

I. HTML. Wybrane tagi i parametry sekcji <body>

1. Formatowanie tekstu

Znaczniki

Znacznik	Opis
<!-- Komentarz -->	
 	Tekst umieszczony po tym znaczniku będzie wyświetlany w nowym wierszu
 	Przy pomocy tego znacznika można zdefiniować własności czcionki dla umieszczonego po nim tekstu (kolor, wielkość, rodzaj)
<hr>	W miejsce tego znacznika przeglądarka wstawi poziomą linię na szerokość okna
<h1> </h1> ... <h6> </h6>	Nagłówki
<p> </p>	Akapit
 	Tekst umieszczony wewnątrz tego znacznika będzie pogrubiony
<i> </i>	Tekst umieszczony wewnątrz tego znacznika będzie napisany czcionką pochyłą
<u> </u>	Tekst umieszczony wewnątrz tego znacznika będzie podkreślony
<blockquote> </blockquote>	Wcięcie akapitu

Atrybuty

Atrybut	Opis	Przykładowe wartości	Znaczniki
align	wyrównanie tekstu lub obiektu	left, right, center, justify	<p> <h1>...<h6>
background	rysunek stanowiący kolor tła strony	<i>nazwa_katalogu/nazwa_pliku</i>	<body>
bgcolor	kolor tła	angielskie nazwy kolorów (np. black, pink, white) lub nasycenie kolorami RGB (red, green, blue) w systemie szesnastkowym, np.: #rrggbb	<body>
color	kolor tekstu		
text	kolor tekstu		<body>
size	wielkość tekstu	wartość podawana np. w pikselach	

2. Listy numerowane i wypunktowane

 rozpoczyna listę numerowaną (oczywiście ją kończy)

 rozpoczyna listę wypunktowaną

 służy do tworzenia elementów obydwu list

3. Tabele

Znaczniki tabel

Znacznik	Opis
<table> </table>	Wewnątrz tego znacznika należy umieścić definicję tabeli
<tr> <th> <td>	Znaczniki te służą do formatowania komórek tabeli. Są umieszczone wewnątrz znacznika <table>. <tr> rozpoczyna nowy wiersz w tabeli <td> rozpoczyna nową komórkę <th> tzw. komórka nagłówkowa, która użyta zamiast <td> (zwykle w pierwszym wierszu tabeli) sprawia, że tekst jest wyróżniony

Atrybuty wykorzystywane w tabelach

Atrybut	Opis	Przykładowe wartości	Znaczniki
align	wyrównanie tabeli lub tekstu w komórce	left, right, center, justify	<table> <td> <th> <tr>
bgcolor	kolor tła tabeli lub komórki	angielskie nazwy (np. black, red, green) lub nasycenie kolorami RGB (kolejno red, green, blue) w systemie szesnastkowym, np.: #AB235C	<table> <td> <th> <tr>
bordercolor	kolor krawędzi	angielskie nazwy (np. black, red, green) lub nasycenie kolorami RGB (kolejno red, green, blue) w systemie szesnastkowym, np.: #AB235C	<table>
border	szerokość ramki	szerokość ramki tabeli w pikselach	<table>
frame	typ obramowania	void brak ramki border wszystkie krawędzie są widoczne	<table>
rules	typ krawędzi wewnątrz ramki	none żadna wewnętrzna krawędź nie jest widoczna all wszystkie krawędzie są widoczne rows widoczne tylko poziome krawędzie cols widoczne tylko pionowe krawędzie	<table>
colspan	łączenie komórek w wierszu	liczba całkowita określająca, ile kolumn ma zostać połączonych w wierszu	<td> <th>
rowspan	łączenie komórek w kolumnie	liczba całkowita określająca, ile wierszy ma zostać połączonych w kolumnie	<td> <th>

4. Encje

 	spacja nierozdzielająca
≈	znak równości przybliżonej (≈)
€ £	znaki walut, odpowiednio euro (€) i funta brytyjskiego (£)
♩ ♬	nuty, pierwszy z nich to ćwierćnuta, a drugi to dwie połączone szesnastki

5. Łąca

Łąca wewnętrzne

Tekst	łąca, które należy kliknąć
	wskaźnik

Łąca zewnętrzne

Tekst

Łąca do plików

Tekst

Odnosińki do adresów e-mailowych

Tekst

6. Formularze

Pola tekstowe (*text*)

```
<input type="text" size="liczba" maxlength="liczba" /><br>
```

Pola tekstowe z ukrytym tekstem (*password*)

```
Hasło:<input type="password" size="liczba"/><br>
```

Duże pola tekstowe (*textarea*)

```
<textarea rows="liczba" cols="liczba">Tekst</textarea>
```

Pola wyboru (*checkbox*)

```
<input type="checkbox" checked />Tekst<br>
```

Grupy opcji (*radio*)

```
<input type="radio" value="abc" />Tekst
```

Listy wyboru (*select*)

```
<select size="liczba" multiple>  
  <option value="tekst" />tekst  
  ...  
</select>
```

Wybór plików z dysku lokalnego (*file*)

```
<input type="file" size="liczba" />
```

Pola ukryte (*hidden*)

```
<input type="hidden" value="abc"/>
```

Przyciski polecenia (*submit* i *reset*)

```
<input type="submit" value="abc"/>  
<input type="reset" value="abc"/>
```

Łączenie tekstu z kontrolką (*label*)

```
<input type="text" id="x">  
<label for="x">Tekst</label>
```

II. Kaskadowe arkusze stylów (CSS)

Definicja elementów formatujących (CSS) może zostać dołączona do tworzonej strony na dwa sposoby:

- może być umieszczona wewnątrz znacznika `<style>`, zarówno w sekcji zawierającej treść, jak i w części nagłówkowej:

```
<head>
  <style>
    definicja stylów...
  </style>
</head>
```

- może zostać zapisana w osobnym pliku i dołączona do strony w sekcji `<head>` przy pomocy znacznika `<link>`:

```
<link rel="stylesheet" type="text/css" href="katalog/plik.css"/>
```

1. Przykładowe właściwości i wartości selektorów CSS:

Nazwa elementu	Właściwość	Wartość (w nawiasie znaczenie wartości)
Jednostki miary	Em ex Npx Ncm Nmm Nin Npt Nem	rozmiar równy wysokości czcionki rozmiar równy wysokości litery x N pikseli (N – liczba całkowita) N centymetrów N milimetrów N cali N punktów (1pt=0,375 mm=1/72 cala) szerokość N liter m
Formatowanie tekstu	color background-color	kolor tekstu kolor tła
Styl tekstu	font-style font-weight text-decoration	italic (czcionka pochylona) normal (czcionka zwykła) bold (pogrubiony) lighter (cienki) underline (podkreślony) overline (linia nad tekstem) line-through (przekreślony) blink (migający)
Rodzaj (nazwa) czcionki	font-family	np.: "Times New Roman" Times Serif
Wielkość czcionki	font-size	jako właściwość należy wykorzystać jednostki miary
Odstęp między znakami	letter-spacing	należy wykorzystać jednostki miary;
Odstęp między wierszami	line-height	należy wykorzystać jednostki miary;
Odstęp między wyrazami	word-spacing	np. word-spacing: 1em; ustali odstęp między wyrazami na szerokość litery m

Rodzaj numerowania list numerowanych 	list-style-type	decimal (liczby arabskie) lower-roman (małe rzymskie) upper-roman (duże rzymskie) lower-alpha (małe litery) upper-alpha (duże litery)
Znak wyróżniający listy wypunktowane 	list-style-type	disc (pełne kółko) circle (okrąg) square (pełny kwadrat)

Nazwa elementu	Właściwość	Wartość (w nawiasie znaczenie wartości)
Formatowanie tabel <table>		
wyrównanie tekstu w tabeli w pionie	vertical-align	top; middle; bottom;
szerokość tabeli	width	może być podawana w jednostkach bezwzględnych (np. 150px;) lub względnych (np. 60)
odległość między komórkami	cellspacing	należy wykorzystać jednostki miary
atrybuty krawędzi wokół tabeli	border (wszystkie) border-left border-right border-top border-bottom	podawane są kolejno: grubość (np. thin, thick, 2px) styl (np. solid, dashed, dotted) kolor (np. black, yellow, red)
wyrównanie tabeli w poziomie	margin-left margin-right	należy ustawić odpowiednio: obydwa równe auto; - tabela na środku jeden auto; a drugi 0 - tabela z lewej lub z prawej

2. Wybrane pseudoklasy

first-letter
first-line

Pseudoklasy związane ze znacznikiem <a>

active
link
visited
hover

III. JavaScript

1. Dołączanie plików zewnętrznych JavaScript do HTML

Choć nie jest to wymóg formalny, dla zachowania porządku na stronie internetowej zaleca się, aby wszystkie funkcje w JavaScript były zdefiniowane w części nagłówkowej `<head>`, zawsze jednak muszą być umieszczone wewnątrz znacznika `<script>`:

```
<script type="text/javascript"> lub <script language="javascript">  
    definicje funkcji (skrypty)  
</script>
```

Funkcje w JavaScript mogą być także umieszczone w pliku zewnętrznym. Aby go dołączyć do dokumentu HTML należy, w sekcji `<head>`, wpisać wiersz:

```
<script language="javascript" src="katalog/nazwa_pliku.js"/>
```

2. Operatory

Nazwa	Składnia
Operatory relacyjne	<code>==</code> równy
	<code>!=</code> różny
	<code>></code> <code>>=</code> większy, większy lub równy
	<code><</code> <code><=</code> mniejszy, mniejszy lub równy
Operatory logiczne	<code>&&</code> iloczyn logiczny (and)
	<code> </code> suma logiczna (or)
	<code>!</code> negacja (not)
Operator przypisania	<code>=</code>

3. Instrukcja warunkowa i pętle

Nazwa	Składnia
Składnia instrukcji warunkowej	<pre>if (warunek) { instrukcje } [else { instrukcje }]</pre>
Pętla WHILE	<pre>while (warunek) { instrukcje }</pre>
Pętla FOR	<pre>for (wartość początkowa; warunek zakończenia; inkrementacja licznika) { instrukcje }</pre>
Pętla FOR IN	<pre>for (zmienna in obiekt) { instrukcje }</pre>

4. Zmienne tablicowe

Tablice są obiektami, deklaruje się je w następujący sposób:

```
t=new Array();
```

Można także, choć nie jest to konieczne, sprecyzować liczbę elementów:

```
t=new Array(n);
```

w miejsce n wpisując liczbę naturalną oznaczającą, ile elementów ma mieć tablica.

Istnieje również możliwość definicji tablicy z wartościami początkowymi:

```
var t = new Array(x1,x2,...,xn);
```

gdzie:

$x1..xn$ to wartości przypisywane kolejnym indeksom tablicy.

Aby otrzymać element tablicy t o indeksie n , należy użyć konstrukcji:

```
t[n]
```

5. Obiekty i metody do wykonywania operacji na datach

Do wykonywania operacji na datach wykorzystywany jest obiekt

```
Date(data)
```

gdzie $data$ jest równa dacie w postaci łańcucha znakowego albo liczb oddzielonych przecinkami (kolejno: rok, miesiąc, dzień), albo w ogóle nie jest podawana.

Do wykonywania operacji na datach można wykorzystywać metody:

<code>getDate()</code>	numer dnia w miesiącu
<code>getMonth()</code>	numer miesiąca (miesiące są numerowane od 0–11)
<code>getFullYear()</code>	rok czterocyfrowy
<code>getDay()</code>	numer dnia w tygodniu (0-niedziela, 1-poniedziałek itd.)
<code>getHours()</code>	godzina
<code>getMinutes()</code>	minuta
<code>getSeconds()</code>	sekunda

6. Wyrażenia regularne

Wybrane znaki wykorzystywane do tworzenia wyrażeń regularnych

Znaki	Opis
.	(kropka) oznacza dowolny znak
\ (backslash)	poprzedza znaki specjalne (m.in. ^\$\/?*.+[]{}())
[] (nawiasy kwadratowe)	wewnątrz nawiasów kwadratowych umieszcza się znaki dopuszczalne
^ (daszek)	oznacza „za wyjątkiem”, musi być umieszczony jako pierwszy znak w nawiasie kwadratowym
sekwencje ze znakiem \ :	
\d	dowolna cyfra (to samo co [0-9])
\D	dowolny znak oprócz cyfry: [^0-9]
\w	cyfra, litera lub podkreślenie [0-9a-zA-Z_]
\W	dowolny znak, byle nie cyfra, litera lub znak podkreślenia [^0-9a-zA-Z_]
Krotności występowania znaków:	
?	brak lub pojedyncze wystąpienie
*	dowolna liczba znaków, także zero
+	dowolna liczba znaków, co najmniej jeden
{i,j}	minimalna i oraz maksymalna j liczba znaków
{i,}	minimalna liczba i, maksymalna dowolna
{i}	dokładnie i znaków
(pipeline)	alternatywa
() nawiasy okrągłe	grupowanie znaków

Aby sprawdzić, czy wartość wpisana w formularzu pasuje do zdefiniowanego wzorca należy użyć instrukcji:

```
wynik = document.nazwa_formularza.nazwa_pola.value.match(wzor);
```

gdzie:

wzor	zmienna zawierająca zdefiniowany
nazwa_formularza	wartość parametru name formularza
value	wartość pola
match	właściwość sprawdzająca dopasowanie do wzorca
wynik	zmienna, której przypisywany jest wynik wyrażenia

Wersja uproszczona:

```
nazwa_pola.value
```

7. Okna dialogowe

- okno informacyjne:

```
alert("łańcuch znakowy")
```

- okno dialogowe:

```
confirm("łańcuch znakowy").
```


8. Wybrane metody sprawdzania poprawności danych

- sprawdzenie, czy wyrażenie (zwykle tekst) jest liczbą

`isNaN(wyrażenie)`

- zamiana wyrażenia na liczbę całkowitą

`parseInt(wyrażenie)`

- wybieranie fragmentu łańcucha znakowego

`s.substring(m, n)`

- zmiana wielkości liter

`toUpperCase('łańcuch_znakowy')`

`toLowerCase('łańcuch_znakowy')`

- wyszukiwanie podciągu w łańcuchu znakowym

`indexOf`

9. Zdarzenia

- `onFocus`

uaktywnienie obiektu

- `onBlur`

dezaktywacja obiektu

- `onChange`

dokonanie zmiany w obiekcie lub dezaktywacja obiektu

- `onClick (onDbClick)`

kliknięcie (podwójne kliknięcie) elementu

- `onSubmit`

zdarzenie oznaczające próbę wysłania formularza (kliknięcie przycisku `Submit`)

- `onMouseOver`

umieszczenie kursora myszy nad obiektem

- `onMouseOut`

przesunięcie kursora myszy poza wskazywany obiekt

10. Właściwości zmiennych i pól formularza

- pobranie wartości obiektu (`value`)

`document.nazwaFormularza.nazwaObiektu.value`

- odczytanie długości wartości obiektu (`length`)

`document.nazwaFormularza.nazwaObiektu.value.length`

- sprawdzenie, czy pole wyboru zostało zaznaczone (dla pól wyboru)

`document.nazwaFormularza.nazwaPolaWyboru.checked`

IV. PHP

1. Operatory

Nazwa zasady	Składnia / przykład
Operatory relacyjne	<code>==</code> równy <code>!=</code> różny <code>></code> <code>>=</code> większy, większy lub równy <code><</code> <code><=</code> mniejszy, mniejszy lub równy
Operatory logiczne	<code>and</code> iloczyn logiczny <code>or</code> suma logiczna <code>!</code> zaprzeczenie (not)
Operator przypisania	<code>=</code>

2. Instrukcja warunkowa i wybrane pętle

Nazwa instrukcji	Składnia / przykład
Składnia instrukcji warunkowej <code>if</code>	<pre>if (warunek) { ciąg instrukcji }[else { ciąg instrukcji }];</pre>
Składnia pętli <code>for</code>	<pre>for (zmiennaSterująca=wartośćPoczątkowa; warunekZakończenia; zmianaLicznika) { ciąg instrukcji; };</pre>
Składnia pętli <code>while</code>	<pre>while (warunek) { ... };</pre>

3. Składnia funkcji w PHP

```
function nazwa (lista parametrów formalnych)  
{  
    instrukcje wchodzące w skład funkcji;  
    return wyrażenie;  
}
```

4. Dołączanie kodu z pliku zewnętrznego

```
include "plik.php";
```

5. Operacje na plikach dyskowych

- otwarcie pliku:

```
$f = fopen("katalog/plik", "parametr");
```

jako parametr należy wskazać sposób otwarcia pliku, np.:

- "a" plik jest otwierany do zapisu, wskaźnik plikowy jest ustawiany na końcu pliku, jeśli plik nie istnieje, jest tworzony
- "a+" plik jest otwierany do zapisu i odczytu, wskaźnik plikowy jest ustawiany na końcu pliku, jeśli plik nie istnieje, jest tworzony
- "r" plik jest otwierany do odczytu, wskaźnik plikowy jest ustawiany na początku pliku
- "r+" plik jest otwierany do odczytu i zapisu, wskaźnik plikowy jest ustawiany na początku pliku

- "w" plik jest otwierany do zapisu, jeśli nie istnieje, jest tworzony, jeśli istnieje, jego zawartość jest usuwana
- "w+" plik jest otwierany do zapisu i odczytu, jeśli nie istnieje, jest tworzony, jeśli istnieje, jego zawartość jest usuwana
- zamknięcie pliku:
`fclose($f);`
- zapis danych do pliku:
`fwrite($f, "tekst");`
- przykładowe znaczniki:
 - `\n` koniec wiersza
 - `\t` tabulator

6. Przesyłanie plików z dysku lokalnego na serwer

W nagłówku formularza musi być parametr `enctype="multipart/form-data"`.

- zawartość tablicy asocjacyjnej `$_FILES`:

<code>\$_FILES[f_plik][tmp_name]</code>	tzw. tymczasowa nazwa pliku (nadawana przez system) przesłanego za pośrednictwem formularza,
<code>\$_FILES[f_plik][name]</code>	właściwa nazwa pliku wybranego w formularzu;
<code>\$_FILES[f_plik][size]</code>	rozmiar przesłanego pliku (w bajtach),
<code>\$_FILES[f_plik][type]</code>	typ przesłanego pliku,
<code>\$_FILES[f_plik][error]</code>	numer błędu, przykładowo: 0 – operacja przebiegła bezbłędnie, 3 – plik nie został przesłany poprawnie, 4 – nie wybrano pliku w formularzu,
- funkcje:
 funkcja logiczna sprawdzająca, czy plik został przesłany na serwer:
`is_uploaded_file($_FILES[f_plik][tmp_name])`
 funkcja zapisująca plik na serwerze we wskazanej lokalizacji:
`move_uploaded_file (tymcz. nazwa pliku, katalog i nazwa pliku)`

7. Komunikacja z bazą danych i wykonywanie operacji na danych

```
pg_connect("host=serwer dbname=baza_danych
           user=uzytkownik password=haslo");
pg_exec($con, $query);
pg_numrows($result);
pg_numfields($result);
pg_result($result, nr_wiersza, nr_kolumny)
pg_field_name($result, nr_kolumny)
pg_affected_rows($result);
```

```
$row=pg_fetch_array($result, numerWiersza);
```

Aby w tym przypadku uzyskać nazwę indeksu (czyli kolumny tabeli zwróconej przez zapytanie), należy użyć funkcji *key*:

```
key($row);
```

Tablica będąca wynikiem działania funkcji `pg_fetch_array` składa się z jednego wiersza i tylu kolumn, ile pól zostało zwróconych przez zapytanie. Jest to więc zawsze tablica jednowymiarowa. Dostęp do poszczególnych jej elementów następuje więc po podaniu jej nazwy i jednego indeksu umieszczonego w nawiasach kwadratowych:

```
$row[nr_kolumny]
```

lub, dzięki temu, że tablica jest asocjacyjna:

```
$row[nazwa_pola]
```

V. PostgreSQL

1. Nawiązanie połączenia z SZBD PostgreSQL:

```
psql -h sbazy
```

2. Wybrane polecenia systemowe SZBD PostgreSQL

Akcja	Polecenie
wyświetlenie listy wszystkich tabel w bazie danych	\dt
wyświetlenie struktury tabeli o podanej nazwie	\d <i>nazwa_tabeli</i>
zerwanie połączenia z PostgreSQL i powrót do SO Linux	\q
wyświetlenie listy poleceń PostgreSQL	\?
pomoc nt. wybranej instrukcji SQL	\h <i>nazwa_instrukcji</i>
wykonanie wszystkich instrukcji SQL zapisanych w pliku tekstowym	\i <i>nazwa_pliku</i>

3. Wybrane typy pól w PostgreSQL

Typ pola	Nazwa typu w PostgreSQL
autonumer	serial, bigserial
tekst	char(n), varchar(n), text
całkowity	int2 int4 int8
rzeczywisty	float4 float8
walutowy	money
data	date
godzina	time
data i godzina	dateTime, timeStamp
stałe daty i czasu	current_date current_time
logiczny	bool