerName":
SELECT * FROM Customers
ORDER BY Country ASC, CustomerName DESC;

KROK 4

Witryna internetowa
(przykład)

Przygotowanie grafiki:

- Plik *obraz.jpg*, wypakowany z archiwum, należy przeskalować z zachowaniem proporcji tak, aby jego wysokość wynosiła dokładnie XXX px (jeżeli będzie samo skalowanie to można wykorzystać np. MS Paint, jeżeli będzie skalowanie z wycięciem tła to GIMP'a)

Cechy witryny:

- Składa się ze strony o nazwie *nazwa.html* oraz skryptu *nazwa_skryptu.php*. Poniższe wymagania dotyczą tylko pliku *nazwa.html*
- Zastosowany właściwy standard kodowania polskich znaków
- Tytuł strony widoczny na karcie przeglądarki: „TYTUŁ"
- Arkusz stylów w pliku o nazwie nazwa_*styl.css* prawidłowo połączony z kodem strony
- Podział strony na bloki:

Tu będzie zdefiniowany podział na bloki div – pamiętajcie o nadanie odpowiednich nazw bloków które będą miały swoje odzwierciedlenie w pliku css !

Zanim przejdziemy do szczegółów bloków to przedstawię wzorcową strukturę pliku html:

```
<!DOCTYPE html>
<html>
 <head>
  <title>Tytuł strony</title>
  <meta charset="utf-8">
  <meta name="author" content="Tu wpisz swój PESEL">
  <link rel="stylesheet" href="styl1.css">
 </head>
<body>

Tu będzie treść strony

</body>
</html>
```

Wprowadzamy podział na bloki (div)

Sekcje strony

Do budowy sekcji możemy użyć elementu blokowego (ang. division – oddział, wydział, sekcja). Element ten nazywany jest również box'em (pudełkiem), albo kontenerem (ang. container, contain – zawierać w sobie), z tego względu, że zawiera on w sobie inne elementy.

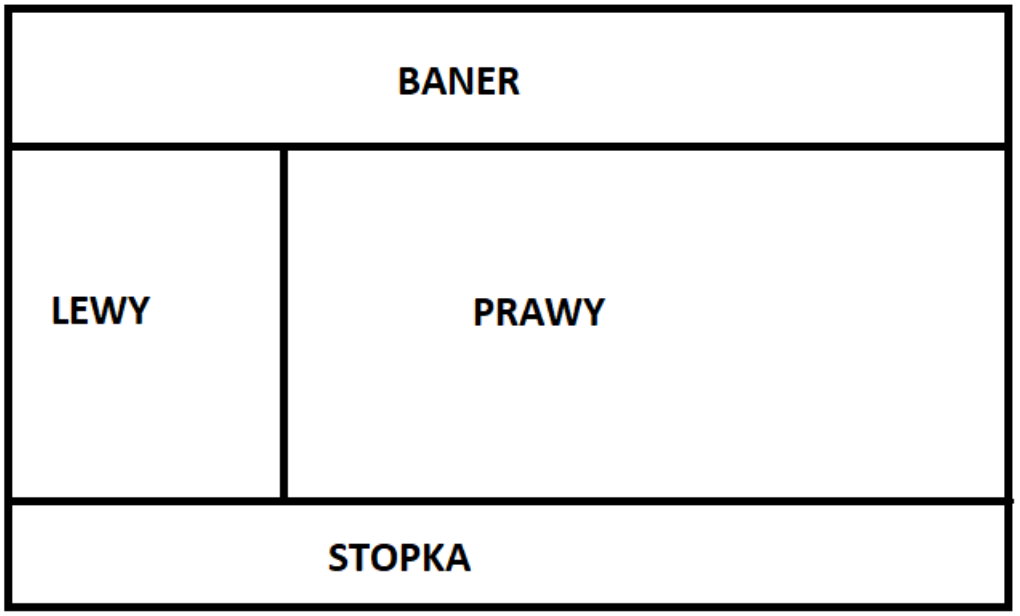
Element jest elementem blokowym, tzn. zajmie całą szerokość strony. Sam element nic szczególnego nie wnosi do naszego kodu, jeśli chodzi o formatowanie. Domyślnie nie zostanie dodana żadna ramka, czy inny kolor tła, żeby wizualnie odróżnić poszczególne sekcje.

W celu odróżnienia poszczególnych sekcji od siebie, możemy posłużyć się znanym nam już atrybutem id, gdyż atrybut ten nadaje unikalność elementom html (żaden inny element html nie powinien posiadać tego samego id). Można powiedzieć, że wartości

atrybutu id jaką będziemy nadawać poszczególnym sekcjom to są ich nazwy. Poszczególnym sekcjom możemy nadać dowolne nazwy, jednak powszechnie stosowanymi są angielskie, m.in. header, main oraz footer. Dodajmy nazwę do naszej sekcji nagłówkowej poprzez atrybut id.

```
<div id="header"></div>
```

Przykład:



Plik.html (index.html)

```
<!doctype html>
<html lang="pl">
    <head>
        <meta charset="UTF-8" />
        <meta name="author" content="Tu wpisz swój PESEL">
        <title>Tytuł strony na pasku...</title>
                <link rel="stylesheet" href="styl.css" type="text/css" />
    </head>
    <body>
        <div id="baner">
        </div>
        <div id="lewy">

        <div id="prawy">

        </div>
        <div id="stopka">
```

```
        </div>
    </body>
</html>
```

Do tego pliku - styl.css

```
#baner
{
        padding-top: 80px;

        float: left;
        width: 100%;
      font-size: 85px;
        color: yellow;
      border-width: 1px;
        border-style: solid;
        border-color: red;
        height: 200px;
        background-color: red


}

#lewy
{
        float: left;
        width: 50%;
}

#prawy
{
        float: right;
        width: 50%;
}

#stopka
{
        clear: both;
        width: 100%;

}
```

HTML – podstawowe pojęcia

Do zbudowania poprawnej strony potrzebujemy znaczników HTML:

Formatowanie Tekstu – Elementy HTML

wyraz pogrubiony
<b> wyraz pogrubiony
* wyraz pochylony*
<i> wyraz pochylony
<small> wyraz pomniejszony
<mark> podświetlony wyraz
<sup> indeks górny
<sub> indeks dolny
<s> wyraz nieaktualny
<ins> wyraz wstawiony
<del> wyraz usunięty

Lista Uporządkowana HTML

Listy uporządkowane w języku html to takie, których elementy posiadają pewną numerację, stąd też zwyczajowa nazwa lista numerowana. Numeracje w tych listach możemy uzyskać poprzez cyfry, bądź litery.

element listy
element listy

element listy
element listy

element listy
element listy

Listy uporządkowane tworzymy przy pomocy elementu <ol> (ang. ordered list – lista uporządkowana).

Jednak sam zapis znacznikowy listy uporządkowanej <ol> nie wystarczy, do tej listy musimy jeszcze dodać tzw. elementy listy.
Element listy HTML

Elementy listy to po prostu kolejne jej podpunkty. Tworzymy je przy pomocy elementu <li> (ang. list item – element listy). Pomiędzy znacznikami <li> oraz dodajemy kolejne elementy listy.

```
<ol>
  <li>element listy</li>
  <li>element listy</li>
</ol>
```

element listy
element listy

Element <ol> wskazuje jedynie na rodzaj listy. Wszystkie elementy listy muszą się znaleźć pomiędzy znacznikami <li> oraz .

Lista Nieuporządkowana HTML

Drugim rodzajem listy w języku html, jest lista nieuporządkowana, zwana też listą wypunktowaną. Elementy tej listy nie są numerowane, a luźno wymienione po tzw. myślnikach. Domyślnie rolę tych myślników przejmują kropki.

element listy
element listy

Listę nieuporządkowaną tworzymy za pomocą elementu <ul> (ang. unordered list – lista nieuporządkowana). Tak jak w liście uporządkowanej elementy listy również tworzymy poprzez element <li>.

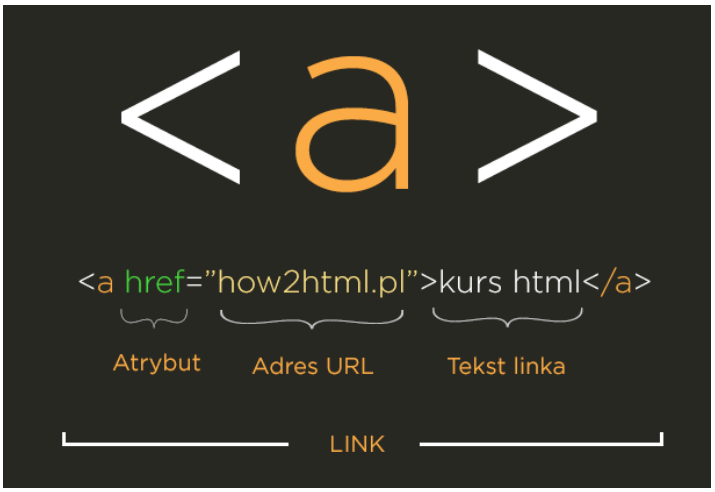
```
<ul>
  <li>element listy</li>
  <li>element listy</li>
</ul>
```

element listy
element listy

Jeśli chcemy zatytułować listę nieuporządkowaną również musimy skorzystać ze znanego nam już elementu nagłówka.

```
<h3>Lista nieuporządkowana</h3>
<ul>
  <li>element listy</li>
  <li>element listy</li>
</ul>
```

Link HTML



Najczęściej używane wartości atrybut target

_self - otwiera link w tym samym oknie / karcie (wartość domyślna)
_blank - otwiera link w nowej karcie

```
<a href="how2html.pl" target="_self">kurs html</a>
<a href="how2html.pl" target="_blank">kurs html</a>
```

```
<a href="kwerendy.txt" download>Pobierz kwerendy</a>
```

Tabela HTML – Komórka

<td>

Żeby nasza tabela była kompletna i poprawnie wyświetlała dane, należy jeszcze dodać do niej komórki. Komórki tabeli dodajemy poprzez element <td> (ang. table data, table cell – dane tabeli, komórka tabeli). Do rozwijania skrótu td powszechnie używa się table cell, jednak nazwa pochodzi od table data.

```
<table>
  <tr>
    <td>Treść</td>
  </tr>
</table>
```

Treść umieszczamy zawsze w elemencie <td>. Element ten podobny jest trochę do kolumny. Jednak nie zapisujemy jej raz tak jak ma to miejsce w przypadku wiersza, tylko musimy za każdym razem dodać tyle komórek (kolumn) do wiersza ile aktualnie nam potrzeba.

```
<table>
  <tr>
    <td>1 wiersz 1 kolumna</td>
    <td>1 wiersz 2 kolumna</td>
  </tr>
  <tr>
    <td>2 wiersz 1 kolumna</td>
    <td>2 wiersz 2 kolumna</td>
  </tr>
</table>
```

1 wiersz 1 kolumna	1 wiersz 2 kolumna
2 wiersz 1 kolumna	2 wiersz 2 kolumna

Klasy i id

Identyfikator
id

Do identyfikacji poszczególnych elementów używamy tzw. identyfikatorów w skrócie id. Id to atrybuty elementów html, które pozwalają je odróżniać. Zapisujemy je skrótem id i przypisujemy do nich wartości tak jak do innych atrybutów, używając do tego znaku = oraz " ".

```
id="some-id"
```

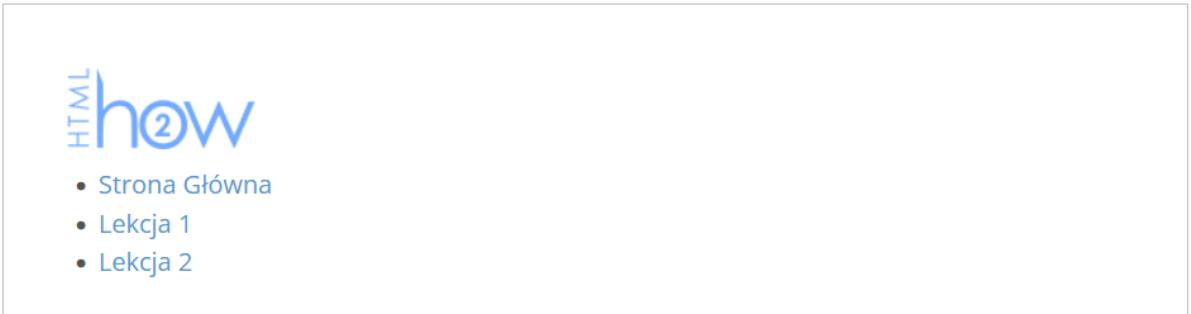
Atrybut id jest unikalnym identyfikatorem elementu html. Znaczy to, że daną wartość atrybutu id możemy przypisać jedynie jednemu elementowi html.

```
<div id="header"></div>
<div id="footer"></div>
```

Oznaczyliśmy w ten sposób dwie sekcje, nagłówkową (ang. header) i stopkę (ang. footer). Nazwy te nie są standardem w języku html, a dobrą praktyką. Występuje pełna dowolność w nadawaniu nazw atrybutu id. Ja jednak polecam używanie nazw angielskich, wtedy nasz kod będzie prawdziwie międzynarodowy…

Oznaczmy teraz naszą sekcję nagłówkową id o wartości header.

```
<div id="header">
   <img src="logo.png" alt="logo how2html">
   <ul>
      <li><a href="#">Strona Główna</a></li>
      <li><a href="lekcja-1">Lekcja 1</a></li>
      <li><a href="lekcja-2">Lekcja 2</a></li>
   </ul>
</div>
```



Formularz HTML

<form action="">

W celu wysłania danych naszym formularzem html, musielibyśmy stworzyć plik z instrukcjami np. w języku PHP i zapisać go w folderze z naszą stroną internetową np. pod nazwą form.php. Do elementu <form> dodalibyśmy następnie atrybut action, który wskazywałby ścieżkę dostępu do naszego pliku form.php. Po naciśnięciu klawisza submit akcja z pliku form.php zostałaby wykonana.

```
<form action="form.php">
  <label>Twój Login <input type="text"></label>
  <label>Twój email <input type="email"></label>
  <label>Twoje hasło <input type="password"></label>
  <input type="submit">
</form>
```

Combobox HTML

Element HTML <select> reprezentuje kontrolkę, która udostępnia menu opcji:

```
<label for="pet-select">Choose a pet:</label>

<select name="pets" id="pet-select">
```

```
    <option value="">--Please choose an option--</option>
    <option value="dog">Dog</option>
    <option value="cat">Cat</option>
    <option value="hamster">Hamster</option>
    <option value="parrot">Parrot</option>
    <option value="spider">Spider</option>
    <option value="goldfish">Goldfish</option>
</select>
```

<input type="liczba">

<input> elementy typu **numer** służą do umożliwienia użytkownikowi wprowadzenia liczby. Obejmują one wbudowaną walidację w celu odrzucenia wpisów nie liczbowych.

Przeglądarka może zdecydować się na udostępnienie strzałek krokowych, aby użytkownik mógł zwiększać i zmniejszać wartość za pomocą myszy lub dotykając palcem.

Przykład:

```
<input type="number" id="tentacles" name="tentacles"
      min="10" max="100">
```

Wstawienie obrazka

W jaki sposób wstawić obrazek (grafikę, zdjęcie) na stronę WWW?

<img src="Tu podaj względną ścieżkę dostępu do obrazka" alt="Tu podaj tekst alternatywny">

Zamiast tekstu: "Tu podaj względną ścieżkę dostępu do obrazka", należy podać miejsce na dysku, gdzie znajduje się nasz obrazek, który chcemy umieścić na stronie. Jeżeli obrazek znajduje się w tym samym katalogu co strona, na której chcemy go wstawić, wystarczy wpisać tutaj samą nazwę pliku obrazka, nie zapominając przy tym o podaniu rozszerzenia (".jpg" lub ".gif").
Natomiast w miejsce "Tu podaj tekst alternatywny" wpisuje się krótką informację, która pojawi się w przypadku, kiedy obrazek nie zostanie wyświetlony (np. jeśli użytkownik wyłączy wyświetlanie grafiki w swojej przeglądarce internetowej).

CSS podstawy

Zależność pomiędzy językiem HTML, a CSS można zapisać następująco. Język Html opisuje dane elementy na stronie, możemy powiedzieć, że służy do nazywania ich. W języku html możemy określić nagłówek i paragraf. Natomiast w języku CSS możemy określić jaki rozmiar i kolor czcionki mogą one posiadać.

Rozwinięciem skrótu CSS (ang. Cascading Style Sheets) są kaskadowe arkusze stylów. Słowo arkusz można traktować jako plik, w którym zapisujemy reguły formatowania. Reguły formatowania przypisujemy do konkretnych elementów HTML poprzez znane nam z lekcji wcześniejszej selektory. Dowiedzmy się trochę więcej o samych regułach formatowania, a później przejdziemy do selektorów.

Reguły CSS to inaczej zbiory zasad służące do formatowania treści. Zbudowane są one z własności (ang. property) oraz jej wartości (ang. value).

Nazwa wartości musi być zawsze poprzedzona dwukropkiem : , a cała reguła zakończona średnikiem ; .

własność: wartość;

Jeśli nazwa własności jest kilkuczłonowa, zazwyczaj kolejne wyrazy oddzielane są myślnikiem -, tak jak background-color.

Własność background-color używana jest do dodania koloru tła do elementu. Wartość blue to kolor niebieski.

background-color: blue;

Jeśli do własności dodamy więcej niż jedną wartość, wówczas oddzielamy je zwykłą spacją.

font: 16px Arial;

Przy pomocy reguły font (ang. czcionka) możemy modyfikować m.in. rozmiar czcionki i jej rodzaj (rodzinę).

Rozmiar czcionki podajemy najczęściej w znanych nam już pikselach px. Przez rozmiar czcionki mam na myśli jej wysokość, liczoną od szczytu najwyższej litery do dołu litery, która jest najniżej, np. ly.

Deklaracja czcionki, której chcemy użyć, wymaga znajomości jej pełnej nazwy.

PHP - Proste operacje matematyczne

```
<?
$iNumber1 = 5;
$iNumber2 = 3;
?>
```

i wykonujemy na nich działania:

```
<?php
$iResult1 = $iNumber1 + $iNumber2; // dodawanie (wynik 5+3=8)
$iResult2 = $iNumber1 - $iNumber2; // odejmowanie (wynik 5-3=2)
$iResult3 = $iNumber1 * $iNumber2; // mnożenie (wynik 5*3=15)
$iResult4 = $iNumber1 / $iNumber2; // dzielenie (wynik 5/3=1.6666666666667)
// modulo, reszta z dzielenia całkowitego
$iResult5 = $iNumber1 % $iNumber2; // wynik 5%3=2
?>
```

Formularz – PHP, MySQL

Tyle teorii – teraz zajmiemy się przykładowy połączeniem formularza ze strony html przesyłanego do poliku skrypt.php !!!!

Dla pliku w HTML w sekcji <body> :

```
<table>
<tr>
<td><form action="skrypt.php" method="post">
Imię:<br>
<input type="text" name="imie" /><br>
Nazwisko:<br>
<input type="text" name="nazwisko" /><br>
e-mail:<br>
<input type="text" name="email" /><br>
<br><br>
<input type="submit" value="Dodaj" />
</form></td>
</tr>
</table>
```

<table>	*Wstawieniu tabelę – znacznik otwarcia tabeli*
<tr>	*table row - wiersz tabeli. Element HTML, za pomocą, którego dodajemy wiersze.*
<td>	*table data - komórka tabeli. Element HTML, za pomocą, którego dodajemy treść do tabeli. Jeśli element potraktujemy jako wiersz tabeli, element możemy tłumaczyć jako kolumna tabeli.*
<form action="skrypt.php" method="post">	*Przesyłamy dane do pliku skrypt.php*
*Imię: <input type="text" name="imie" /> *	*Wprowadzamy pole „imię:” typu tekstowego oraz nadajemy mu nazwę „imie”*

	Tę zmienną pod tą nazwą będziemy przesyłać do pliku skrypt.php
*Nazwisko: * *<input type="text" name="nazwisko" /> * *e-mail: * *<input type="text" name="email" /> *	*Analogicznie jak wyżej.* *Znacznik oznacza przejście linię niżej*
<input type="submit" value="Dodaj" />	*Przycisk przesyłający formularz*
</form></td> *</tr>* *</table>*	*Zamykanie znaczników*

Plik skrypt.php

```
<?PHP
// odbieramy dane z formularza

$xi = $_POST['imie'];
$xn = $_POST['nazwisko'];
$xe = $_POST['email'];


$baza = mysqli_connect("localhost", "root", "", "baza");

if ($baza->connect_error) {
    die("Connection failed: " . $baza->connect_error);
}

$sql1 = "INSERT INTO ksiazka (imie, nazwisko, email)
VALUES ('$xi','$xn','$xe')";

if ($baza->query($sql1) === TRUE) {
    echo "Dodano nowy rekord","<br>";
} else {
    echo "Błąd: " . $sql1 . "<br>" . $baza->error;
}
echo "<br>";
$sql2 = "SELECT * FROM ksiazka";
$result = $baza->query($sql2);

if ($result->num_rows > 0) {
    // Wyświetlenie tabeli w postaci wierszy

    while($row = $result->fetch_assoc()) {
```

```
        echo  "id: " . $row["id"]. "    Imię: " . $row["imie"]."  
Nazwisko:" . $row["nazwisko"]. "    e-mail: " . $row["email"]. "<br>";
    }
} else {
    echo "0 results";
}
$baza->close();
?>
```

<?PHP	Otwarcie skryptu w języku PHP
// odbieramy dane z formularza $xi = $_POST['imie']; $xn = $_POST['nazwisko']; $xe = $_POST['email'];	Odbieranie danych z formularza metodą POST Symbol $ - informujemy, że wprowadzamy zmienną
$baza = mysqli_connect("localhost", "root", "", "baza");	Definiujemy zmienną $baza składającą się z komendy mysqli_connect.

if ($baza->connect_error) {
  die("Connection failed: " . $baza->connect_error);
}
	Instrukcja warunkowa – jeżeli połączenie z bazą zostało zainicjowane poprawnie to przechodzi następnej komendy PHP – jeżeli nie to wyświetlony zostanie komunikat „Connection failed:”

$sql1 = "INSERT INTO ksiazka (imie, nazwisko, email) VALUES ('$xi','$xn','$xe')";	Wprowadzamy zmienną z poleceniem SQL dodawania rekordu do bazy danych

if ($baza->query($sql1) === TRUE) {
  echo "Dodano nowy rekord","
";
} else {
  echo "Błąd: " . $sql1 . "
" . $baza->error;
}
	Instrukcja warunkowa sprawdza czy zapytanie SQL zostało poprawnie wykonane - wyświetlony zostanie komunikat "Dodano nowy rekord".
	
	Do tego momentu skrypt nic więcej nie wyświetli na stronie !

echo " ";	Komenda - echo " "; Służy do wykonania znacznika z HTML
$sql2 = "SELECT * FROM ksiazka"; $result = $baza->query($sql2);	Wprowadzamy kolejne zmienne – pierwsza zmienna to zapytanie w języku SQL (Select … from) Druga zmienna wykonuje połączenie z bazą danych i wykonanie powyższego zapytania SQL

*if ($result->num_rows > 0) {
  // Wyświetlenie tabeli w postaci wierszy
 	
  while($row = $result->fetch_assoc()) {
  echo "id: " . $row["id"]. "   Imię: " . $row["imie"]."   Nazwisko:" . $row["nazwisko"]. "   e-mail: " . $row["email"]. "
";
  }
} else {
  echo "0 results";
}*
	*Instrukcja warunkowa która sprawdza czy połączenie zostało wykonane i poprawnie zinterpretowane zapytanie SQL.
	Efektem będzie wyświetlenie w wierszach rekordów z bazy danych.
	
	  – spacja w języku HTML*

$baza->close(); ?>	*Zamknięcie połączenia z bazą danych – KONIECZNE !!!*

Wyszukiwarka

Plik html

```
<body>
<table>
<tr>
<td><form action="szukaj.php" method="post">
<b>Kogo szukasz?<br />
<input type="text" name="szukaj" /><br />

<br />
<br />
<input type="submit" value="Szukaj" />
</form></td>
</tr>
</table>
```

Plik szukaj.php

```
<?PHP
// odbieramy dane z formularza

$szukaj = $_POST['szukaj'];


$baza=mysqli_connect("localhost", "root", "", "baza");
if(!$baza)
echo 'BŁĄD: '.mysqli_connect_error();
                        else
                        {
                                $zapytanie="SELECT * FROM ksiazka WHERE imie or
nazwisko like '%$szukaj%' ;";
                                $wynik=mysqli_query($baza,$zapytanie);
                                 if(!$wynik)
                                        echo 'BŁĄD ZAPYTANIA: '.mysqli_error($baza);
                                 else
                                 {
                                        while($wiersz=mysqli_fetch_row($wynik))
                                        echo "id:  ".$wiersz[0].' &nbspImię: 
'.$wiersz[1].'&nbspnazwisko:  '.$wiersz[2].'&nbspe-mail: 
'.$wiersz[3].'<br>';
                                 }
                                 mysqli_close($baza);
                        }

                        ?>
```

COMBOBOX z danymi z mySQL

```
<?php
$baza = mysqli_connect("localhost", "root", "", "baza");

if ($baza->connect_error) {
   die("Connection failed: " . $baza->connect_error);
}
?>


<tr>
      <strong> Do kogo wysłać? : </strong>
          <select name="usluga">
             <option value=""> Do kogo? </option>
```

```
        <?php
 $query = "SELECT email FROM ksiazka ";
            $dd_res =  mysqli_query($baza, $query);
            while($r = mysqli_fetch_row($dd_res))
            {
                echo "<option value='$r[0]'> $r[0]  </option>";

            }
  mysqli_close($baza);
        ?>
         </select>
   </tr>
```

Formularz pocztowy – wysyła email z klienta skonfigurowanego w domyślnym programie pocztowym użytkownika

```
<h2>Kontakt</h2>
   <form action="mailto:e-mail" method="post" enctype="text/plain">
   <table id="tabela">
   <tr>
      <td>Imię:</td><td><input type="text" name="imie"></td>
   </tr>
    <tr>
      <td>Nazwisko:</td><td><input type="text" name="nazwisko"></td>
   </tr>
   <tr>
      <td>E-mail:</td><td><input type="text" name="email"></td>
   </tr>
   <tr>
   <td>Usługa: </td><td><textarea name="usluga" cols="40"
rows="8"></textarea></td>
   </tr>
   <tr>
      <td></td>
      <td><input type="checkbox" name="regulamin"> Zapoznałam/em się z
regulaminem</td>

   </tr>
   <tr>
   <td></td>
   <td><input type="reset" value="Resetuj">
        <input type="submit" value="Prześlij"></td>

   </tr>
   </table>    </form>
```

Do CSS

Ustawienie bloków
nav, main, aside {
float: left;
}
footer {
clear: both;
}
Wyrównywanie
selektor {
text-align: ;
}
Kolor tła i tekstu
selektor {
background: ;
color: ;
}
Marginesy zewnętrzne, wewnętrzne i obramowanie
selektor {
margin: ;
padding: ;
border: ;
}
Wysokość i szerokość
selektor {
height: ;
width: ;
}
Font
selektor {
font: ;
}
Podkreślenie tekstu
selektor {
text-decoration: ;
}

JavaScript - podstawy

1. Wyrażenia

WSTĘP

JavaScript
Skryptowy język programowania. Kod skryptu zapisujemy w pliku o rozszerzeniu .js
<script src="skrypt.js"></script>
lub bezpośrednio w stronie w elemencie script dokumentu HTML, który może być umieszczony w elemencie head lub body.
Do uruchomienia skryptu JavaScript wymagane jest przeglądarka internetowa.
Wykonanie kodu wymaga interpretowania.

Przykład interaktywny 1. Wstęp

```
<!DOCTYPE html>
<html>
 <head>
  <title>Przykład 1</title>
  <meta charset="utf-8" />
  <script>
document.write("Witaj")
  </script>
 </head>
 <body>
  <script>
document.write(" świecie")
  </script>
 </body>
</html>
```

Witaj świecie

Sposoby wprowadzania danych:

- metoda prompt()
- pola formularzy (document.getElementById(id).value)
- elementy strony własność innerHTML

Sposoby wyprowadzania:

- metoda alert()
- pola formularzy (document.getElementById(id).value)
- metody document.write() i document.writeln()
- elementy strony, własność innerHTML
- konsola console.log()

KONWENCJE

W języku JavaScript wielkość liter ma znaczenie.

Instrukcje oddzielamy średnikiem ; który na końcu linii jest opcjonalny (ale zalecany).

KOMENTARZE

- jednoliniowy: *// tekst komentarza*
- wieloliniowy: */* tekst komentarza */*

INNE

- nowa linia: \n

Przykład interaktywny 2. Konwencje leksykalne

```
<script>
//alert("te\nst")
</script>
```

TYPY ZMIENNYCH

- **Undefined** - niezdefiniowany, wartość undefined
- **Null** - pusty, wartość null
- **Boolean** - logiczny, wartości: true i false
- **String** - ciąg znaków, wartości w cudzysłowiu " " lub ' '
- **Number** - numeryczny, liczby całkowite, wymierne (separatorem dziesiętnym jest kropka)
- **Object** - obiekty języka EcmaScript i DOM

Deklaracja
Deklarację zmiennej poprzedzamy var (w trybie ścisłym obowiązkowo).
Typ zmiennej w języku JavaScript zostaje określony poprzez przypisanie wartości.

Przykład interaktywny 3. Typy zmiennych

```
<pre><script>
document.writeln(typeof '')
document.writeln(typeof 1)
document.writeln(typeof true)
document.writeln(typeof {})
document.writeln(typeof [])
</script></pre>
```

```
string
number
boolean
object
object
```

W przykładzie zastosowano operator `typeof`, który zwraca typ argumentu.

KONWERSJA ZMIENNYCH

W języku JavaScript zmienne są konwertowane automatycznie (na podstawie tablic konwersji).

np. +"l" (zwraca 1)

WYRAŻENIA

Literały

Obiektu

- {} literał obiektu

Przykład
var o = {a:1, b:2}
Pary *klucz : wartość*. Odwołujemy się przez *zmienna.klucz* np. o.a

Tablic

- []

Przykład
var t = []

Tablice indeksujemy od zera. Odwołujemy się przez np. t[i], gdzie t jest zmienną, i jest indeksem elementu.

Operatory

Operatory arytmetyczne

Dodawanie (+)
Operator dodawania zwraca sumę argumentów lub
jeśli chociaż jeden argument był łańcuchem znaków, ich **konkatenacje**.

Odejmowanie (-)
Operator odejmowania zwraca różnicę między odjemną, a odjemnikiem.

Mnożenie (*)
Operator mnożenia zwraca iloczyn czynników.

Dzielenie (/)
Operator dzielenia zwraca iloraz. Po lewej stronie jest dzielna, a po prawej dzielnik.

Dzielenie modulo (%)
Operator modulo zwraca resztę z dzielenia pierwszej liczby przez drugą.

Inkrementacja (++)
Operator inkrementacji zwiększa o 1 wartość argumentu i zwraca taką wartość (x++ definiujemy jako x=x+1).

- Przyrostek (x++) będzie działał jako **postinkrementacja**, zwracana wartość nie będzie powiększona.
- Przedrostek (++x) będzie działał jako **preinkrementacja**, zwracana wartość będzie powiększona.

Dekrementacja (--)
Operator dekrementacji zmniejsza o 1 wartość argumentu i zwraca taką wartość (x-- definiujemy jako x=x-1).

- Przyrostek (x--) będzie działał jako **postdekrementacja**, zwracana wartość nie będzie pomniejszona.
- Przedrostek (--x) będzie działał jako **predekrementacja**, zwracana wartość będzie pomniejszona.

Operatory przypisania

- podstawowy (=),
- mieszane (+=), (*=), (-=), (/=)

np. x+=y <=> x=x+y

Operatory porównania

zwracają wartość logiczną true lub false zależną od tego, czy wartość porównania jest fałszywa czy prawdziwa.

- równe (==),
- większe (>),
- mniejsze (<),
- większe lub równe (>=),
- mniejsze lub równe (<=),
- ściśle równe (===) (sprawdzany jest typ argumentów)

Operatory logiczne

łącza zdania logiczne

- i (&&),
- lub (||)
- nie (!)

Inne

Kolejność wykonywania

Nawiasy ()

Przykład interaktywny 4. Wyrażenia

```
<pre><script>
var x = y = a = b = 1
document.writeln("x++ = "+(x++))
document.writeln("++y = "+(++y))
document.writeln("a-- = "+(a--))
document.writeln("--b = "+(--b))
</script></pre>
```

```
x++ = 1
++y = 2
a-- = 1
--b = 0
```

2. Instrukcje i funkcje

INSTRUKCJE

- {} - blok instrukcji
- var - deklaracja zmiennych

WARUNKOWE

- if

- if else
- switch

PĘTLE

- while
- do while
- for

FUNKCJE

Definiowanie funkcji
function *nazwa_funkcji*(*argumenty*) {*instrukcje*;}
nazwa_funkcji - musi być poprawnym identyfikatorem JavaScript
argumenty - opcjonalne, oddzielone przecinkami
instrukcje - umieszczone w bloku {}

- function
- return
- this
- new

OBIEKTY

nazwaobiektu.mazwametody()

Funkcje globalne

- isNaN
- NaN
- parseInt()
- parseFloat()

PREDEFINIOWANE

- Date: Date(), getMonth(), getDay()
- Math: Math.max(), Math.pow()
- .substring
- .length - długości napisu (stringa) lub tablicy
- array

DOM

DOM - (ang. Document Object Model, obiektowy model dokumentu) dostarcza metod i własności, które w języku JavaScript pozwalają na pobieranie i modyfikowanie elementów strony wyświetlonej przez przeglądarkę.

window

- prompt()
- alert()

document

- document.write()
- document.getElementById() - metoda odwołuje się do znacznika o podanym id.
- document.createElement()
- document.getElementsByClassName() - metoda odwołuje się do podanej klasy.

element

- .innerHTML()
- .appendChild()
- .setAttribute()
- .*nazawa_atrybutu* - np. style (style.color),

zdarzenia

- onkeydown
- ondblclick - podwójne kliknięcie elementu
- onload
- onclick - kliknięcie elementu, np. przycisku
- onselect

Tabela 1. Wybór funkcji języka PHP do obsługi bazy MySQLi i MariaDB

Funkcje biblioteki MySQLi	Zwracana wartość
mysqli_connect(*serwer, użytkownik, hasło, nazwa_bazy*)	id połączenia lub FALSE, gdy niepowodzenie
mysqli_select_db(*id_polaczenia, nazwa_bazy*)	TRUE/FALSE w zależności od stanu operacji
mysqli_error(*id_polaczenia*)	Tekst komunikatu błędu
mysqli_close(*id_polaczenia*)	TRUE/FALSE w zależności od stanu operacji
mysqli_query(*id_polaczenia, zapytanie*)	Wynik zapytania
mysqli_fetch_row(*wynik_zapytania*)	Tablica numeryczna odpowiadająca wierszowi zapytania
mysqli_fetch_array(*wynik_zapytania*)	Tablica zawierająca kolejny wiersz z podanych w wyniku zapytania lub FALSE, jeżeli nie ma więcej wierszy w wyniku zapytania
mysqli_num_rows(*wynik_zapytania*)	Liczba wierszy w podanym zapytaniu
mysqli_num_fields(*wynik_zapytania*)	Liczba kolumn w podanym zapytaniu

UWAGA: po zakończeniu pracy utwórz plik tekstowy. Zapisz w nim nazwę przeglądarki internetowej, w której weryfikowałeś poprawność działania witryny. Nazwij plik przeglądarka.txt i zapisz go w folderze z numerem PESEL. Nagraj płytę z rezultatami pracy.

Czas przeznaczony na wykonanie zadania wynosi 150 minut.
Ocenie będzie podlegać 5 rezultatów:
– operacje na bazie danych,
– zawartość witryny internetowej,
– działanie witryny internetowej,
– styl CSS witryny internetowej,
– skrypt połączenia z bazą.

Przycisk animowany w CSS

```
#przycisk {
  /*przeźroczystość przycisku*/
  background: transparent;
  /*dopełnienie ze wszystkich stron 25px;*/
  padding: 25px;
  /*obramowanie - szerokość 2px, linia ciągła, kolor biały*/
  border: 2px solid #fff;
  /*kolor czcionki*/
  color: #fff;

  /*teraz dodamy efekt płynnego przejścia*/
  transition: 0.5s;
}

/*możemy teraz opisać jak ma wyglądać przycisk po najechaniu na niego kursorem*/
#przycisk:hover {
  /*tło - kolor biały*/
  background: #fff;
  /*czcionka w kolorze tła strony*/
  color: #415967;
  /*możemy również zmienić rodzaj kursora, np. na "łapkę"*/
  cursor: pointer;
}
```

Obliczanie wskaźnika BMI – javascript

```
<script type="text/javascript">
function f()
{
   // przypisanie obiektu formularza do zmiennej

var f = document.forms['BMI'];
   // Sprawdzenie masy ciała
    if (f.masa.value == '' && f.wzrost.value == '')
   {
       alert('Musisz podać masę oraz wzrost!');
       f.masa.focus();
       return false;
   }
if (f.masa.value == '')
   {
       alert('Musisz podać swoją masę!');
       f.masa.focus();
       return false;
   }
   // sprawdzenie wzrostu
   if (f.wzrost.value == '')
   {
       alert('Musisz podać wzrost w centymetrach!');
       f.wzrost.focus();
       return false;
   }
   // formularz jest wypelniony poprawnie
  var masa = f.masa.value;
  var wzrost = (f.wzrost.value / 100);
  var suma = masa / (wzrost * wzrost);
```

```
 if (suma <= '18') {
    var odp = 'Niedowagę';
  }
  else if (suma >= '18' && suma < '25') {
    var odp = 'Prawidłową masę ciała';
    }
  else if (suma >= '25' && suma < '30') {
    var odp = 'Nadwagę';
    }
  else if (suma >= '30') {
    var odp = 'Otyłość';
    };

  alert('Masz ' + odp + "\n" + suma + " w skali BMI");
}
</script>


</head>
<body>



<form id="BMI" method="post" onsubmit="return f()"><div>
Masa (w kg): <input type="text" onKeyUp="this.value=this.value.replace(/\D/g,'')"
name="masa"><br />
Wzrost (w cm): <input type="text" onKeyUp="this.value=this.value.replace(/\D/g,'')"
name="wzrost"><br />

<input type="submit" value="Wyślij">
</div></form>
```

Proste działania matematyczne w javascript

```
<!doctype html>
<html lang="pl">
    <head>
        <meta charset="UTF-8" />
        <meta name="author" content="Tu wpisz swój PESEL">
        <title>Tytuł strony na pasku...</title>

    </head>
    <body>

<main>
   <h1>PROSTE DZIAŁANIA</h1>
   <p><i>Podaj pierwszą liczbę:</i>
 <input type="number" id="i1"/></p>
   <p><i>Podaj drugą liczbę:</i>
 <input type="number" id="i2"/></p>
   <p>
 <button onclick="f(1)">DODAWANIE</button>
 <button onclick="f(2)">ODEJMOWANIE</button>
 <button onclick="f(3)">MNOŻENIE</button>
 <button onclick="f(4)">DZIELENIE</button>
   </p>
   <output id="o1"></output>
  </main>
  <script>
function f(x) {
var a=parseInt(i1.value)
var b=parseInt(i2.value)
var w
 if (isNaN(a) || isNaN(b)) {
 w = "Proszę uzupełnić obie liczby"
 }
 else if(x==4 && b==0) {
 w = "Nie wolno dzielić przez zero"
 }
 else {
  if (x==1) w=a+b
  else if(x==2) w=a-b
  else if(x==3) w=a*b
  else if(x==4 && b!=0) w=a/b
  w="Wynik działania wynosi: "+w
 }
o1.value=w
}
  </script>
 </body>
</html>
```

Javascript – „paliwo”

```
<html lang="pl">
<head>
    <meta charset="utf-8" />
    <title>zadanie 1</title>
      </head>
 <body>

<h1>Oblicz koszt paliwa</h1>

<label>Rodzaj paliwa (1-benzyna, 2-olej napedowy)</br><input type="number" id
="a"></label></br>
<label>Ile litrów:</br><input type="number" id="b"> </label> </br></br>
<input type="button" value="OBLICZ" onclick="paliwo()">

<div id="wynik" style="margin-top:20px;"></div>
<script>
function paliwo()
{
var a=document.getElementById("a").value;
var b=document.getElementById("b").value;

if (a=== '1')
{
    var typ= 4;
   var suma= typ*b;
  document.getElementById("wynik").innerHTML="Koszt paliwa"+ suma +"zł";
}

else if(a === '2')
{
 var  typ=3.5;
  var suma= typ*b;

  document.getElementById("wynik").innerHTML="Koszt paliwa"+ suma +"zł";

}
else
{

 document.getElementById("wynik").innerHTML="Koszt paliwa 0 zł";
}
}

 </script>

</body>
</html>
```

Javascript – „checkbox”

```
<!DOCTYPE HTML>
<html lang="pl">
<head>
    <meta charset="utf-8" />
    <title>zadanie 2</title>

</head>
 <body>

Checkbox: <input type="checkbox" id="myCheck" onclick="myFunction()">

<p id="text" style="display:none">Checkbox is CHECKED!</p>
<script>
function myFunction() {
  // Get the checkbox
  var checkBox = document.getElementById("myCheck");
  // Get the output text
  var text = document.getElementById("text");

  // If the checkbox is checked, display the output text
  if (checkBox.checked == true){
    text.style.display = "block";
  } else {
    text.style.display = "none";
  }
}
</script>

</body>
</html>
```

PHP – „ciasteczka”

```
<?php

    if(isset($_COOKIE["TestCookie"]))
    echo $_COOKIE["TestCookie"];
    else
    {
    echo 'Dzień Dobry! Sprawdź regulamin naszej strony.';
    $value='Miło nam, że nas znowu odwiedziłeś.';
    setcookie("TestCookie",$value,time()+3600);
    }


?>
```

Javascript – „ciągi” pętla for

```
<div id="prawy">
        <h2>Generowanie ciągu arytmetycznego</h2>
        Pierwszy wyraz pierwszy: <input type="number" id="pierwszy"><br>
        Różnica ciągu R: <input type="number" id="r"><br>
        Liczba wyrazów ciągu: <input type="number" id="n"><br>
        <button onclick="ciag()">Generuj Ciąg</button><br>
        <span id="wynik"></span>
    </div>
    <div id="stopka">
        <p>Autor: 0123456789</p>
    </div>
  </body>

 <script>
    function ciag()
    {
        var pierwszy = document.getElementById('pierwszy').value;
        pierwszy = parseInt(pierwszy);
        var r = document.getElementById('r').value;
        r = parseInt(r);
        var n = document.getElementById('n').value;
        n = parseInt(n);
        var wynik = "Ciąg artmetyczny zawiera wyrazy: ";
        wynik += pierwszy;
        for(var i = 1; i < n; i++) {
            pierwszy = pierwszy + r;
            wynik += ", " + pierwszy;
        }
        document.getElementById('wynik').innerHTML = wynik;
    }
  </script>
```

Javascript – „kolory” RGB

```
<div id="p">
        <p>Podaj numer kształtu</p>
        <input type="number" id="ksztalt">
        <p>skomponuj swój kolor, podaj RGB: </p>
        <label>R:<input type="number" id="R" min="0" max="255"></label>
        <label>G:<input type="number" id="G" min="0" max="255"></label>
        <label>B:<input type="number" id="B" min="0" max="255"></label><br>
        <button onclick=zamowienie()>Zamów</button>
        <p id="zelek">Wybrany kształt</p>
        <button id="kolor">Wybrany kolor</button>
    </div>

    <script>
        function zamowienie() {
            var ksztalt = +document.getElementById("ksztalt").value;
            if (ksztalt ==1 ) {
                document.getElementById("zelek").innerHTML = "Zamówiłeś żelka: miś";
            }
            else if (ksztalt == 2) {
                document.getElementById("zelek").innerHTML = "Zamówiłeś żelka:
żabka";
            }
            else if (ksztalt == 3) {
                document.getElementById("zelek").innerHTML = "Zamówiłeś żelka: serce";
            }
            else{
                document.getElementById("zelek").innerHTML = "Zamówiłeś żelka: inny
kształt";
            }
            var R = +document.getElementById("R").value;
            var G = +document.getElementById("G").value;
            var B = +document.getElementById("B").value;

document.getElementById("kolor").style.backgroundColor="rgb("+R+","+G+","+B+")";
        }
    </script>
```

Javascript – „zamówienie” , złożona instrukcja IF

```
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Hurtownia</title>
  <link rel="stylesheet" href="styl4.css">
  <script>
    function order() {
      var number = document.getElementById("numer").value;
      var weight = document.getElementById("waga").value;
      var price = 0;
      if (number == 1) {
        price = weight * 5;
      }
      else if (number == 2) {
        price = weight * 7;
      }
      else if (number == 3) {
        price = weight * 6;
      }
      else {
        price = 0;
      }

      document.getElementById("wynik").innerHTML = "Koszt zamówienia wynosi: " +
price + " zl";

    }
  </script>
</head>
<body>
  <section id="logo">
    <img src="logo.png" alt="hurtownia">
  </section>

  <section id="menu">
    <a href="index.html">Strona glowna</a>
    <a href="zamowienia.html">Zamówienia</a>
    <a href="blog.html">Blog o kawie</a>
    <a href="http://gatunki-kawy.pl/" target="_blank">Odwiedź także</a>
  </section>

  <section class="header">
    <h1>Hurtownia kawy</h1>
  </section>
```

```
<section class="main">
  <h2>Oferta</h2>
  <ol>
    <li>Kawa palona Arabica</li>
    <li>Kawa palona Robusta</li>
    <li>Kawa bezkofeinowa</li>
    <li>Kawa zielona</li>
  </ol>
</section>

<section class="main">
  Podaj numer kawy: <br />
  <input id="numer" type="number"> <br />
  Podaj wagę w dekagramach: <br />
  <input id="waga" type="number">
  <button onclick="order()">Zamów</button>
  <p id="wynik"></p>
</section>
```

PHP – „skrypty ze sklepu eko” - select, insert

```
<!-- SKRYPT 1 -->
        <?php
        $polaczenie=mysqli_connect("localhost","root","","dane2");
        if(!$polaczenie)
        echo 'BŁĄD:'.mysqli_connect_error();
        else
        {
        $zapytanie="SELECT
produkty.nazwa,produkty.ilosc,produkty.opis,produkty.cena,produkty.zdjecie FROM
produkty WHERE produkty.Rodzaje_id IN(1,2);";
        $wynik=mysqli_query($polaczenie,$zapytanie);
        if(!$wynik)
            echo 'BŁĄD ZAPYTANIA: '.mysqli_error($polaczenie);
        else
        {
            while($wiersz=mysqli_fetch_row($wynik))
                echo '<div id="skrypt1"><img src="'.$wiersz[4].'"
alt="warzywniak"><h5>'.$wiersz[0].'</h5><p>opis: '.$wiersz[2].'</p><p>na stanie
:'.$wiersz[1].'</p><h2>'.$wiersz[3].' zł</h2></div>';
        }

        }
        mysqli_close($polaczenie);
        ?>
<!--  KONIEC SKRYPTU 1 -->
```

```
  <form method="post" action="sklep.php">
      <label>Nazwa:</label><input type="text" name="nazwa">  
      <label>Cena:</label><input type="number" step="0.01" value="0.00"
name="cena"> <input type="submit" value="Dodaj produkt">
      </form>
      <p>Stronę wykonał: 31111111111</p>

<!-- SKRYPT 2 -->
<?php
   if(isset($_POST['nazwa']) and isset($_POST['cena']))
   {
   $polaczenie=mysqli_connect("localhost","root","","dane2");
   if(!$polaczenie)
      echo 'BŁĄD:'.mysqli_connect_error();
   else
   {
      $zapytanie="INSERT INTO produkty VALUES
(NULL,'1','4','".$_POST['nazwa']."','10','","'.$_POST['cena']."','owoc.jpg');";
      $wynik=mysqli_query($polaczenie,$zapytanie);
      if(!$wynik)
         echo 'BŁĄD ZAPYTANIA: '.mysqli_error($polaczenie);
      else
         echo 'REKORD ZAPISANO DO BAZY ;)';

   }
      mysqli_close($polaczenie);
   }

?>
<!--  KONIEC SKRYPTU 2 -->

         </div>
```

PHP – „skrypty ze wypożyczalni video” - select, insert, delete

```
<!doctype html>
<html lang="pl">
<head>
<meta charset="utf-8">
<title>Video On Demand</title>
<link rel="stylesheet" href="styl3.css" type="text/css">
</head>
<body>
<div id="baner1">
<h1>Internetowa wypożyczalnia filmów</h1>
</div>
<div id="baner2">
<table>
<tr>
    <th>Kryminał</th><th>Horror</th><th>Przygodowy</th>
</tr>
<tr>
<td>20</td><td>30</td><td>20</td>
</tr>
</table>
</div>
<div id="polecamy">
<h3>Polecamy</h3>


   <?php
$polaczenie=mysqli_connect("localhost","root","","dane3");
    if(!$polaczenie)
        echo 'BŁĄD:'.mysqli_connect_error();
    else
    {
        $zapytanie="SELECT
produkty.id,produkty.nazwa,produkty.opis,produkty.zdjecie FROM produkty WHERE
produkty.id IN(18,22,23,25);";
        $wynik=mysqli_query($polaczenie,$zapytanie);
        if(!$wynik)
            echo 'BŁĄD ZAPYTANIA: '.mysqli_error($polaczenie);
        else
        {
            while($wiersz=mysqli_fetch_row($wynik))
                echo '<div id="film1"><h4>'.$wiersz[0].'. '.$wiersz[1].'</h4><img
src="'.$wiersz[3].'" alt="film"><br>'.$wiersz[2].'</div>';
        }

    }
 mysqli_close($polaczenie);
```

```
   ?>


</div>
<div id="filmy_fantastyczne">
<h3>Filmy fantastyczne</h3>


    <?php
$polaczenie=mysqli_connect("localhost","root","","dane3");
   if(!$polaczenie)
       echo 'BŁĄD:'.mysqli_connect_error();
   else
   {
       $zapytanie="SELECT
produkty.id,produkty.nazwa,produkty.opis,produkty.zdjecie FROM produkty WHERE
produkty.Rodzaje_id=12;";
       $wynik=mysqli_query($polaczenie,$zapytanie);
       if(!$wynik)
           echo 'BŁĄD ZAPYTANIA: '.mysqli_error($polaczenie);
       else
       {
           while($wiersz=mysqli_fetch_row($wynik))
               echo '<div id="film2"><h4>'.$wiersz[0].'. '.$wiersz[1].'</h4><img
src='".$wiersz[3]."' alt="film"><br>'.$wiersz[2].'</div>';
       }

   }


   mysqli_close($polaczenie);
   ?>

</div>
<div id="stopka">
<form action="video.php" method="post">
<label>Usuń film nr.:</label><input type="number" name="filmik">
<input type="submit" value="Usuń film">
</form>

    <?php
   if(isset($_POST['filmik']))
   {
   $polaczenie=mysqli_connect("localhost","root","","dane3");
   if(!$polaczenie)
       echo 'BŁĄD:'.mysqli_connect_error();
   else
```

```
    {
        $zapytanie='DELETE FROM produkty WHERE produkty.id='.$_POST['filmik'];
        $wynik=mysqli_query($polaczenie,$zapytanie);
        if(!$wynik)
            echo 'BŁĄD ZAPYTANIA: '.mysqli_error($polaczenie);


    }
        mysqli_close($polaczenie);
    }

   ?>


    <p>Stronę wykonał: <a href="mailto:ja@poczta.com">33333333333</a></p>
    </div>
</body>
</html>
```

PHP – „Portal Społecznościowy - panel administratora”

```
<div id="baner">
<h3>Portal Społecznościowy - panel administratora</h3>
</div>
<div id="lewy">
<h4>Użytkownicy</h4>
   <!-- Skrypt 1 -->
<?php
$polaczenie=mysqli_connect("localhost","root","","dane4");
$zapytanie="SELECT
osoby.id,osoby.imie,osoby.nazwisko,osoby.rok_urodzenia,osoby.zdjecie FROM
osoby LIMIT 0,30;";
$wynik=mysqli_query($polaczenie,$zapytanie);
   if(!$wynik)
      echo 'BŁAD ZAPYTANIA:'.mysqli_error($polaczenie);
   else
   {
      while($wiersz=mysqli_fetch_row($wynik))
      {
         $wiek=2021-$wiersz[3];
         echo $wiersz[0].'. '.$wiersz[1].' '.$wiersz[2].', '.$wiek.' lat<br>';
      }

   }
   mysqli_close($polaczenie);
?>

<a href="settings.html">Inne ustawienia</a>
</div>
<div id="prawy">
<h4>Podaj id użytkownika</h4>
<form action="users.php" method="post">
<input type="number" name="liczba">  
<input type="submit" value="ZOBACZ" id="przycisk">
</form>
<hr>
<!-- Skrypt 2 -->
<?php
$polaczenie=mysqli_connect("localhost","root","","dane4");
if(isset($_POST['liczba']))
{
$zapytanie='SELECT
osoby.imie,osoby.nazwisko,osoby.rok_urodzenia,osoby.opis,osoby.zdjecie,hobby.na
zwa FROM osoby JOIN hobby ON osoby.Hobby_id=hobby.id WHERE
osoby.id='.$_POST['liczba'];
$wynik=mysqli_query($polaczenie,$zapytanie);
   if(!$wynik)
      echo 'BŁAD ZAPYTANIA:'.mysqli_error($polaczenie);
```

```
    else
    {
      while($wiersz=mysqli_fetch_row($wynik))
          echo '<h2>'.$_POST['liczba'].'. '.$wiersz[0].' '.$wiersz[1].'</h2><img
src="'.$wiersz[4].
'" alt="'.$_POST['liczba'].'"><p>Rok urodzenia: '.$wiersz[2].'</p><p>Opis:
'.$wiersz[3].'</p><p>Hobby: '.$wiersz[5].'</p>';


    }
}

    mysqli_close($polaczenie);
?>
</div>
<div id="stopka">
<p>Stronę wykonał: 25555555555</p>
</div>
</body>
```