|  |  |  |
| --- | --- | --- |
| **Laboratorium Projektowania Aplikacji Internetowych 1**  Wydział Elektrotechniki Automatyki i Informatyki  Politechnika Świętokrzyska | | |
| Studia: **Stacjonarne I stopnia** | Kierunek: **Informatyka** | |
| Data wykonania:  **21.11.2017** | Zespół: **06** | Grupa: **3ID13B** |
| Godzina wykonania:  **8:00 – 9:30** | Skład zespołu:   1. **Bartłomiej Osak** 2. **Tomasz Pasternak** | |
| Numer laboratorium: | Temat ćwiczenia: | |
| **3** | **HTML, Formularze, Arkusze stylów CSS,**  **PHP wprowadzenie** | |

1. **Korzystając z zewnętrznych arkuszy styli (bez użycia tabel) przygotować następujący szablon strony:**
2. **Strona zawiera pięć sekcji:**

* **Nagłówek.**

**Kod w HTML:**

<header>

<h3>PAI LABORATORIUM 3</h3>

<h2 class="signature">Bartłomiej Osak, Tomasz Pasternak</h2>

<h2 class="signature">Grupa: 6 Grupa dziekańska: 3ID13B</h2>

</header>

**Opis:**

W celu stworzenia sekcji nagłówka posłużyliśmy się znacznikami header. W nich użyliśmy znaczników nagłówków h3 oraz h2, w których kolejno wpisaliśmy numer laboratorium, imiona i nazwiska członków grupy oraz numer grupy dziekańskiej.

* **Menu kontekstowe z lewej strony.**

**Kod w HTML:**

<div id="leftmenu">

<ul>

<li>

<a href="#">STRONA GŁÓWNA</a>

</li>

<li>

<a href="#">PODSTRONA 1</a>

</li>

<li>

<a href="#">PODSTRONA 2</a>

</li>

<li>

<a href="#">PODSTRONA 3</a>

</li>

</ul>

</div>

**Opis:**

W celu stworzenia menu kontekstowego z lewej strony posłużyliśmy znacznikiem div, który tworzy określony blok. W tym bloku stworzyliśmy listę za pomocą znaczników ul. Kolejne miejsca listy określaliśmy za pomocą znacznika li. W punktach listy umieszczaliśmy hiperłącza za pomocą znaczników a, lecz bez konkretnego przekierowania do podstron.

* **Menu kontekstowe u góry strony.**

**Kod w HTML:**

<div id="topmenu">

<ul>

<li atrybut="a b">

<a href="#">STRONA GŁÓWNA</a>

</li>

<li atrybut="b">

<a href="#">O NAS</a>

</li>

<li atrybut="a">

<a href="#">OFERTA</a>

</li>

<li atrybut="b">

<a href="#">KONTAKT</a></li></ul></div>

**Opis:**

W celu stworzenia menu kontekstowego u góry strony posłużyliśmy się znacznikiem div, który tworzy określony blok. W tym bloku stworzyliśmy listę za pomocą znaczników ul. Kolejne miejsca listy określaliśmy za pomocą znacznika li. W punktach listy umieszczaliśmy hiperłącza za pomocą znaczników a, lecz bez konkretnego przekierowania do podstron.

* **Stopka.**

**Kod w HTML:**

<footer>

<h2 class="signature">Bartłomiej Osak, Tomasz Pasternak</h2>

</footer>

**Opis:**

W celu stworzenia stopki strony posłużyliśmy się znacznikiem footer, który tworzą określoną stopkę strony. W stopce użyliśmy znacznika nagłówka h2, w którym zawarliśmy nasze imiona i nazwiska.

* **Zawartość strony.**

**Kod w HTML:**

<article>

<div id="content">

<p>Witamy na naszej stronie!

<br>Laboratorium 3</p>

<hr>

<p class="vertical">

<span>HTML 5.0 CSS 3.0 PHP 5</span>

</p>

<span>LABORATORIUM</span>

<p>Projektowanie Aplikacji Internetowych 1</p>

<span>PŚk 2017</span>

<hr>

<span>Formularz:</span>

<br>

<form id="php\_form" action="formularz.php" method="post">

<span class="form">IMIĘ:

<input type="text" name="imie" placeholder=" Wprowadź imię">

</span>

<br>

<span class="form">NAZWISKO:

<input type="text" name="nazwisko" placeholder=" Wprowadź nazwisko">

</span>

<br>

<span class="form">WIEK:

<input type="text" name="wiek" placeholder=" Wprowadź wiek">

</span>

<br>

<span class="form">KONTAKT:

<input type="text" name="kontakt" placeholder=" Wprowadź kontakt">

</span>

<br>

<span class="form">PŁEĆ:

<input type="radio" name="plec" value="k"> Kobieta

<input type="radio" name="plec" value="m"> Mężczyzna

</span>

<br>

<span class="form">UCZELNIA:

<select name="uczelnia">

<option value="ujk">UJK</option>

<option value="psk">PŚk</option>

<option value="wseip">WSEPiNM</option>

</select>

</span>

<br>

<span class="form">O TOBIE:

<textarea name="opis" placeholder=" Napisz coś o sobie"></textarea>

</span>

<br>

<span class="form">

<input type="submit" value="Wyślij">

</span>

</form>

</div>

</article>

**Opis:**

Celem stworzenia sekcji głównej zawartości strony posłużyliśmy się znacznikiem article. Zawartość strony umieszczona jest w bloku stworzonym za pomocą znacznika div. Ponadto w bloku tym używamy p – znacznik akapitu, span – znacznik grupowania oraz różnych typów danych wejściowych do formularza (formularz tworzyliśmy używając znacznika form), czyli znacznik input z różnymi wariantami takimi jak: text, radio oraz submit. Ponadto użyliśmy do formularza znacznika tworzącego wysuwającą listę wyboru, czyli znacznik select.

1. **W poszczególnych elementach strony zobrazować wykorzystanie:**

* **Selektory odwołujące się do atrybutów: atrybuty *class* oraz *id*, selektory typu [x], [x=”y”] oraz [x~=”y”].**

**Przykłady użycia:**

* **Przykładowy atrybut class:**

**HTML:**

<span class="form">IMIĘ:

<input type="text" name="imie" placeholder=" Wprowadź imię">

</span>

**CSS:**

.form {

font-size: 15px;

font-weight: bold;

}

**Zobrazowanie:**



**Opis:**

Dla klasy o nazwie form, która użyta jest dla znacznika span zdefiniowaliśmy w arkuszach styli wielkość fonta oraz jego pogrubienie za pomocą elementu stylu font-size oraz font-weight. Ponadto odwoływanie się do atrybutów za pomocą class do elementu akapitu (znacznik p) oraz do sygnatur w stopce oraz nagłówku.

* **Przykładowy atrybut id:**

**HTML:**

<div id="topmenu">

…

</div>

**CSS:**

#topmenu {

display: inline-block;

}

**Zobrazowanie:**



**Opis:**

Na stronie wielokrotnie odwołujemy się do elementów za pomocą atrybutu id. W tym przypadku obrazujemy odwołanie się za pomocą id do bloku tworzącego menu kontekstowe u góry strony. W CSS definiujemy wyświetlanie tego elementu za pomocą znacznika stylu display z wartości inline-block, co oznacza wyświetlenie elementu strony w formie bloku z wymuszeniem wyświetlenia poziomego.

* **Przykładowe użycie selektora typu [x]:**

**HTML:**

<header>

…

</header>

**CSS:**

header {

border: 2px solid rgb(37, 36, 100);

font-family: "Tahoma";

padding: 5px;

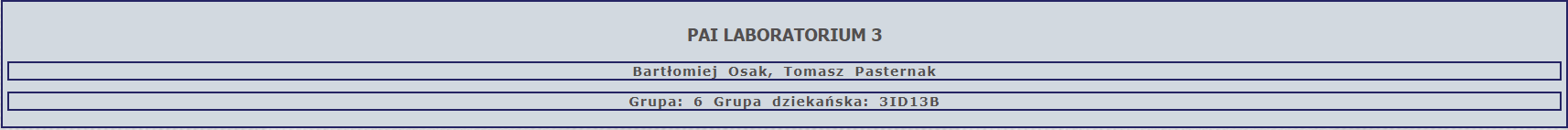
text-align: center;

color: rgb(83, 80, 80);

background: rgb(210, 217, 224);

}

**Zobrazowanie:**



**Opis:**

W tym przykładzie prezentujemy użycie selektora typu [x] na podstawie użycia go w przypadku znacznika header określającego nagłówek strony. W arkuszach styli definiujemy jego wygląd za pomocą elementów border – obramowanie, font-family – nazwa fonta, padding – dopełnienie wewnętrzne, text-align – pozycja tekstu, color – kolor tekstu oraz background – tło bloku.

* **Przykładowe użycie selektora typu [x=”y”]:**

**HTML:**

<li atrybut="b">

<a href="#">O NAS</a>

</li>

<li atrybut="b">

<a href="#">KONTAKT</a>

</li>

**CSS:**

#topmenu ul li[atrybut="b"] {

background: rgb(225, 233, 236);

}

**Zobrazowanie:**



**Opis:**

W tym przykładzie za pomocą selektora typu [x=”y”] definiujemy wygląd elementów li o atrybucie równym b, czyli jest to pole KONTAKT oraz O NAS w menu kontekstowym u góry strony. Ich tło definiowane jest za pomocą elementu stylu background. Ten styl zadziała tylko i wyłącznie dla elementu, w którym pole atrybut posiada tylko i wyłącznie literkę „b”.

* **Przykładowe użycie selektora typu [x~=”y”]:**

**HTML:**

<li atrybut="a b">

<a href="#">STRONA GŁÓWNA</a>

</li>

<li atrybut="a">

<a href="#">OFERTA</a>

</li>

**CSS:**

#topmenu ul li[atrybut~="a"] {

background: rgb(173, 178, 180);

}

**Zobrazowanie:**



**Opis:**

W tym przykładzie za pomocą selektora typu [x~=”y”] definiujemy wygląd elementów li o atrybucie, w którym wystąpi znak „b”, nawet pomiędzy innymi znakami. Są to elementy w górnym menu kontekstowym STRONA GŁÓWNA oraz OFERTA. W arkuszach styli ich wygląd definiowany jest za pomocą elementu background.

* **Selektory kontekstowe E F, E>F, E+F.**
* **Przykładowe zastosowanie selektora kontekstowego E F:**

**HTML:**

<span class="form">IMIĘ:

<input type="text" name="imie" placeholder=" Wprowadź imię">

</span>

**CSS:**

#php\_form span input {

    border-radius: 10px;

}

**Zobrazowanie:**



**Opis:**

Za pomocą selektora kontekstowego podstawowego E F, którego używamy wielokrotnie, w tym przypadku definiujemy wygląd elementu wprowadzającego input w formularzu. W arkuszu styli nadajemy mu określony wygląd za pomocą elementu stylu border-radius.

* **Przykładowe zastosowanie selektora kontekstowego E>F:**

**HTML:**

<div id="content">

……

<hr>

</div>

**CSS:**

#content>hr {

border: 1px solid rgb(37, 36, 100);

display: block;

}

**Zobrazowanie:**



**Opis:**

Za pomocą selektora kontekstowego E>F definiujemy wygląd linii w bloku content, który określony jest za pomocą znacznika div – linia w tym przypadku jest dzieckiem bloku content. Linię tą definiujemy za pomocą elementu styli: border – obramowanie na wartość 1px oraz określony kolor oraz display – sposób wyświetlania elementu blokowego.

* **Przykładowe zastosowanie selektora kontekstowego E+F:**

**HTML:**

<div id="content">

<p class="vertical">

<span>html 5.0 css 3.0 php 5</span>

</p>

<span>LABORATORIUM</span>

<p>Projektowanie Aplikacji Internetowych 1</p>

<span>PŚk 2017</span>

…

</div>

**CSS:**

#content p+span {

background: rgb(151, 163, 163);

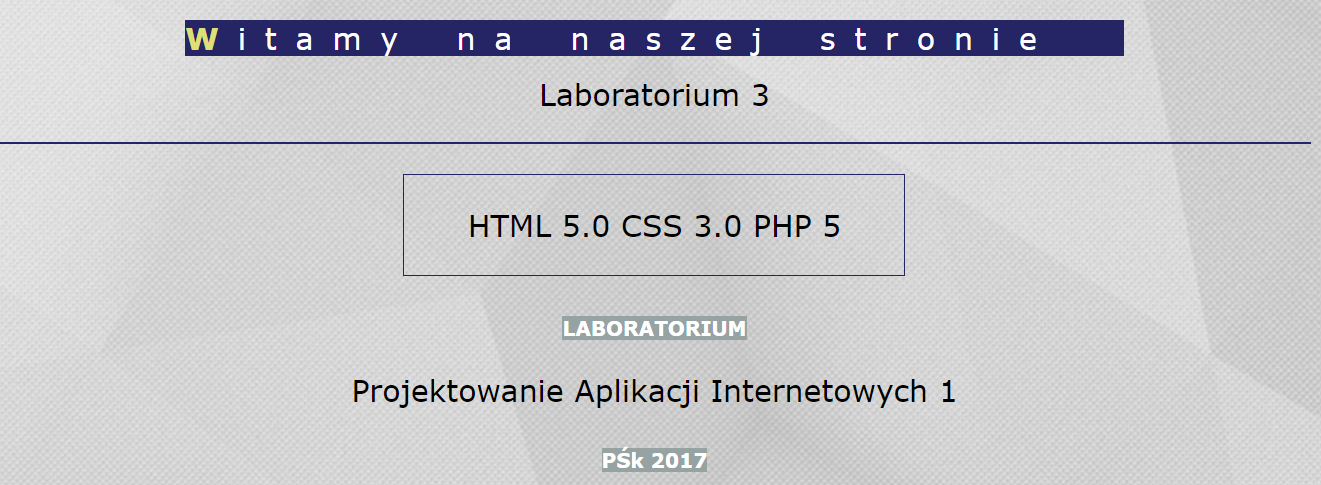
color: rgb(255,255,255);

font-weight: bold;

font-size: 20px;

}

**Zobrazowanie:**



**Opis:**

Za pomocą selektora kontekstowego E+F definiujemy wygląd elementu F który następuje bezpośrednio po elemencie E, czyli są to sąsiedzi. W naszym przypadku tego selektora użyliśmy do elementu span w bloku content. W arkuszach styli definiujemy nadając mu tło szare, color fonta, pogrubienie fonta oraz wielkość fonta. Style te dotyczą tylko i wyłącznie elementów span, które następują bezpośrednio po elemencie akapitu p.

* **Elementy div, span.**
* **Przykładowe wykorzystanie elementu div:**

**HTML:**

<div id="block">

…

</div>

**CSS:**

#block {

position: fixed;

left: 20px;

top: 20px;

background: black;

color: white;

width: 50px;

height: 50px;

font-size: 20px;

font-family: Tahoma;

line-height: 50px;

text-align: center;

    border-radius: 30px;

}

**Zobrazowanie:**



**Opis:**

Elementy typu div są bardzo często wykorzystywane na naszej stronie. Tutaj prezentujemy div, który tworzy nam blok z czarnym kółkiem z napisem PAI. Jego wygląd definiowany jest w arkuszach styli za pomocą elementów m.in. position, left, right,top – położenie elementu, border-radius, font-size, itp. -wygląd elementu.

* **Przykładowe wykorzystanie elementu span:**

**HTML:**

<p class="vertical">

<span>html 5.0 css 3.0 php 5</span>

</p>

**CSS:**

.vertical span {

display: table-cell;

vertical-align: middle;

text-transform: uppercase;

}

**Zobrazowanie:**



**Opis:**

Elementy typu span są bardzo często wykorzystywane na naszej stronie. Tutaj prezentujemy span, który zawarty jest w znaczniku p – znacznik nowego akapitu. Ten element span jest definiowany za pomocą klasy arkuszu CSS o nazwie vertical. W arkuszach styli określamy jego wyświetlanie – traktowany jest jak komórki tablicy, środkujemy jego położenie oraz transformujemy jego zawartość do dużych liter.

* **Ramki: padding, border, margin; rozmiary ramek (rozmiary minimalne i maksymalne), przepełnienie (własność overflow).**
* **Przykładowe zastosowanie ramek: padding, border, margin, rozmiary minimalne:**

**HTML:**

<footer>

<h2 class="signature">Bartłomiej Osak, Tomasz Pasternak</h2>

</footer>

**CSS:**

footer {

min-width: 1100px;

min-height: 80px;

text-align: center;

padding: 10px;

display: block;

margin-top: 20px;

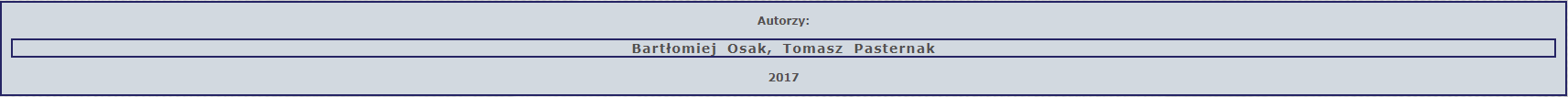
border: 2px solid rgb(37, 36, 100);

background: rgb(210, 217, 224);

color: rgb(83, 80, 80);

}

**Zobrazowanie:**



**Opis:**

W tym przypadku prezentujemy zastosowanie ramek padding, border oraz margin i ich rozmiary minimalne dla sekcji footer w html, która określa stopkę strony. Dokładne wartości poszczególnych ramek są określone w arkuszach styli za pomocą elementów: padding, border, margin. Rozmiar minimalny określany jest za pomocą elementu min-width oraz min-height.

* **Przykładowe zastosowanie rozmiaru minimalnego ramki:**

**HTML:**

<div id="leftmenu">

…

</div>

**CSS:**

#leftmenu {

max-width: 300px;

min-height: 600px;

float: left;

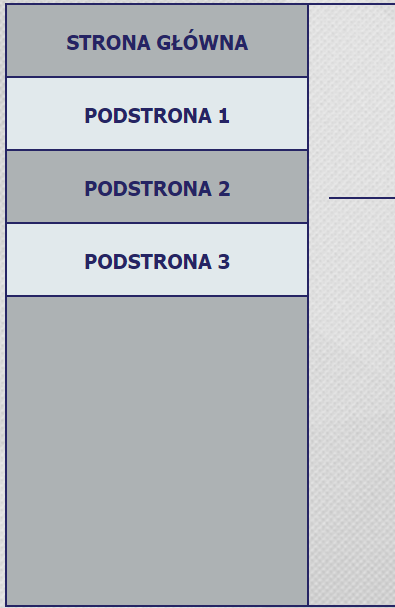
border: 2px solid rgb(37, 36, 100);

display: block;

background: rgb(173, 178, 180);

}

**Zobrazowanie:**



**Opis:**

Za pomocą max-width określiliśmy maksymalną szerokość lewego menu kontekstowego strony, tak, aby strona po przeskalowaniu okna była nadal nie naruszona.

* **Przykładowe zastosowanie przepełnienia:**

**HTML:**

<div id="content">

…

</div>

**CSS:**

#content {

margin-left: 300px;

padding: 20px;

min-width: 452px;

height: 520px;

overflow: auto;

font-family: Verdana;

text-align: center;

white-space: normal;

font-size: 30px;

}

**Zobrazowanie:**



**Opis:**

Za pomocą elementu styli overflow możemy zdefiniować w jakiej sytuacji ma ukazać się na stronie pasek przesuwający (scrollbar). Pojawia się on wtedy, gdy zawartość strony przekroczy zadaną wartość max-height dla bloku content.

* **Właściwości tekstu: wyrównanie (w pionie i poziomie), odstępy pomiędzy liniami, słowami oraz znakami, transformacja do dużych i małych liter.**
* **Przykład użycia: odstępy pomiędzy liniami, słowami oraz znakami, transformacja do małych liter:**

**HTML:**

<div id="content">

<p>WITAMY NA NASZEJ STRONIE

<br>Laboratorium 3</p>

…

</div>

**CSS:**

#content p:first-child::first-line {

background: rgb(37, 36, 100);

color: rgb(255,255,255);

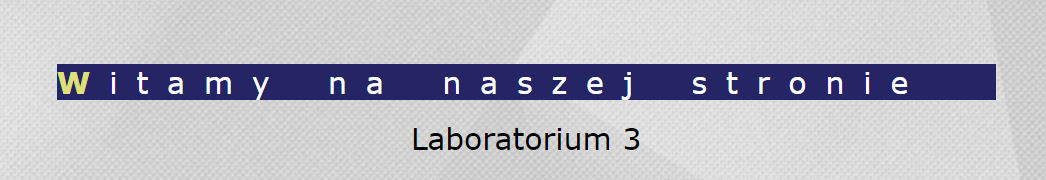
letter-spacing: 5mm;

word-spacing: 10px;

line-height: 75px;

text-transform: lowercase;}

**Zobrazowanie:**



**Opis:**

Odstęp pomiędzy liniami, słowami oraz znakami zobrazowaliśmy dla pierwszej linii w znaczniku p dla bloku content. Ponadto używamy tu transformacji tekstu w pierwszej linii do dużych znaków. Odstępy realizujemy w arkuszach styli stosując kolejno: letter-spacing – odstępy pomiędzy znakami, word-spacing – odstępy pomiędzy słowami, line-height – odstęp pomiędzy liniami. Transformację do małych znaków realizujemy poprzez element styli text-transform: lowercase.

* **Przykład użycia: wyrównywanie w pionie oraz transformacja do dużych liter:**

**HTML:**

<div id="content">

…

<p class="vertical">

<span>html 5.0 css 3.0 php 5</span>

</p>

…

</div>

**CSS:**

.vertical span {

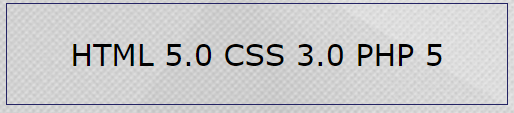
display: table-cell;

vertical-align: middle;

text-transform: uppercase;

}

**Zobrazowanie:**



**Opis:**

Transformację do dużych liter realizujemy w arkuszach styli za pomocą elementu text-transform:uppercase. Ponadto mechanizm wyrównywania w pionie realizujemy stosując do elementu span określanego przez klasę vertical elementy styli tj. display: table-cell – wyświetlanie elementu jako komórek tablicy, vertical-align: wyrównanie wertykalne do środka.

* **Przykład użycia: wyrównywanie w poziomie:**

**HTML:**

<header>

<h3>PAI LABORATORIUM 3</h3>

<h2 class="signature">Bartłomiej Osak, Tomasz Pasternak</h2>

<h2 class="signature">Grupa: 6 Grupa dziekańska: 3ID13B</h2>

</header>

**CSS:**

header {

border: 2px solid rgb(37, 36, 100);

font-family: "Tahoma";

padding: 5px;

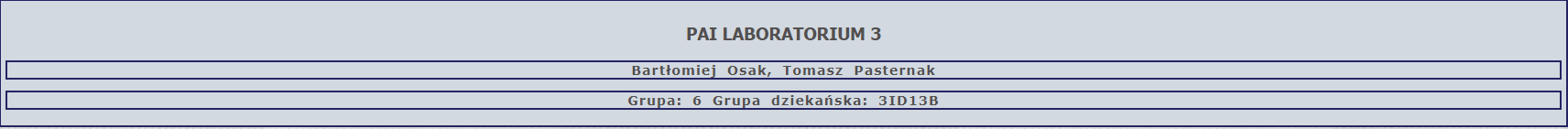
text-align: center;

color: rgb(83, 80, 80);

background: rgb(210, 217, 224);

}

**Zobrazowanie:**



**Opis:**

Wyrównywanie tekstu w poziomie realizujemy poprzez element stylu o nazwie text-align z wartością center. W naszym przypadku stosujemy ten element dla nagłówka określonego znacznikiem header.

* **Właściwości tła: kolor, rysunek (ilość, położenie, krotność).**

**HTML:**

<body>

…

</body>

**CSS:**

body {

background-color: #fff;

background-image: url("tlo.jpg");

background-repeat: repeat;

margin: 0 !important;

}

**Zobrazowanie:**



**Opis:**

Tło personalizujemy w sekcji body. W arkuszach styli za pomocą kilku elementów realizujemy określenie koloru tła – background: #fff, na które nakładany jest rysunek tlo.jpg. Krotność rysunku określamy za pomocą elementu background-repeat a jego położenie za pomocą elementu margin z adnotacją important.

Ponadto sam element background wykorzystywany jest wielokrotnie na naszej stronie w sekcjach lewego menu, górnego menu, w stopce oraz w nagłówku.

* **Pseudoklasy: :first-child:, :hover, :active. Tworzenie odnośnika do innych dokumentów zmieniającego swój wygląd po najechaniu myszką. Utworzyć formularz, którego pola będą wyróżniane po wybraniu.**
* **Przykładowe zastosowanie :first-child:**

**HTML:**

<div id="content">

<p>WITAMY NA NASZEJ STRONIE

<br>Laboratorium 3</p>

</div>

**CSS:**

#content p:first-child::first-letter {

font-weight: bold;

text-transform: uppercase;

color: rgb(223, 223, 119);

}

**Zobrazowanie:**



**Opis:**

Element :first-child zastosowaliśmy w bloku content. Zadziała on tylko dla pierwszego p, czyli dla pierwszego dziecka typu p dla bloku content. Wygląd określamy w arkuszu styli za pomocą elementów określających styl fonta, transformujemy litery do dużych oraz nadajemy kolor literom.

* **Przykładowe zastosowanie :hover:**

**HTML:**

<div id="topmenu">

<ul>

<li atrybut="a b">

<a href="#">STRONA GŁÓWNA</a>

</li>

<li atrybut="b">

<a href="#">O NAS</a>

</li>

<li atrybut="a">

<a href="#">OFERTA</a>

</li>

<li atrybut="b">

<a href="#">KONTAKT</a>

</li>

</ul>

</div>

**CSS:**

#topmenu ul li a:hover {

background: rgb(37, 36, 100);

color: rgb(255, 255, 255);

}

**Zobrazowanie:**

****

**Opis:**

Zgodnie z poleceniem stworzyliśmy odnośniki do innych dokumentów. Najeżdżając na pola na górnym lub dolnym menu kontekstowym pole zmienia kolor na granatowy a znaki na biały. Realizowane jest to za pomocą elementu :hover, który zastosowany jest do elementu html, który tworzy hiperłącze.

* **Przykładowe zastosowanie :active:**

**HTML:**

<form id="php\_form" action="formularz.php" method="post">

<span class="form">IMIĘ:

<input type="text" name="imie" placeholder=" Wprowadź imię">

</span>  
…

</form>

**CSS:**

#php\_form span input:active,

input:focus {

background: rgb(37, 36, 100);

color: rgb(255,255,255);

font-weight: bold;

font-size: 15px;

}

**Zobrazowanie:**



**Opis:**

Zgodnie z poleceniem stworzyliśmy prosty formularz za pomocą znacznika form. Ponadto do elementu input, który znajduje się w elemencie span użyliśmy elementu stylu :active. Jego zastosowanie jest widoczne po kliknięciu w dane pole formularza (tu: imię). Wtedy z pola białego zmieni się na pole granatowe oraz kolor fonta wpisywanego wyrazu zmieniony został z czarnego na biały.

* **Pseudoelementy :first-line:, :first-letter:. Teksty generowane automatycznie :before, :after.**
* **Przykładowe zastosowanie pseudoelementu :first-line:**

**HTML:**

<div id="content">

<p>WITAMY NA NASZEJ STRONIE

<br>Laboratorium 3</p>

…

</div>

**CSS:**

#content p:first-child::first-line {

background: rgb(37, 36, 100);

color: rgb(255,255,255);

letter-spacing: 5mm;

word-spacing: 10px;

line-height: 75px;

text-transform: lowercase;

}

**Zobrazowanie:**



**Opis:**

Pseudoelement :first-line został użyty do zdefiniowania pierwszej linii bloku contentu. Element ten odwołuje się bezpośrednio do tekstu tylko i wyłącznie zawartego w znaczniku p – znacznik akapitu. W arkuszach styli zdefiniowaliśmy tło, kolor fonta, przerwy pomiędzy literami, przerwy pomiędzy wyrazami, przerwy pomiędzy liniami oraz zastosowaliśmy transformację do małych liter.

* **Przykładowe zastosowanie pseudoelementu :first-letter:**

**HTML:**

<div id="content">

<p>WITAMY NA NASZEJ STRONIE

<br>Laboratorium 3</p>

…

</div>

**CSS:**

#content p:first-child::first-letter {

font-weight: bold;

text-transform: uppercase;

color: rgb(223, 223, 119);

}

**Zobrazowanie:**



**Opis:**

Pseudoelement :first-letter został użyty do zdefiniowania pierwszego znaku w pierwszej linii bloku content. Element ten odwołuje się tylko i wyłącznie do pierwszej litery tekstu zawartego w znaczniku p – znacznik akapitu. Użycie tego znacznika jest tutaj nieprzypadkowe. W punkcie poprzednim stosując znacznik :first-line zmieniliśmy litery na małe. Chcąc, aby nasze zdane powitalne na stronie zaczynało się od wielkich liter użyliśmy pseudoelementu :first-letter, gdzie w arkuszach styli realizujemy transformację do dużych liter oraz zmieniamy kolor tej literki na żółty. Ponadto litera jest pogrubiona poprzez zastosowanie elementu font-weight.

* **Przykładowe zastosowanie tekstów generowanych automatycznie: :after oraz :before:**

**HTML:**

<footer>

<h2 class="signature">Bartłomiej Osak, Tomasz Pasternak</h2>

</footer>

**CSS:**

footer::before {

content: "Autorzy:";

font-family: Verdana;

font-weight: bold;

font-size: 10px;

}

footer::after {

content: "2017";

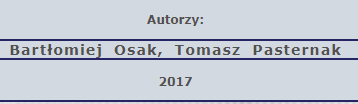
font-family: Verdana;

font-weight: bold;

font-size: 10px;

}

**Zobrazowanie:**



**Opis:**

Teksty generowane automatycznie zastosowaliśmy do pola stopki naszej strony. W stopce domyślnie znajdują się nasze imiona oraz nazwiska. Stosując elementy :after oraz :before możemy dopisać treść, która będzie generowana automatycznie za każdym razem – nie wpisując jej do pliku html, lecz do pliku css. Element :after generuje u nas rok powstania strony, a element :before generuje napis „Autorzy”. Teksty wygenerowane okalają więc nasze imiona i nazwiska. Elementy te zostały określone w arkuszu stylu – oba tak samo. Ustawiliśmy tam rodzaj fonta, jego pogrubienie oraz wielkość.

* **Dziedziczenie: słowo kluczowe inherit.**

**HTML:**

<header>

<h3>PAI LABORATORIUM 3</h3>

<h2 class="signature">Bartłomiej Osak, Tomasz Pasternak</h2>

<h2 class="signature">Grupa: 6 Grupa dziekańska: 3ID13B</h2>

</header>

**CSS:**

.signature {

font-family: Verdana;

font-size: 14px;

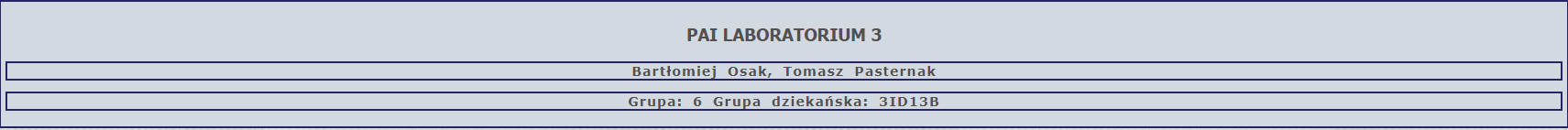
letter-spacing: 0.3mm;

word-spacing: 5px;

border: inherit;

}

**Zobrazowanie:**



**Opis:**

Mechanizm dziedziczenia słowem kluczowym inherit wykorzystaliśmy w przypadku nagłówka oraz stopki. W obu przypadkach dziedziczone jest obramowanie z obszaru zewnętrznego – nagłówek jest obramowany ramką. Następnie tworzymy dwa znaczniki h2 określone klasą, w której dziedziczone jest obramowanie bezpośrednio od tego, które jest użyte w znaczniku header.

* **Pozycjonowanie: własność position.**

**HTML:**

<div id="block">

<span>PAI</span>

</div>

**CSS:**

#block {

position: fixed;

left: 20px;

top: 20px;

background: black;

color: white;

width: 50px;

height: 50px;

font-size: 20px;

font-family: Tahoma;

line-height: 50px;

text-align: center;

    border-radius: 30px;

}

**Zobrazowanie:**



**Opis:**

Użycie własności pozycjonowania danego elementu – position użyliśmy do bloku, który zawiera koło z napisem. Jego pozycja jest niezależna od pozycji pozostałej treści strony. Ustawienie tej własności dokonaliśmy w pliku arkuszy styli nadając elementowi position wartość fixed.

* **Białe spacje.**

**HTML:**

<div id="content">

…

</div>

**CSS:**

#content {

margin-left: 300px;

padding: 20px;

min-width: 452px;

height: 520px;

overflow: auto;

font-family: Verdana;

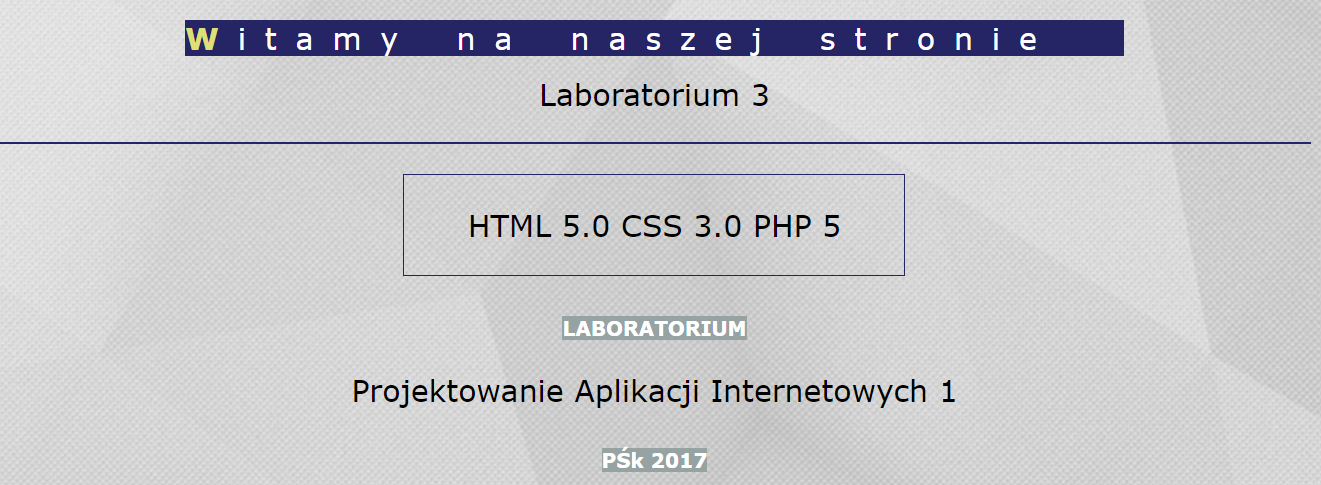
text-align: center;

white-space: normal;

font-size: 30px;

}

**Zobrazowanie:**



**Opis:**

Białe spacje przetestowaliśmy dla bloku content, który skupia zawartość strony – teksty. Stosując różne ustawienia elementu white-space możemy zaobserwować różnicę w formatowaniu się wyświetlanego tekstu. W naszym przypadku zdecydowaliśmy na użycie ustawienia o wartości normal, w celu jak najładniejszego wyświetlania tekstu z białymi znakami.

* **Typy elementów: block, inline.**

**HTML:**

<div id="topmenu">

…

</div>

**CSS:**

#topmenu {

display: inline-block;

}

**Zobrazowanie:**



**Opis:**

Typy block oraz inline użyliśmy w przypadku poprawnego wyświetlania górnego menu kontekstowego.

W CSS definiujemy wyświetlanie tego elementu za pomocą znacznika stylu display z wartości inline-block, co oznacza wyświetlenie elementu strony w formie bloku z wymuszeniem wyświetlenia poziomego.

1. **Skonfigurować serwer Apacha do współpracy z PHP.**
2. **Napisać formularz zawierający elementy różnego typu – minimum 6 pól.**

**HTML:**

<form id="php\_form" action="formularz.php" method="post">

<span class="form">IMIĘ:

<input type="text" name="imie" placeholder=" Wprowadź imię">

</span>

<br>

<span class="form">NAZWISKO:

<input type="text" name="nazwisko" placeholder=" Wprowadź nazwisko">

</span>

<br>

<span class="form">WIEK:

<input type="text" name="wiek" placeholder=" Wprowadź wiek">

</span>

<br>

<span class="form">KONTAKT:

<input type="text" name="kontakt" placeholder=" Wprowadź kontakt">

</span>

<br>

<span class="form">PŁEĆ:

<input type="radio" name="plec" value="k"> Kobieta

<input type="radio" name="plec" value="m"> Mężczyzna

</span>

<br>

<span class="form">UCZELNIA:

<select name="uczelnia">

<option value="ujk">UJK</option>

<option value="psk">PŚk</option>

<option value="wseip">WSEPiNM</option>

</select>

</span>

<br>

<span class="form">O TOBIE:

<textarea name="opis" placeholder=" Napisz coś o sobie"></textarea>

</span>

<br>

<span class="form">

<input type="submit" value="Wyślij">

</span>

</form>

**CSS:**

.form {

font-size: 15px;

font-weight: bold;

}

#php\_form span input:active,

input:focus {

background: rgb(37, 36, 100);

color: rgb(255,255,255);

font-weight: bold;

font-size: 15px;

}

#php\_form span input {

    border-radius: 10px;

}

#php\_form span textarea:active,

textarea:focus {

background: rgb(37, 36, 100);

color: rgb(255,255,255);

font-weight: bold;

font-size: 15px;

}

#php\_form span textarea {

    border-radius: 10px;

}

#php\_form span select:active,

select:focus {

background: rgb(37, 36, 100);

color: rgb(255,255,255);

font-weight: bold;

font-size: 15px;

}

#php\_form span input[type="submit"] {

background: rgb(37, 36, 100);

color: rgb(255,255,255);

font-family: Verdana;

font-size: 20px;

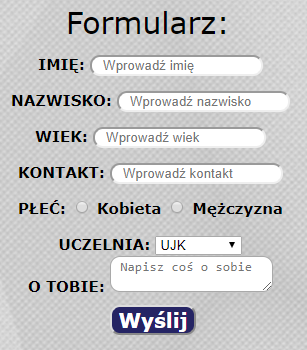
font-weight: bold;

    border-radius: 10px;

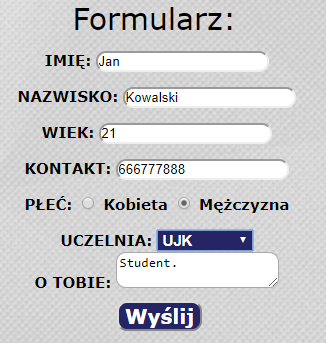
}

**Zobrazowanie:**

* **Pusty formularz:**



* **Formularz uzupełniany:**



* **Formularz gotowy do wysłania:**



**Opis:**

Formularz tworzyliśmy używając znacznika form w bloku article. Formularz zawiera pola input – pola danych wejściowych różnego typu. Dla imienia, nazwiska, wieku oraz kontaktu są to inputy o type tekstowym. Dla wyboru płci zastosowaliśmy input typu radio. Do wyboru uczelni zastosowaliśmy listę wysuwaną z różnymi pozycjami do wyboru za pomocą znacznika select. Ostatnim element jest pole, w którym możemy opisać się – jest to określone za pomocą textarea.

1. **Napisać stronę obsługi formularza.**

**PHP:**

<head>

<meta charset="utf-8">

<title>Bartłomiej Osak - Tomasz Pasternak - 3ID13B</title>

</head>

<?php

if ($\_POST) {

$imie = $\_POST['imie'];

$nazwisko = $\_POST['nazwisko'];

$wiek = $\_POST['wiek'];

$kontakt = $\_POST['kontakt'];

$plec = $\_POST['plec'];

$uczelnia = $\_POST['uczelnia'];

$opis = htmlspecialchars(trim($\_POST['opis']));

if ($imie && $nazwisko && $wiek && $kontakt && $plec && $uczelnia && $opis) {

echo '<b>Imie: </b>' . $imie . '<br>';

echo '<b>Nazwisko: </b>' . $nazwisko . '<br>';

echo '<b>Wiek: </b>' . $wiek . '<br>';

echo '<b>Kontakt: </b>' . $kontakt . '<br>';

if ($plec == 'k') {

echo '<b>Płeć:</b> kobieta<br>';

} else {

echo '<b>Płeć:</b> mężczyzna<br>';

}

if ($uczelnia == 'ujk') {

echo '<b>Uczelnia: UJK Kielce </b><br>';

} else if ($uczelnia == 'psk') {

echo '<b>Uczelnia: </b>PŚk Kielce<br>';

} else {

echo '<b>Uczelnia: </b>WSEPiNM Kielce<br>';

}

echo '<b>Coś o sobie: </b>' . $opis . '<br>';

} else {

echo '<b>Wypełnij wszystkie pola formularza!</b>';

}

}

if ($\_GET)

{

$imie = $\_GET['imie'];

$nazwisko = $\_GET['nazwisko'];

$wiek = $\_GET['wiek'];

$kontakt = $\_GET['kontakt'];

$plec = $\_GET['plec'];

$uczelnia = $\_GET['uczelnia'];

$opis = htmlspecialchars(trim($\_GET['opis']));

if ($imie && $nazwisko && $wiek && $kontakt && $plec && $uczelnia && $opis) {

echo '<b>Imie: </b>' . $imie . '<br>';

echo '<b>Nazwisko: </b>' . $nazwisko . '<br>';

echo '<b>Wiek: </b>' . $wiek . '<br>';

echo '<b>Kontakt: </b>' . $kontakt . '<br>';

if ($plec == 'k') {

echo '<b>Płeć:</b> kobieta <br>';

} else {

echo '<b>Płeć:</b> mężczyzna <br>';

}

if ($uczelnia == 'ujk') {

echo '<b>Uczelnia:</b> UJK Kielce <br>';

} else if ($uczelnia == 'psk') {

echo '<b>Uczelnia:</b> PŚk Kielce <br>';

} else {

echo '<b>Uczelnia:</b> WSEPiNM Kielce <br>';

}

echo '<b>Coś o sobie:</b> ' . $opis . '<br>';

} else {

echo '<b>Wypełnij wszystkie pola formularza!</b>';

}

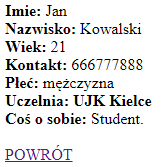
}

?>

<br>

<a href="index.html">POWRÓT</a>

**Zobrazowanie:**



**Opis:**

Po kliknięciu przycisku wyślij i uzupełnieniu wszystkich pól formularza następuje przekierowanie do drugiej strony, która wypisuje nam wszystkie dane, które wpisaliśmy w formularzu. Strona została napisana przy użyciu PHP. Celem wykonania zadania z ppkt c) stworzyliśmy dwie instrukcje warunkowe, które uruchamiają się w zależności od zmiany w kodzie strony www typu przesyłania danych – można wybrać GET oraz POST. W obu instrukcjach warunkowych w zależności od metody przesyłania pobieramy dane z formularza za pomocą $\_POST lub $\_GET. Wszystkie dane pobrane są zapisywane do zmiennych i wypisywane poleceniem echo, które zawiera znaczniki html na stronę. Ponadto na końcu istnieje hiperłącze, które umożliwia użytkownikowi powrót do strony głównej.

1. **Przećwiczyć przekazywanie danych między formularzami metodą POST oraz GET.**

* **dla POST:**

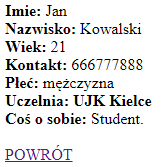
**Metoda wysyłania w HTML:**

<form id="php\_form" action="formularz.php" method="post">

**Pasek adresu:**



**Efekt zadziałania:**



* **dla GET:**

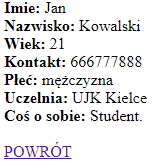
**Metoda wysyłania w HTML:**

<form id="php\_form" action="formularz.php" method="get">

**Pasek adresu:**



**Efekt zadziałania:**



**Opis:**

Dane metodą GET wysyłane są w sposób jawny w adresie skryptu, zaś dane wysyłane metodą POST są umieszczone w treści wiadomości http.

1. **Kody źródłowe strony:**

**HTML:**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<link rel="stylesheet" type="text/css" href="style.css">

<title>Bartłomiej Osak - Tomasz Pasternak - 3ID13B</title>

</head>

<body>

<div id="container">

<div id="block">

<span>PAI</span>

</div>

<div id="nav">

<div id="topmenu">

<ul>

<li atrybut="a b">

<a href="#">STRONA GŁÓWNA</a>

</li>

<li atrybut="b">

<a href="#">O NAS</a>

</li>

<li atrybut="a">

<a href="#">OFERTA</a>

</li>

<li atrybut="b">

<a href="#">KONTAKT</a>

</li>

</ul>

</div>

</div>

<header>

<h3>PAI LABORATORIUM 3</h3>

<h2 class="signature">Bartłomiej Osak, Tomasz Pasternak</h2>

<h2 class="signature">Grupa: 6 Grupa dziekańska: 3ID13B</h2>

</header>

<div id="leftmenu">

<ul>

<li>

<a href="#">STRONA GŁÓWNA</a>

</li>

<li>

<a href="#">PODSTRONA 1</a>

</li>

<li>

<a href="#">PODSTRONA 2</a>

</li>

<li>

<a href="#">PODSTRONA 3</a>

</li>

</ul>

</div>

<article>

<div id="content">

<p>WITAMY NA NASZEJ STRONIE

<br>Laboratorium 3</p>

<hr>

<p class="vertical">

<span>html 5.0 css 3.0 php 5</span>

</p>

<span>LABORATORIUM</span>

<p>Projektowanie Aplikacji Internetowych 1</p>

<span>PŚk 2017</span>

<hr>

<span>Formularz:</span>

<br>

<form id="php\_form" action="formularz.php" method="get">

<span class="form">IMIĘ:

<input type="text" name="imie" placeholder=" Wprowadź imię">

</span>

<br>

<span class="form">NAZWISKO:

<input type="text" name="nazwisko" placeholder=" Wprowadź nazwisko">

</span>

<br>

<span class="form">WIEK:

<input type="text" name="wiek" placeholder=" Wprowadź wiek">

</span>

<br>

<span class="form">KONTAKT:

<input type="text" name="kontakt" placeholder=" Wprowadź kontakt">

</span>

<br>

<span class="form">PŁEĆ:

<input type="radio" name="plec" value="k"> Kobieta

<input type="radio" name="plec" value="m"> Mężczyzna

</span>

<br>

<span class="form">UCZELNIA:

<select name="uczelnia">

<option value="ujk">UJK</option>

<option value="psk">PŚk</option>

<option value="wseip">WSEPiNM</option>

</select>

</span>

<br>

<span class="form">O TOBIE:

<textarea name="opis" placeholder=" Napisz coś o sobie"></textarea>

</span>

<br>

<span class="form">

<input type="submit" value="Wyślij">

</span>

</form>

</div>

</article>

<footer>

<h2 class="signature">Bartłomiej Osak, Tomasz Pasternak</h2>

</footer>

</div>

</body>

</html>

**CSS:**

body {

background-color: #fff;

background-image: url("tlo.jpg");

background-repeat: repeat;

margin: 0 !important;

}

header {

border: 2px solid rgb(37, 36, 100);

font-family: "Tahoma";

padding: 5px;

text-align: center;

color: rgb(83, 80, 80);

background: rgb(210, 217, 224);

}

.signature {

font-family: Verdana;

font-size: 14px;

letter-spacing: 0.3mm;

word-spacing: 5px;

border: inherit;

}

#container {

width: 90%;

min-width: 1100px;

min-height: 1000px;

margin-left: auto;

margin-right: auto;

}

#nav {

height: 75px;

min-width: 1100px;

text-align: center;

}

#topmenu {

display: inline-block;

}

#topmenu ul li[atrybut~="a"] {

background: rgb(173, 178, 180);

}

#topmenu ul li[atrybut="b"] {

background: rgb(225, 233, 236);

}

#topmenu ul,

#leftmenu ul {

font-family: "Tahoma";

font-weight: bold;

height: 75px;

font-size: 20px;

list-style-type: none;

margin: 0px;

padding: 0px;

}

#topmenu ul li {

display: block;

float: left;

width: 260px;

background: rgb(204, 213, 218);

}

#topmenu ul li a {

height: 71px;

text-decoration: none;

display: block;

line-height: 75px;

text-align: center;

border: 2px solid rgb(37, 36, 100);

color: rgb(37, 36, 100);

}

#topmenu ul li a:hover {

background: rgb(37, 36, 100);

color: rgb(255, 255, 255);

}

#leftmenu {

max-width: 300px;

min-height: 600px;

float: left;

border: 2px solid rgb(37, 36, 100);

display: block;

background: rgb(173, 178, 180);

}

#leftmenu ul li {

display: block;

width: 300px;

background: rgb(173, 178, 180);

float: left;

}

#leftmenu ul li:nth-of-type(2n) {

background: rgb(225, 233, 236);

}

#leftmenu ul li a {

height: 71px;

text-decoration: none;

display: block;

line-height: 75px;

text-align: center;

border-bottom: 2px solid rgb(37, 36, 100);

color: rgb(37, 36, 100);

}

#leftmenu ul li a:hover {

background: rgb(37, 36, 100);

color: rgb(255, 255, 255);

}

article,

#leftmenu {

display: table-cell;

vertical-align: top;

margin-top: 20px;

}

article {

min-width: 792px;

min-height: 600px;

border: 2px solid rgb(37, 36, 100);

display: block;

}

#content {

margin-left: 300px;

padding: 20px;

min-width: 452px;

height: 520px;

overflow: auto;

font-family: Verdana;

text-align: center;

white-space: normal;

font-size: 30px;

}

footer {

min-width: 1100px;

min-height: 80px;

text-align: center;

padding: 10px;

display: block;

margin-top: 20px;

border: 2px solid rgb(37, 36, 100);

background: rgb(210, 217, 224);

color: rgb(83, 80, 80);

}

footer::before {

content: "Autorzy:";

font-family: Verdana;

font-weight: bold;

font-size: 10px;

}

footer::after {

content: "2017";

font-family: Verdana;

font-weight: bold;

font-size: 10px;

}

#block {

position: fixed;

left: 20px;

top: 20px;

background: black;

color: white;

width: 50px;

height: 50px;

font-size: 20px;

font-family: Tahoma;

line-height: 50px;

text-align: center;

    border-radius: 30px;

}

#content p:first-child::first-line {

background: rgb(37, 36, 100);

color: rgb(255,255,255);

letter-spacing: 5mm;

word-spacing: 10px;

line-height: 75px;

text-transform: lowercase;

}

#content p:first-child::first-letter {

font-weight: bold;

text-transform: uppercase;

color: rgb(223, 223, 119);

}

#content p+span {

background: rgb(151, 163, 163);

color: rgb(255,255,255);

font-weight: bold;

font-size: 20px;

}

#content>hr {

border: 1px solid rgb(37, 36, 100);

display: block;

}

.vertical {

display: table;

margin-left: auto;

margin-right: auto;

width: 500px;

height: 100px;

border: 1px solid rgb(37, 36, 100);

}

.vertical span {

display: table-cell;

vertical-align: middle;

text-transform: uppercase;

}

.form {

font-size: 15px;

font-weight: bold;

}

#php\_form span input:active,

input:focus {

background: rgb(37, 36, 100);

color: rgb(255,255,255);

font-weight: bold;

font-size: 15px;

}

#php\_form span input {

    border-radius: 10px;

}

#php\_form span textarea:active,

textarea:focus {

background: rgb(37, 36, 100);

color: rgb(255,255,255);

font-weight: bold;

font-size: 15px;

}

#php\_form span textarea {

    border-radius: 10px;

}

#php\_form span select:active,

select:focus {

background: rgb(37, 36, 100);

color: rgb(255,255,255);

font-weight: bold;

font-size: 15px;

}

#php\_form span input[type="submit"] {

background: rgb(37, 36, 100);

color: rgb(255,255,255);

font-family: Verdana;

font-size: 20px;

font-weight: bold;

    border-radius: 10px;

}