



Szkielet strony internetowej

- Wprowadzenie
- Przeczytaj
- Mapa myśli
- Sprawdź się
- Dla nauczyciela



Szkielet strony internetowej

Źródło: domena publiczna.

Podczas nauki programowania w języku HTML kluczowe jest zrozumienie przeznaczenia każdej sekcji kodu źródłowego, która znajduje się w podstawowym szkielecie tworzącym współczesną witrynę WWW. Po przyswojeniu tej wiedzy, można rozpocząć projektowanie własnych stron internetowych.

Poznanie anatomii podstawowego pliku źródłowego HTML, pozwoli zdobyć praktyczną wiedzę na temat działania witryn.

Kod źródłowy strony internetowej stanowi zbiór przemyślanych, logicznych zabiegów, dokonanych przez programistę. Celem tych zabiegów jest poprawa działania witryny w przeglądarce.

Twoje cele

- Przeanalizujesz podstawowe znaczniki HTML, które występują w szablonach współczesnych witryn.
- Stwierdzisz, jak wiele różnorodnych aspektów technicznych przekłada się na poprawność działania strony internetowej w przeglądarce.
- Prześledzisz budowę oraz przeznaczenie podstawowych znaczników HTML wykorzystywanych we współczesnych kodach źródłowych.

- Ocenisz skuteczność zabiegów technicznych, które wykorzystują programiści webowi podczas pracy nad szablonem strony.

Przeczytaj

Szkielet współczesnej strony internetowej to zestaw najbardziej elementarnych znaczników, których obecność w kodzie zagwarantuje poprawne działanie witryny w przeglądarce oraz właściwe jej zaindeksowanie przez wyszukiwarki.

Na początku tworzenia witryny warto uwzględnić cały ogół czynności mających zapewnić jak najwyższą pozycję strony w wynikach wyszukiwania w Google – jest to tak zwany kontekst [SEO](#) programowania webowego.

Analizę kodu źródłowego rozpoczniemy od wstępnego przyjrzenia się całemu szkieletowi dokumentu zapisanemu w standardzie HTML5:

```
1 <!DOCTYPE html>
2 <html lang="pl">
3   <head>
4
5     <meta charset="utf-8">
6     <title>Tytuł strony</title>
7
8     <meta name="description" content="Opis zawartości strony">
9     <meta name="keywords" content="słowa, kluczowe, opis, zawartość">
10    <meta name="author" content="Jan Programista">
11
12    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
13    <meta name="viewport" content="width=device-width, initial-scale=1">
14
15    <link rel="shortcut icon" type="image/x-icon" href="ikonka.ico">
16    <link rel="stylesheet" href="arkusz.css">
17    <script src="skrypt.js"></script>
18
19    <!--[if lt IE 9]>
20    <script src="html5shiv.min.js"></script>
21    <script src="respond.min.js"></script>
22    <![endif]-->
23
24  </head>
25  <body>
26
27    <!-- Zawartość podstrony -->
```

```
28
29 </body>
30 </html>
```

Zajmijmy się rozszyfrowaniem przeznaczenia każdej przedstawionej instrukcji.

Deklaracja typu dokumentu

Kod źródłowy witryny inicjujemy poinformowaniem przeglądarki, której wersji standardu języka użyjemy do stworzenia podstrony. To ważne, gdyż nawet starsze wersje języka (HTML 4.01, XHTML) są nadal obsługiwane w ramach zachowania w internecie zasady tzw. [kompatybilności wstecznej](#). Używana współcześnie deklaracja dla HTML5 prezentuje się następująco:

```
1 <!DOCTYPE html>
```

Taka deklaracja informuje przeglądarkę, iż dokument został zapisany w standardzie HTML5. Wcześniejsze deklaracje były dużo dłuższe w zapisie, gdyż występowały w trzech odmianach: `Strict`, `Transitional` oraz `Frameset`. Przez to zawierały dodatkowo wpis `DTD` (ang. *Document Type Definition*). Oto kilka przykładów starszych deklaracji:

- HTML 4.01 Strict:

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
2 "http://www.w3.org/TR/html4/strict.dtd">
```

- HTML 4.01 Frameset:

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
2 "http://www.w3.org/TR/html4/frameset.dtd">
```

- XHTML 1.1 DTD:

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
2 "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

Oczywiście wraz z upływem czasu, starsze (dużo bardziej rozbudowane) deklaracje zostały zastąpione prostym zapisem wprowadzonym od standardu HTML5.

Nagłówek i ciało witryny, polskie znaki

Już od lat 70-tych XX wieku, tradycją w programowaniu strukturalnym stało się konstruowanie własnych funkcji w taki sposób, aby zawierały dwie sekcje:

- **nagłówek**, który składa się z nazwy funkcji oraz z typów przetwarzanych parametrów i zwracanych przez nią wartości,
- **ciało**, czyli zamknięty (najczęściej w nawiasach klamrowych) blok instrukcji realizujących konkretne zadanie.

Odzwierciedleniem tej klasycznej metodologii tworzenia funkcji jest także anatomia elementarnej struktury strony internetowej:

- sekcja head to **część nagłówkowa**, która określa wiele parametrów technicznych witryny – np. tytuł, używany zestaw znaków, słowa kluczowe, opis, autor kodu, uwzględniane pliki składowe i tym podobne elementy,
- sekcja body to **ciało witryny**, czyli elementy składowe wyświetlające zawartość strony – znaczniki strukturalne, akapity tekstu, grafiki, listy, tabele, kontrolki formularzy i tak dalej.

Obie te klasyczne części składowe zamknięto dodatkowo w zbiorcze ramy podwójnego znacznika html:

```
1 <html lang="pl">
2   <head>
3     <-- Część nagłówkowa -->
4   </head>
5   <body>
6     <-- Ciało witryny -->
7   </body>
8 </html>
```

Ważnym atrybutem zapisu jest `lang="pl"`. Aby polskie znaki diakrytyczne na pewno wyświetlały się poprawnie, należy dopilnować obecności dwóch zapisów HTML, jak również poprawnie ustawić kodowanie samego pliku.

Warunki konieczne do spełnienia:

- język polski ustawiony zapisem `<html lang="pl">`,

- zastosowanie zestawu znaków utf-8; odpowiada za to znacznik `<meta charset="utf-8">`, który omówiono w sekcji „Mapa myśli”,
- ustawienie tego samego zestawu znaków dla pliku z rozszerzeniem `.htm` lub `.html`; rodzaj zastosowanego kodowania można zawsze zmienić w edytorze kodu źródłowego.

Znaczniki meta w sekcji nagłówkowej

Wskazanie używanego przez witrynę zestawu znaków powinno być pierwszym znacznikiem wstawionym do sekcji `head`. Kolejne znaczniki, np. tytuł, opis witryny, autor, mogą już zawierać znaki diakrytyczne:

```
1 <meta charset="utf-8">
```

Wybieramy współcześnie stosowany zestaw znaków utf-8, który zastąpił wcześniej używany dla polskich witryn standard iso-8859-2. Istnieje także dłuższa wersja tagu, którą można stosować zamiennie z wersją pierwotną:

```
1 <meta http-equiv="Content-Type" content="text/html; charset=utf-8
```

Ten wariant przypomina w zapisie analogiczny znacznik ze standardu HTML 4.01 i XHTML. Ponieważ oba zapisy są w HTML5 równoważne, to warto zdecydować się na prostą, krótszą wersję.

Kolejny ważny w kontekście [SEO](#) znacznik `meta` to opis strony w rezultatach wyszukiwania:

```
1 <meta name="description" content="Opis zawartości strony">
```

Do dyspozycji mamy około 155-160 znaków. Opis powinien składać się zarówno z kluczowych fraz, jak również bezpośrednio zwracać się do internauty, aby zainteresować go właśnie naszą witryną.

Wyszukiwarka Google, w miejscu opisu znalezionej witryny, często wstawia fragmenty stron, zawierających poszukiwane frazy. Internauta widzi więc kontekst użycia szukanych słów. To właśnie dlatego oryginalny opis z podanego znacznika nie zawsze będzie widoczny w Google.

Następny znacznik `meta` to kolekcja słów kluczowych:

```
1
```

```
1 <meta name="keywords" content="słowa, kluczowe, opis, zawartości"
```

Obecnie wyszukiwarka Google najczęściej ignoruje tę sekcję (z powodu nadużyć internautów w przeszłości), jednak wciąż umieszczamy ją w kodzie witryny, wpisując kilka najbardziej kluczowych, reprezentatywnych fraz. Znacznik ten nie przynosi więc korzyści podczas pozycjonowania danej strony WWW.

Możemy określić autora witryny:

```
1 <meta name="author" content="Jan Programista">
```

Jest to także znacznik opcjonalny, służący do podpisania pracy. W praktyce jednak dużo lepiej podpisać się na stronie linkiem do własnej witryny bądź portfolio (znacznik <a>). Zyskujemy wówczas kolejny link prowadzący do nas w rezultatach wyszukiwania Google – a to już wymierna korzyść zastosowania znacznika.

Warto jeszcze zatroszczyć się o poprawne wyświetlenie witryny w starszych wersjach przeglądarki Internet Explorer:

```
1 <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
```

Znacznik ustawia możliwie najnowszy sposób renderowania witryny w starszych przeglądarkach z grupy IE, wcześniejszych od wersji 11 – szczególnie w IE 8 i 9.

Ponadto, dodano opcjonalny zapis `chrome=1`, który poprawia wygląd witryny wyświetlonej z użyciem pluginu do IE, nazwanego Chrome Frame. Obecnie niewielu jest użytkowników IE + Chrome Frame, ale ponieważ sam zapis nie zajmuje wiele miejsca w kodzie, to wciąż spotkamy go w niektórych źródłach HTML5.

W przypadku tworzenia strony responsywnej, czyli dopasowującej sposób wyświetlania do ekranów urządzeń mobilnych, należy koniecznie dopisać znacznik:

```
1 <meta name="viewport" content="width=device-width, initial-scale="
```

Słowo `viewport` oznacza wirtualne okno, wewnątrz którego przeglądarka na urządzeniu mobilnym wyświetla responsywną witrynę. Ustawiamy szerokość `width` na wartość pobraną z właściwości ekranu urządzenia oraz początkowy poziom zbliżenia widoku, tuż po załadowaniu witryny do okna.

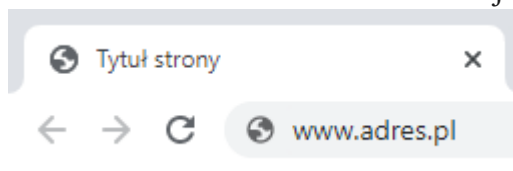
Tytuł bieżącej strony

Definicja tego elementu witryny wydaje się na pozór bardzo prosta – odpowiednią frazę wystarczy zamknąć wewnątrz podwójnego znacznika:

```
1 <title>Tytuł strony</title>
```

Warto jednak uzmysławić sobie, gdzie w praktyce, podczas codziennego użytkowania internetu, mamy do czynienia z tytułem witryny. Najważniejsze dwa miejsca to:

- **zakładka w przeglądarce internetowej** – po prawej stronie ikonki reprezentującej serwis w sposób wizualny, w przypadku gdy długość zakładki pozwala na zmieszczenie kilkunastu do kilkudziesięciu znaków, ujrzymy tytuł podstrony, który z oczywistych powodów powinien dobrze i celnie streścić zawartość tej otwartej karty przeglądarki:



Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

- **rezultaty wyszukiwania w Google** – tytuł strony jest za sprawą dużej, niebieskiej czcionki (lub fioletowej dla serwisów już odwiedzonych) mocno wizualnie wyeksponowany – internauci poszukujący informacji w sieci zdecydowanie zwrócą na niego uwagę; poza tym często na podstawie przekazu zawartego w tytule, podejmujemy decyzję o odwiedzeniu danego serwisu.

adres.pl › adres znalezionej strony

Tytuł strony

Fraza znaleziona w treści witryny, którą wpisał internauta w pole wyszukiwania Google. Absolutnie nie musi ona zgadzać się w żaden sposób z zawartością tagu meta description czy keywords...

Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

To właśnie kontekst **SEO** sprawia, że wybór zawartych pomiędzy znacznikami `<title></title>` wyrazów, to jedna z najważniejszych decyzji projektowych, której należy poświęcić należytą uwagę.

Ciekawostka

Wyszukiwarka respektuje tytuł ustawiony przez programistę w kodzie źródłowym. Nawet jeśli szukanej frazy w tytule nie ma, to Google i tak prezentuje w wynikach oryginalny tytuł, pobrany ze źródła – dokładnie tak, jak go zapisał na etapie

projektowania programista webowy (pamiętajmy jednak, że zbyt długie tytuły zostaną przycięte i zakończone wielokropkiem).

Jak zatem tworzyć dobre tytuły stron? Czy skupić się na często wyszukiwanych frazach, czy raczej na kreatywnych skojarzeniach, poczuciu humoru, a może na socjotechnice i filozofii *click-bait*?

To kwestia wiedzy i doświadczenia z zakresu marketingu webowego.

Jeżeli chodzi o kwestie czysto techniczne, warto zapamiętać, że tytuł powinien składać się łącznie maksymalnie z około 50-70 znaków. Dokładną wartość limitu określa firma Google w zmieniających się co roku wytycznych, przy czym zazwyczaj tytuły wyświetlane na urządzeniach mobilnych mogą być nieco dłuższe. W praktyce dobry, spójny tytuł powinien mieć aktualnie około 55-60 znaków.

Uwzględnianie zewnętrznych plików

Oprócz tytułu witryny, ważnym aspektem wizualnym na zakładce w przeglądarce jest ikonka kojarząca się z otwartą stroną:

```
1 <link rel="shortcut icon" type="image/x-icon" href="ikonka.ico">
```

Ta ikonka (określana jako *favicon*) widoczna będzie także na pasku zakładek oraz w skrótach na ekranie głównym pustej karty przeglądarki. Nie należy więc lekceważyć tego charakterystycznego elementu, kojarzącego się ze stroną.

Oprócz plików `.ico` możliwe jest także zastosowanie formatu np. `.svg` lub `.png`, dzięki odpowiedniej modyfikacji wartości atrybutu `type`.

Podpięcie zewnętrznego arkusza stylów, czyli zmiana wyglądu poszczególnych elementów składowych dokumentu:

```
1 <link rel="stylesheet" href="arkusz.css">
```

Podpięcie zewnętrznego skryptu napisanego w języku JavaScript, czyli dodanie funkcji której nie potrafią obsłużyć języki opisowe HTML i CSS:

```
1 <script src="skrypt.js"></script>
```

Poprawne działanie HTML5 w Internet Explorer

Standard HTML5 wprowadził wiele nowości, w tym użycie w sekcji body znaczników semantycznych, czyli określających przeznaczenie każdego elementu w kontekście SEO. Są to między innymi znaczniki: `<nav>`, `<header>`, `<footer>`, `<aside>`, `<article>`, `<section>`. W przypadku wykorzystania tych tagów, warto zadbać o ich poprawne wyświetlenie (emulowanie) w starszych przeglądarkach z rodziny Internet Explorer (dalej: IE):

```
1 <!--[if lt IE 9]>
2 <script src="html5shiv.min.js"></script>
3 <script src="respond.min.js"></script>
4 <![endif]-->
```

Oprócz skryptu `html5shiv.min.js` naprawiającego brak obsługi znaczników semantycznych w IE w wersjach poniżej dziewiątej, rozwiązano także problem braku rozpoznania zapisów `media queries` w CSS, które potrzebne są do prawidłowej obsługi responsywności w tych przeglądarkach.

Ważne!

W starszych wersjach IE zapewniamy emulowaną obsługę współczesnych mechanizmów wykorzystywanych na stronach internetowych: responsywności oraz semantyki.

Jak prawidłowo domykać znaczniki?

Na koniec wspomnijmy jeszcze o wiecznie żywej dyskusji na temat sposobu domykania znaczników pojedynczych w HTML. Czy należy zakończyć pojedyncze znaczniki znakiem `/` czy jednak nie? Innymi słowy, czy w kodzie źródłowym powinniśmy użyć zapisu:

```
1 <meta charset="utf-8">
```

czy jednak wykorzystać następujący sposób:

```
1 <meta charset="utf-8"/>
```

Zgodnie ze specyfikacją HTML5 w tagach pojedynczych obowiązuje brak kończącego znaku, aczkolwiek zapis znany z XHTML (czyli domknięcie tagu – wersja druga) nie spowoduje błędu w przeglądarce.

Wielu ludzi z przyzwyczajenia używa wersji starszej języka znaczników. Wprowadzanie nowych standardów sieciowych przebiega wolno, ale będąc na początku nauki

programowania, warto od razu wyrobić sobie nawyk użycia wersji pozbawionej domknięcia.

Słownik

kompatybilność wsteczna

sposób zastępowania aktualnie obowiązującej wersji standardu w taki sposób, aby pomimo wprowadzonych zmian oraz rozszerzeń w funkcjach, nowa wersja umożliwiała poprawną współpracę z obecnie istniejącą infrastrukturą oraz możliwie największą liczbą wersji poprzedzających

SEO

akronim od ang. *Search Engine Optimization* – ogół działań, których dokonują programiści webowi w celu poprawy widoczności witryny w wynikach wyszukiwania – w szczególności w najbardziej popularnej wyszukiwarce Google

Mapa myśli

Polecenie 1




Zapoznaj się z analizą szkieletu witryny. Poszczególne elementy rozrysowano w postaci mapy myśli. Dzięki temu lepiej zrozumiesz powiązania i role instrukcji w szablonie.

Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

Polecenie 2

Uzupełnij mapę myśli o istotne informacje dotyczące poszczególnych sekcji. Przykładowy wpis znajdziesz pod elementem „Sekcja head”.

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Wskaż, który z podanych znaczników powinien pojawić się jako pierwszy w sekcji nagłówkowej (head) witryny.

meta name="description" content="..."

title

meta name="author" content="..."

meta name="keywords" content="..."

meta charset="utf-8"

Ćwiczenie 2



Połącz w pary atrybuty HTML z odpowiadającymi im znacznikami, w których znajdują one zastosowanie.

lang

html

charset

meta

rel

link

src

script

Ćwiczenie 3



Uporządkuj podane instrukcje szablonu strony HTML, zgodnie z kolejnością ich zastosowania.

tytuł podstrony: `<title> ... </title>`



znacznik otwierający: `<html lang="pl">`



rozpoczęcie sekcji: `<body>`



deklaracja: `<!DOCTYPE html>`



rozpoczęcie sekcji: `<head>`



kodowanie znaków: `<meta charset="utf-8">`



Ćwiczenie 4



Wstaw w odpowiednie miejsca fragmenty meta-tagów tak, aby stworzyć znaczniki zdefiniowane zgodnie z ich przeznaczeniem w kodzie.

`<meta name=" " content="U nas kupisz zioła, herbaty i yerba mate">`

`<meta name=" " content="herbata, yerba, mate, zioła">`

`<meta name=" " content="Herbaciarnia Niezłe Ziółko">`

`<meta " "="X-Ua-Compatible" content="IE=edge,chrome=1">`

http-equiv

charset

description

keywords

viewport

author

Ćwiczenie 5



Wskaż wszystkie znaczniki, których obecność w kodzie przyczyni się do poprawy responsywności witryny internetowej, wyświetlanej na urządzeniach mobilnych.

```
<!--[if lt IE 9]><script src="respond.min.js"></script><![endif]-->
```

```
<link rel="shortcut icon" href="...">
```

```
<meta name="description" content="...">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<meta charset="...">
```

Ćwiczenie 6



Przypisz każdy znacznik HTML do sekcji, w której wystąpi w prawidłowo skonstruowanym kodzie źródłowym HTML.

sekcja <head>

```
<div> ... </div>
```

```
<title> ... </title>
```

sekcja <body>

```
<p> ... </p>
```

Ćwiczenie 7



Wskaż, który z zapisów oznacza warunkowe użycie w starszych przeglądarkach z rodziny Internet Explorer.

`<link rel="stylesheet" if="lt IE 9" href="...">`

`<!--[if lt IE 9]> ... <![endif]-->`

`<script src="..." use="lt IE 9"></script>`

`<!DOCTYPE html lt IE 9>`

Ćwiczenie 8



Wstaw w przedstawione deklaracje nazwy plików, którymi dysponuje programista webowy – tak, aby zostały poprawnie zinterpretowane przez przeglądarkę.

```
<link rel="shortcut icon" type="image/x-icon" href=" " >
<!--[if lt IE 9]> <script src=" " ></script> <![endif]-->
<link rel="stylesheet" href=" " >
<script src=" " ></script>
```

respond.min.js

skrypt.js

ikona.ico

arkusz.css

Dla nauczyciela

Autor: Mirosław Zelent

Przedmiot: Informatyka

Temat: Szkielet strony internetowej

Grupa docelowa:

Liceum ogólnokształcące i technikum, liceum ogólnokształcące, technikum, zakres podstawowy

Podstawa programowa:

Cele kształcenia – wymagania ogólne

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

III. Posługiwanie się komputerem, urządzeniami cyfrowymi i sieciami komputerowymi, w tym: znajomość zasad działania urządzeń cyfrowych i sieci komputerowych oraz wykonywania obliczeń i programów.

Treści nauczania – wymagania szczegółowe

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.

Zakres podstawowy. Uczeń:

3) przygotowuje opracowania rozwiązań problemów, posługując się wybranymi aplikacjami:

f) tworzy stronę internetową zgodnie ze standardami, wzbogaconą tabelami, listami, elementami dynamicznymi, posługuje się arkuszem stylów, korzysta z oprogramowania i serwisów przeznaczonych do tworzenia stron; potrafi opublikować własną stronę w internecie;

Kształtowane kompetencje kluczowe:

- kompetencje cyfrowe;
- kompetencje osobiste, społeczne i w zakresie umiejętności uczenia się;
- kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii.

Cele operacyjne (językiem ucznia):

- Przeanalizujesz podstawowe znaczniki HTML, które występują w szablonach współczesnych witryn.
- Stwierdzisz, jak wiele różnorodnych aspektów technicznych przekłada się na poprawność działania strony internetowej w przeglądarce.
- Prześledzisz budowę oraz przeznaczenie podstawowych znaczników HTML wykorzystywanych we współczesnych kodach źródłowych.
- Ocenisz skuteczność zabiegów technicznych, które wykorzystują programiści webowi podczas pracy nad szablonem strony.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimediu i ćwiczeń interaktywnych;
- metody aktywizujące.

Formy pracy:

- praca indywidualna;
- praca w parach;
- praca w grupach;
- praca całego zespołu klasowego.

Środki dydaktyczne:

- komputery z głośnikami, słuchawkami i dostępem do internetu;
- zasoby multimedialne zawarte w e-materiałach;
- tablica interaktywna/tablica, pisak/kreda.

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Szkielet strony internetowej”. Nauczyciel prosi uczniów o zapoznanie się z treściami w sekcji „Przeczytaj”.

Faza wstępna:

1. Przedstawienie tematu i celów zajęć.

2. Prowadzący prosi uczniów, aby zgłaszali swoje propozycje pytań do tematu. Jedna osoba może zapisywać je na tablicy. Gdy uczniowie wyczerpią swoje pomysły, a pozostały jakieś ważne kwestie do poruszenia, nauczyciel je dopowiada.

Faza realizacyjna:

1. **Praca z tekstem.** Jeżeli przygotowanie uczniów do lekcji jest niewystarczające, nauczyciel prosi o indywidualne zapoznanie się z treścią zawartą w sekcji „Przeczytaj”. Każdy uczestnik zajęć podczas cichego czytania wynotowuje najważniejsze kwestie poruszane w tekście.
2. **Praca z multimediami.** Uczniowie w zespołach dwuosobowych zapoznają się z treścią polecenia nr 1: „Zapoznaj się z analizą szkieletu witryny. Poszczególne elementy rozrysowano w postaci mapy myśli. Dzięki temu lepiej zrozumiesz powiązania i role instrukcji w szablonie.” z sekcji „Mapa myśli” i wspólnie analizują kolejne kroki rozwiązania postawionego problemu.
3. **Ćwiczenie umiejętności.** Uczniowie wykonują ćwiczenia nr 1-8 z sekcji „Sprawdź się”. Nauczyciel sprawdza poprawność wykonanych zadań, omawiając je wraz z uczniami.
4. Uczniowie samodzielnie wykonują ćwiczenie w sekcji „Sprawdź się”. Chętne lub wybrane osoby przedstawiają rozwiązania i je omawiają.

Faza podsumowująca:

1. Nauczyciel ponownie wyświetla na tablicy temat lekcji zawarty w sekcji „Wprowadzenie” i inicjuje krótką rozmowę na temat zrealizowanych celów (czego uczniowie się nauczyli).

Praca domowa:

1. Uczniowie piszą opis przykładowej strony internetowej, składający się z około 155-160 znaków. Opis powinien zawierać kluczowe frazy, jak również bezpośrednio zwracać się do internauty, aby zainteresować go właśnie tą konkretną witryną.

Wskazówki metodyczne:

- Treści w sekcji „Przeczytaj” można wykorzystać jako podsumowanie i utrwalenie wiedzy uczniów.