

Instrukcja 1 – Wprowadzenie do języka HTML

Wszystkie serwisy internetowe, w tym aplikacje internetowe, niezależnie od obranej technologii wykonania mają jeden, wspólny współczynnik – język HTML (HyperText Markup Language), czyli hipertekstowy język znaczników.

Aktualnie w programowaniu HTML znajduje coraz szersze zastosowanie – prócz projektowania stron internetowych służy również do tworzenia dokumentów tekstowych, menu w urządzeniach elektronicznych (jak telewizory, projektory, odtwarzacze multimedialne), formatek (GUI) w programach komputerowych i telefonach komórkowych (aplikacje HTML5, tworzone zarówno w językach wyższego jak i niższego poziomu, w tym C/C++) czy wręcz wyglądu całego systemu operacyjnego (przykładem może być FirefoxOS, ChromiumOS/ChromeOS czy Tizen). Istnieje także kilka gier komputerowych napisanych tylko i wyłącznie w języku HTML 5.

Co sprawia, że to właśnie ten język, a nie inny wybierany jest coraz częściej przez programistów/projektantów? Przede wszystkim prostota – niemal każdy jest w stanie się go nauczyć w bardzo szybkim czasie. Pozwala to na przerzucenie projektowania wyglądu dowolnej aplikacji na specjalnie wydzielonych projektantów wyglądu/grafików (bądź zapewnia lepszą kooperację pomiędzy programistami a grafikami). Innymi profitami są:

- prostota implementacji – bardzo łatwo podłączyć go do dowolnego kodu napisanego w niemal dowolnym języku
- przejrzystość kodu – ponieważ polecenia (znaczniki) reprezentują pełne słowa/skróty słów języka ludzkiego (angielskiego) kod w nich napisany jest klarowny i przejrzysty; dodatkowy atut to budowa plików HTML – kod układany jest hierarchicznie, a większość znaczników jest otwierana i zamykana przez co widać ich „zasięg”
- przenośność – programy z interfejsem HTML są wysoce przenośne pomiędzy różnymi systemami operacyjnymi i urządzeniami (tzw. międzyplatformowość). Raz zaprojektowany układ wyglądu zawsze tak samo na różnych systemach/urządzeniach.
- rozwijalność – aplikacje wykorzystujące znaczniki dają się w bardzo prosty „rozwijać”. Oznacza to, że łatwo jest dodać nowe okno, przycisk czy inny element graficzny (bądź usunąć/zmodyfikować);

Oczywiście można by dalej wymieniać zalety stosowania znaczników hipertekstowych. Jednak wykorzystywanie ich ma też pewne wady:

- powolność – aplikacje wykorzystujące interfejsy użytkownika pisane przy wykorzystaniu HTML są znacznie wolniejsze od „natywnych” odpowiedników; dzieje się tak przez nieefektywność interpreterów tego języka
- niestandardyzowanie – pomimo tzw. ujednolicenia standardu poprzez konsorcjum W3C niektóre z większych firm (m in. Microsoft czy Apple) wprowadziły własne znaczniki, które przeważnie nigdy nie weszły do ogólnego standardu. Powoduje to pewien chaos zwłaszcza, że niektóre z tzw. silników HTML nadal interpretują te znaczniki.
- niestandardyzowanie interpreterów – obecnie na rynku jest wiele interpreterów (inaczej mówiąc parserów lub silników) HTML. Niestety każdy z nich inaczej radzi sobie z interpretacją niektórych znaczników i/lub ich parametrów, a przez to ostatecznie utworzona strona/aplikacja może nie zawsze wyglądać identycznie we wszystkich przeglądarkach dostępnych na rynku (wtedy trzeba ją optymalizować pod konkretne wersje)
- najnowsza wersja nadal w wersji draft (szkic) – najnowsza wersja języka (5 odsłona) wprowadza szereg udoskonaleń (bezpośrednia obsługa multimediiów, grafiki oraz tzw. dynamizmu języka) oraz upragnione ujednolicenie składni (wyrzucenie nieużywanych/niestandardowych znaczników oraz dodanie nowych na potrzebę obsługi nowej funkcjonalności). Standard nadal jest jednak w fazie rozwoju, a przewidywane zakończenie prac nad nim zaplanowano dopiero na rok 2022; do tego czasu pisanie w nim czegokolwiek jest możliwe, jednak może zdarzyć się przypadek iż pewne używane aktualnie znaczniki mogą nie zostać włączone do ostatecznego standardu.

Na zajęciach zapoznamy się z najnowszą, 5 wersją HTML (draft).

1. Podstawy HTML.

Jak wygląda najprostsza strona napisana w HTML od strony źródła (kodu)? Przykład poniżej:

```
<!DOCTYPE html>                                <!--1-->
<html lang="pl">                                <!--2-->
<head>                                          <!--3-->
    <meta charset="UTF-8"/>                    <!--4-->
    <title>Pierwsza strona w HTML5</title>      <!--5-->
</head>                                        <!--6-->
<body>                                         <!--7-->
    <p>Pierwszy paragraf pierwszej strony</p>  <!--8-->
</body>                                        <!--9-->
</html>                                       <!--10-->
```

Omówmy teraz szczegółowo kolejne linie.

Pierwsza określa nam z jakim dokumentem mamy do czynienia. Od wersji HTML5, aby powiadomić parsery/aplikacje o typie html wystarczy przedstawiona powyżej deklaracja; we wcześniejszych wersjach była ona nieco dłuższa i trudniejsza do zapamiętania.

Druga linia zawiera właściwy znacznik mówiący parserowi, że od tego momentu zaczyna się treść całego dokumentu HTML. Przy okazji mamy tu do czynienia z naszym pierwszym parametrem znacznika; parametry mogą zawierać dodatkowe informacje dla danego znacznika określając jego wygląd, ułożenie, przechowywaną wartość czy zachowanie w określonych warunkach. W tym konkretnym przypadku określony został język zastosowany do przechowywania treści dokumentu. Informacja ta wykorzystywana jest przez parsery i/lub roboty wyszukiwarek WWW.

Linia numer trzy otwiera nagłówek strony (head). W nagłówku strony umieszczane są przeważnie meta znaczniki (czyli znaczniki kontekstowe z dodatkowymi informacjami o stronie). W powyższym przykładzie nagłówek zawiera dwa tagi – meta z parametrem charset (określa kodowanie znaków na stronie; informacja ważna dla przeglądarki – jeżeli kodowanie nie zostanie określone użyte zostanie domyślne, niekoniecznie poprawne) oraz title – tytuł strony (wyświetlany np. na zakładce strony w przeglądarce bądź belce tytułowej okna). Linia nr 6 zamyka nagłówek. Od linii numer 7 zaczyna się właściwa treść dokumentu HTML. To właśnie w znaczniku body (ciało) umieszcza się teksty, obrazy, formularze i inne elementy strony, które wyświetlają się docelowemu użytkownikowi serwisu. W zaprezentowanym przykładzie w ciele znajduje się tylko jedna linia tekstu – umieszczona w znaczniku p (paragraf). To właśnie ten tekst wyświetli się na naszej przykładowej stronie.

Ostatnie dwie linie to zamknięcie wcześniej otwartych znaczników – body oraz html.

Oznaczenia „<!--” oraz „-->” stanowią tag komentarza – wszystko co w nim napiszemy NIE ZOSTANIE zinterpretowane przez parser HTML. Pomiedzy wspomnianą sekwencję znaków można wpisać dowolne frazy, opisy itp.. Ponadto nie muszą mieścić się w jednej linii – komentarz może wyglądać np. tak:

```
<!--Tutaj jest komentarz
wieloliniowy; w nim możemy wpisać sobie np. opis dlaczego
w danej chwili tak a nie inaczej zaprojektowaliśmy fragment strony WWW.
Później może to stanowić przyczółek do dalszych prac nad rozwojem projektu -->
```

Aby wypróbować powyższy przykład (stworzyć stronę na jego podstawie) nie potrzeba żadnych specjalnych narzędzi – wystarczy systemowy notatnik. Oczywiście nie jest on zalecany (chyba, że pracujemy w systemie Unix/Linux) ze względu na nie ergonomiczność – nie posiada wielu

niezwykle przydatnych funkcji.

O wiele lepiej jest użyć darmowego i bardzo wygodnego programu notepad++ (<http://notepad-plus-plus.org/>). Jego instalacja przebiega standardowo i zajmuje niewiele miejsca na dysku twardym. Poza tym posiada przejrzysty i jasny interfejs użytkownika (dostępny także po polsku), a wszelkie pliki tekstowe domyślnie zapisuje w kodowaniu UTF-8.

Jeżeli mamy już wpisany kod naszej strony do edytora plików tekstowych to należy go zapisać pod dowolną nazwą z rozszerzeniem .htm lub .html (oba są poprawnymi rozszerzeniami dokumentów HTML). Po zapisaniu pliku wystarczy otworzyć go w przeglądarce WWW – Firefox, Chrome, Chromium, Internet Explorer itp...

2. Parametry znaczników HTML

Każdy ze znaczników HTML może posiadać pewne dodatkowe, ustawiane przez projektanta opcje. W języku HTML nazywa się je atrybutami bądź parametrami. Większość atrybutów jest wspólna dla wszystkich znaczników (choć niewielka część będzie działać poprawnie tylko z niektórymi tagami). Poniżej przedstawione atrybuty są wspólne dla wszystkich znaczników.

- id – atrybut określa JEDNOZNACZNĄ w całym dokumencie nazwę, pod którą dany element występuje. Identyfikator niezbędny jest do tworzenia odpowiedniego stylu CSS (następne zajęcia) lub dla skryptów JavaScript.

```
<p id="Paragraf1">To jest paragraf pierwszy</p>
<p id="paragraf2">W tym paragrafie będzie także element <span id="spanParagraf1">typu span</span> także posiadający swoje id</p>
```

- hidden – parametr określa, że dany znacznik (wraz z zawartością) będzie niewidoczny na stronie

```
<p>Ten paragraf jest widoczny</p>
<p hidden="">Ten już nie powinien; chyba że używasz Internet Explorer</p>
```

- dir – parametr działa z większością tagów tekstowych; dzięki niemu można określić kierunek wyświetlanego tekstu; możliwe wartości to „ltr” (lewo->pravo), „rtl” (pravo->lewo) lub „auto” (przeglądarka sama ma określić kierunek tekstu na podstawie zawartości elementu)

```
<p dir="rtl">Tekst pisany od prawej</p>
<p> Część tekstu pisana normalnie, a część <span dir="rtl">odwrócona (nawiasy zostaną przemieszczone)</span></p>
```

- title – atrybut pozwala na nadanie dodatkowego tytułu wybranemu fragmentowi tekstu bądź np. całemu paragrafowi, artykułowi itd.

```
<p title="Tytuł: Pierwszy paragraf">Tutaj umieszczamy tekst pierwszego paragrafu na stronie</p>
<p>Paragraf bez tytułu; ale zawsze można dodać <span title="Tytuł fragmentu"> pole span z osobnym tytułem</span></p>
```

- contenteditable – określa, że dany element strony umożliwia użytkownikowi zmianę swojej zawartości:

```
<p contenteditable="true">To jest edytowalny paragraf</p>
<p>Ten paragraf nie, ale już <span contenteditable="true">ten jego fragment jak najbardziej</span></p>
```

- style – pod tym parametrem można ustawić wartość stylu dla wybranego elementu; ustawienia spod tego parametru NADPISZĄ wszelkie ustawienia globalne dostępnych stylów.

```
<p style="color:red;text-decoration:underline;">Tekst z ustawionym stylem</p>
```

(szczegóły stylów zostaną omówione wraz z arkuszami stylów CSS).

- class – parametr pozwala przypisać do elementu pożądane ustawienie stylu CSS wcześniej zdefiniowanego w znaczniku style lub osobnym arkuszu stylów CSS. Przykład działania tego atrybutu zostanie zaprezentowany wraz z omówieniem CSS.

Więcej atrybutów (wraz z ich opisem) można znaleźć pod adresem:

http://www.w3schools.com/tags/ref_standardattributes.asp

3. Więcej tekstu na stronie WWW.

Przeważająca ilość stron, poza obrazami, składa się z tekstu. W punkcie pierwszym została zaprezentowana jedna z metod wypisania tekstu na stronie WWW (poprzez znacznik <p></p>). Oczywiście można było wypisać tekst bez jakichkolwiek znaczników lecz takie rozwiązanie nie jest preferowane w 5 odsłonie HTML (dozwolone we wcześniejszych wersjach). Aby zobaczyć rzeczywiste działanie znacznika paragrafu należy dodać większą ilość tekstu:

```
<p>Pierwszy paragraf pierwszej strony ewfree stb srtxhb xbsezhb rhes xbzr hsrxtdraz bth
sdxttdhsxtddhdbxtn dnrntxnrd ftnrdtndrntdntyn dfnrdndtyfntyd ntydn dtynr nrdn rdn drd n trr mjrdj fd
dtn fnrd jsxths ntdfxhjtrdb rtsrtdns tnrd tndn trd jndfgndfhrtrt rth rdth rth tr hdr ntrdxjarqawq q
gqh strd ntrdn n</p>
```

```
<p>Pierwszy paragraf pierwszej strony</p>
```

```
<p>Pierwszy paragraf pierwszej strony ewfree stb srtxhb xbsezhb rhes xbzr hsrxtdraz bth
sdxttdhsxtddhdbxtn dnrntxnrd ftnrdtndrntdntyn dfnrdndtyfntyd ntydn dtynr nrdn rdn drd n Pierwszy
paragraf pierwszej strony rrea hnesxdj hnxd ntrdxrjnsxdd6j sdrd 6jndx gynt nsr ndy mnftcgmyndc
fmydte mrdtjxf dcftn d</p>
```

```
<p>Pierwszy paragraf pierwszej strony</p>
```

Jak widać kolejne bloki oddzielone są od siebie przez pusta linię wysokości jednego wiersza.

Trzeba jednak pamiętać, że tekst wypisywany w znaczniku p, jak i w samym HTML, podlega

pewnym ograniczeniom; wiele pustych (białych) spacji (tworzonych klawiszem SPACJA) jest

zamieniane na jedną. Złamanie linii (za pomocą klawisza ENTER/RETURN) także nie jest

rozpoznawalne w HTML (tekst będzie wypisywany, jeżeli wielkość strony nie została ograniczona,

aż do jej prawego końca po czym parser sam przeniesie dalszy tekst do nowej linii).

Jeżeli chcemy przełamać tekst do nowej linii należy zastosować samo zamykający się tag br (break

line – przełamanie linii):

```
<br/>
```

Przykład:

```
<p>Pierwszy paragraf pierwszej strony ewfree stb <br/>srtxhb xbsezhb rhes xbzr hsrxtdraz bth
sdxttdhsxtddhdbxtn dnrntxnrd ftnrdtndrntdntyn dfnrdndtyfntyd <br/>ntydn dtynr nrdn rdn drd n trr
mjrdj fd dtn fnrd jsxths ntdfxhjtrdb rtsrtdns tnrd tndn <br/>trd jndfgndfhrtrt rth rdth rth tr hdr
ntrdxjarqawq q gqh strd ntrdn n</p>
```

Podczas tworzenia tekstowego nagłówka (nazwa strony, nazwa sekcji na stronie itp.) możemy

zastosować jeden z sześciu predefiniowanych tagów: <h1></h1>, <h2></h2>, <h3></h3>,

<h4></h4>, <h5></h5>, <h6></h6>

Przykład:

```
<h1>Nagłówek H1</h1>
<h2>Nagłówek H2</h2>
<h3>Nagłówek H3</h3>
<h4>Nagłówek H4</h4>
<h5>Nagłówek H5</h5>
<h6>Nagłówek H6</h6>
```

Jeżeli chcemy natomiast oddzielić jedną część tekstu od drugiej (np. dwa paragrafy bądź dwie linie w bieżącym paragrafie) wystarczy użyć znacznika `<hr/>`.

Przykład:

```
<p>paragraf pierwszy</p>
<hr/>
<p>paragraf drugi oddzielony linią; <hr/> tekst po linii znajdującej się w bieżącym paragrafie</p>
```

Aby napisać tekst indeksu dolnego (przykład) lub górnego (przykład) używamy dwóch specjalnych znaczników: `<sup>` dla indeksu górnego oraz `<sub>` dla dolnego.

Przykład:

```
<p>W tym akapicie mamy przykład tekstu <sub>dolnego</sub> oraz górnego jako np.
przypis<sup>[1]</sup>. Dalsza część tekstu jest standardowa</p>
```

Oto lista wybranych tagów operujących na tekście:

- `<wbr>` - wskazuje miejsce możliwego przełamania długiego tekstu (jeżeli będzie ono wymagane):

```
<p>Jeżeli treść tego akapitu będzie musiała być przeniesiona do nowej linii, a przeniesienie będzie
miało nastąpić na słowie XML<wbr>Http<wbr>Request to będzie ono mogło być „złamane” po
części XML, Http lub Request (najlepiej wypróbować samemu!)</p>
```

- `<abbr>` - używany przeważnie dla skrótów; znacznik powinien wystąpić wraz z parametrem `title`, w którym podajemy rozwinięcie/wytłumaczenie skrótu. Bez parametru znacznik nie spełni swojej roli.

```
<p>Strony internetowe są dostępne dla użytkowników dzięki usłudze <abbr title=„World Wide
Web”>WWW</abbr>.</p>
```

- `<mark>` - pozwala na zaznaczenie części tekstu (tła, na którym jest pisany) w taki sposób, jakby został zakreślony mazakiem

```
<p>W tym paragrafie pewien fragment tekstu będzie <mark>zakreślony</mark>.</p>
```

- `<bdo>` - umożliwia zmianę wypisywania tekstu „z lewej do prawej” na „z prawej do lewej” (wraz z lustrzanym odbiciem); aby znacznik zadziałał poprawnie trzeba zaopatrzyć go w parametr `dir`, mogący przyjąć tylko dwie wartości: „ltr” lub „rtl”

```
<p><bdo dir=„ltr”>Tekst napisany w lewo</bdo><br/>
<bdo dir=„rtl”>Tekst obrócony w prawo</bdo></p>
```

- `<blockquote>` - używa się do tekstów cytowanych z innych źródeł. Tekst jest lekko wcięty względem pozostałych elementów (np. `<p>`), a w jego parametrze `cite` ustawia się pochodzenie cytowanej wypowiedzi/fragmentu.

```
<p>Tutaj jakiś przykładowy tekst.</p>
```

```
<p>Kolejny akapit</p>
<blockquote cite="http://www.example.com">Tutaj z kolei mamy do czynienia z cytowaniem
tekstu z zewnętrznego źródła. Widoczne jest wcięcie po lewej stronie, a w kodzie źródłowym widać
jest adres cytowania. Szczególnie ważne jeżeli cytujemy tekst chroniony prawami autorskimi!
</blockquote>
```

Oczywiście nie są to wszystkie znaczniki HTML5 dotyczące tekstu. Spis wszystkich znaczników można znaleźć pod adresem <http://dev.w3.org/html5/html-author/#the-elements> lub <http://www.w3schools.com/tags/default.asp>.

4. Używanie wielu znaczników stosowanych w obrębie jednej zawartości.

HTML umożliwia obudowanie np. fragmentu tekstu większą ilością znaczników:

```
<p>Paragraf <span>posiadający <abbr title="Trzecie obramowanie">tekst
obudowany<mark>wieloma znacznikami HTML</mark></abbr></span>; koniec paragrafu</p>
```

Powyższy przykład zbudowany został poprawnie – wszystkie znaczniki zostały zamknięte poprawnie (w kolejności otwierania). Niepoprawne byłoby zamknięcie w ten sposób:

```
<p>Paragraf <span>posiadający <abbr title="Trzecie obramowanie">tekst
obudowany<mark>wieloma znacznikami HTML</abbr></mark>; koniec paragrafu</p></span>
```

Co prawda interpreter przeglądarki powinien sobie poradzić z przedstawionymi znacznikami jednak taka niefrasobliwość może spowodować błędne przystosowanie pożądanego fragmentu strony (np. złe pokolorowanie tekstu czy też zbyt długi adres odnośnika).

Zadania do wykonania:

1. Zainstalować program notepad++ (jeżeli nie ma zainstalowanego w systemie)
2. Utworzyć plik strona.htm. Strona ma zostać zbudowana wedle standardu (ma posiadać deklarację typu dokumentu, nagłówek, ciało oraz stopkę). Tytuł strony ma posiadać wartość „Moja pierwsza strona HTML”.
3. Znaleźć w sieci dostępne sposoby/generatory losowej treści na stronach WWW (przydatne podczas projektowania stron WWW).
3. Wypróbować wszystkie opisane w instrukcji znaczniki tekstowe. Należy pamiętać by wpisany tekst był odpowiednio długi/by na stronie znalazła się jego odpowiednia ilość.
4. Sprawdzić jak w ten sposób stworzona strona zachowa się przy zamianie deklaracji typu dokumentu na XHTML; można także sprawdzić pozostałe deklaracje (dla starszych wersji).
5. Odnaleźć niekompatybilne (usunięte) znaczniki HTML z wersji 5. Wypróbować ich działanie.