

Tworzenie stron WWW z wykorzystaniem stylów i elementów programowania

1. Jak zapisana jest strona WWW?
2. Przykład tworzenia strony w języku HTML
 - 2.1. Stosowanie podstawowych znaczników języka HTML
 - 2.2. Tworzenie łączy hipertekstowych
 - 2.3. Tworzenie podstrony
 - 2.4. Wstawianie obrazów
 - 2.5. Wstawianie tabel
 - 2.6. Stosowanie kolorów
3. Kaskadowe arkusze stylów CSS
 - 3.1. Najczęściej wykorzystywane atrybuty CSS i sposoby określania ich wartości
 - 3.2. Formatowanie hiperłączy
 - 3.3. Tło strony
 - 3.4. Klasy elementów
4. Elementy dynamiczne na stronie – skrypty w języku JavaScript
5. Publikowanie i promowanie stron internetowych w Internecie
 - 5.1. Publikowanie strony WWW
 - 5.2. Promowanie strony WWW



Warto powtórzyć

1. Przypomnij pojęcia: strona WWW, witryna internetowa, adres WWW.
2. Czym są hiperłącza?
3. Jakie znasz znaczniki języka HTML?
4. W jaki sposób tworzyliśmy i jak wykorzystywaliśmy style w edytorze tekstu?

1. Jak zapisana jest strona WWW?

Do opisu stron WWW stosuje się specjalny język – **HTML** (ang. *Hypertext Markup Language*, czyli język znaczników hipertekstowych). Serwer WWW przesyła do przeglądarki internetowej kod HTML. Przeglądarka interpretuje otrzymany kod i wyświetla stronę (rys. 1.).

Serwer

Plik HTML

```
<html>
<body>
<p>Witaj!</p>
...
```

Internet

Klient

```
<html>
<body>
<p>Witaj!</p>
...
```

wyświetlanie
przez
przeglądarkę

Witaj!

Rys. 1. Schemat działania statycznej strony WWW



Plik w języku HTML to plik tekstowy, zapisany z rozszerzeniem *htm* bądź *html*. Jeden plik w języku HTML opisuje jedną stronę **WWW**, natomiast **witryna internetowa** składa się najczęściej z wielu (od kilkudziesięciu do nawet kilkuset tysięcy) stron, wraz z dodatkowymi danymi: obrazami, animacjami, dźwiękami itp.

Oprócz języka HTML do opisywania wyglądu i funkcjonowania witryn internetowych wykorzystuje się **kaskadowe arkusze stylów** (CSS – ang. *Cascading Style Sheet*) oraz skryptowy język programowania **JavaScript**. Niektóre strony internetowe posiadają też dodatkowe (coraz rzadziej spotykane) elementy (np. odtwarzacz multi-mediów, interaktywne gry), wykonane w technologiach Java lub Adobe Flash.

Do utworzenia prostej witryny wystarczą: edytor plików tekstowych lub edytor tekstu, edytor grafiki oraz program umożliwiający przesyłanie na serwer WWW (klient FTP) plików składających się na witrynę. Rozbudowane witryny tworzy się zwykle z wykorzystaniem specjalistycznych narzędzi. Witryny takie korzystają również z baz danych i języków skryptowych.

Przykładowe narzędzia	Opis, przeznaczenie
edytory plików tekstowych	wymagają znajomości znaczników HTML, m.in.: - dostępny w systemie Windows program Notatnik; - dostępne bezpłatnie w Internecie (np. Notepad++, Programmer's, Notepad, PSPad, BlueFish), wyróżniają (różnymi kolorami) znaczniki i inne elementy języka HTML
edytor tekstu Microsoft Word	tworzenie strony bez znajomości języka HTML (w oknie edytora widzimy stronę tak, jak będzie wyglądać w przeglądarce); umożliwia zapis dokumentu w formacie HTML (opcja Zapisz jako/Strona sieci Web); zawiera narzędzia pozwalające na przygotowanie strony WWW na podstawie szablonu lub istniejącego dokumentu (udostępnia na pasku narzędzi specjalne polecenia, które automatycznie tworzą odpowiednie znaczniki HTML)

systemy zarządzania treścią (CMS)	pozwalają na utworzenie witryny internetowej i zarządzanie nią za pomocą przeglądarki internetowej, bez konieczności poznawania języka HTML; nie wymagają szczegółowej wiedzy dotyczącej technologii internetowych; rozpowszechniane na licencji GNU GPL, Joomla!, Drupal, WordPress
serwisy przeznaczone do tworzenia stron	rozbudowane serwisy, pozwalające tworzyć profesjonalne strony
specjalistyczne narzędzia	przeznaczone dla osób zajmujących się profesjonalnie tworzeniem stron internetowych; podczas tworzenia witryny wymagana jest zazwyczaj znajomość baz danych i skryptowych języków programowania

Tabela 1. Wybrane narzędzia do tworzenia stron

Etapy wykonywania strony

1. Określenie tematyki strony internetowej.
2. Planowanie, projektowanie:
 - treści, w tym elementów grafiki (rysunków, zdjęć), które chcemy zamieścić na stronie,
 - ustalenie sposobu rozmieszczenia grafiki względem tekstu,
 - ustalenie kolorystyki strony (tła, tekstu, elementów graficznych),
 - parametrów formatowania tekstu,
 - numeracji, wyliczeń, tabel,
 - systemu nawigacji (wstawienie hiperłączy do innych stron, w tym podstron).
3. Wykonanie strony – utworzenie kodu w języku HTML w oparciu o projekt.
4. Sprawdzenie poprawności tekstów i funkcjonowania strony.



Ćwiczenie 1. Porównujemy wygląd witryn

Porównaj kilka najczęściej odwiedzanych przez ciebie witryn. Zwróć uwagę na strukturę strony: tytuł, rozmieszczenie tekstu, przejrzystość, grafikę, sposób przechodzenia do innych stron.



Ćwiczenie 2. Oglądamy źródło strony

Wejdź na wybraną stronę internetową, np. swojej szkoły. Obejrzyj źródło strony, korzystając z odpowiedniej opcji przeglądarki (dostępnej zwykle w menu kontekstowym strony).

2. Przykład tworzenia strony w języku HTML

Zamierzamy utworzyć stronę internetową o Wyspach Kanaryjskich. Na stronie chcemy umieścić fotografie i informacje o najciekawszych zabytkach lub ważnych obiektach, wskazówki dla turystów dotyczące godzin otwarcia muzeów oraz opłat za zwiedzanie, ciekawostki historyczne, przyrodnicze itp. Jak wykonać taką stronę, korzystając z edytora plików tekstowych?

2.1. Stosowanie podstawowych znaczników języka HTML



Aby utworzyć stronę w języku HTML, należy zapisać jej kod w pliku z rozszerzeniem *htm* lub *html*. Taki plik możemy następnie otworzyć w przeglądarce internetowej, aby zobaczyć efekt naszej pracy.

Do utworzenia przykładowej strony internetowej o Wyspach Kanaryjskich wykorzystamy program Notepad++. Aby znaczniki HTML były wyróżnione kolorem, należy w menu **Składnia** wybrać z listy wyboru literę **H**, a następnie **HTML**.

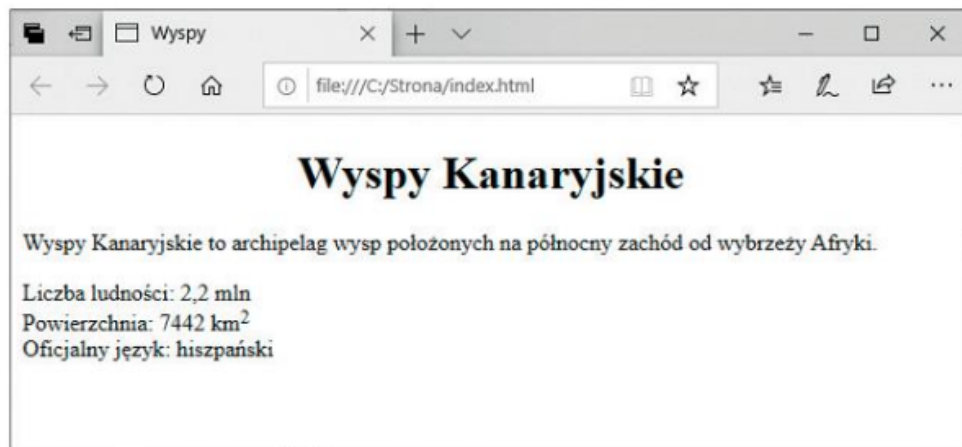
```
1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Wyspy</title>
6   </head>
7
8   <body>
9     <h1 align="center">Wyspy Kanaryjskie</h1>
10    <p>Wyspy Kanaryjskie to archipelag wysp położonych na północny zachód
11    od wybrzeży Afryki.</p>
12    <p></p>
13    <p>Liczba ludności: 2,2 mln<br>
14    Powierzchnia: 7442 km<sup>2</sup><br>
15    Oficjalny język: hiszpański</p>
16  </body>
17
18 </html>
19
20
21
```

informacja dla przeglądarki o standardzie, w jakim wykonano dokument – dla standardu HTML5 stosuje się

`<!DOCTYPE html>`

informacja o kodowaniu znaków w standardzie kodowania UTF-8

Rys. 2a. Kod źródłowy strony widoczny w oknie programu Notepad++ – ćwiczenie 3.



Rys. 2b. Strona widoczna w przeglądarce – ćwiczenie 3.

Tworząc akapit, można określić sposób jego wyrównania (podobnie jak w edytorze tekstu: do lewej, do środka, do prawej, wyjustowany). Służy do tego atrybut `align` (z ang. *wyrównywać*), przyjmujący jedną z wartości wymienionych w tabeli 2. Atrybut `align` jest wykorzystywany także w innych znacznikach języka HTML.

Postać atrybutu <code>align</code>	Funkcja
<code>align="left"</code>	wyrównanie do lewej
<code>align="center"</code>	wyrównanie do środka
<code>align="right"</code>	wyrównanie do prawej
<code>align="justify"</code>	justowanie

Tabela 2. Wartości atrybutu `align`



Przykład 1. Stosowanie atrybutu `align`

```
<p align="center">Wyrównanie akapitu do środka strony</p>
```

W języku HTML mamy do dyspozycji sześć poziomów tytułów, w kolejności od najważniejszych do najmniej ważnych.

```
<h1>Tytuł pierwszego poziomu</h1>
...
<h6>Tytuł szóstego poziomu</h6>
```

`<h1 align="center">Tytuł</h1>` spowoduje wyśrodkowanie tytułu względem szerokości strony.

Uwagi:

- Domyślnie (jeżeli nie użyjemy atrybutu `align`) akapity i tytuły wyrównywane są do lewej strony.
- W języku HTML wartości atrybutów wpisuje się w znakach cudzysłowu.

Jeżeli strona ma zostać umieszczona w Internecie, należy pamiętać o następujących zasadach nazywania plików:

- nie używamy w nazwach plików polskich znaków diakrytycznych,
- nie używamy w nazwach plików spacji,
- używamy tylko małych liter (większość serwerów WWW oparta jest na systemie operacyjnym Linux, który w nazwach plików rozróżnia małe i wielkie litery).



Ćwiczenie 3. Zapisujemy kod źródłowy strony w programie Notepad++

1. Utwórz folder o nazwie *Strona*, w którym zapiszesz swoją stronę WWW.
2. Uruchom program Notepad++ i przepisz kod strony podany na rysunku 2a.
3. Wyjaśnij znaczenie znaczników umieszczonych w kodzie, wykorzystując wiadomości z lekcji informatyki ze szkoły podstawowej.
3. Zapisz kod strony w pliku o nazwie *index.html* i otwórz go w przeglądarce internetowej.



Uwaga

Strona o Wyspach Kanaryjskich jest tylko przykładem – możesz na jej wzór od razu wykonać podobną stronę na wybrany przez siebie temat.

Wskazówki:

- Aby w programie Notepad++ zastosować kodowanie UTF-8, należy z menu **Format** wybrać opcję **Koduj w UTF-8**.
- Przy zapisywaniu pliku w programie Notepad++ w polu **Zapisz jako typ** należy wybrać opcję **All types (Wszystkie pliki)**.

Czasami zachodzi potrzeba rozdzielenia fragmentów tekstu poziomą linią. Do utworzenia takiej linii służy znacznik `<hr>`. Znacznik ten posiada także dodatkowe atrybuty.

Nazwa atrybutu	Funkcja
<code>align="wyrównanie"</code>	Określa położenie linii względem strony. Możliwe wartości: <code>left</code> , <code>right</code> , <code>center</code> .
<code>size="grubość"</code>	Określa grubość linii (w pikselach).
<code>width="szerokość"</code>	Określa szerokość linii. Szerokość może być określona w pikselach lub procentowo, względem szerokości elementu nadrzędnego (np. całej strony).

Tabela 3. Atrybuty znacznika `<hr>`

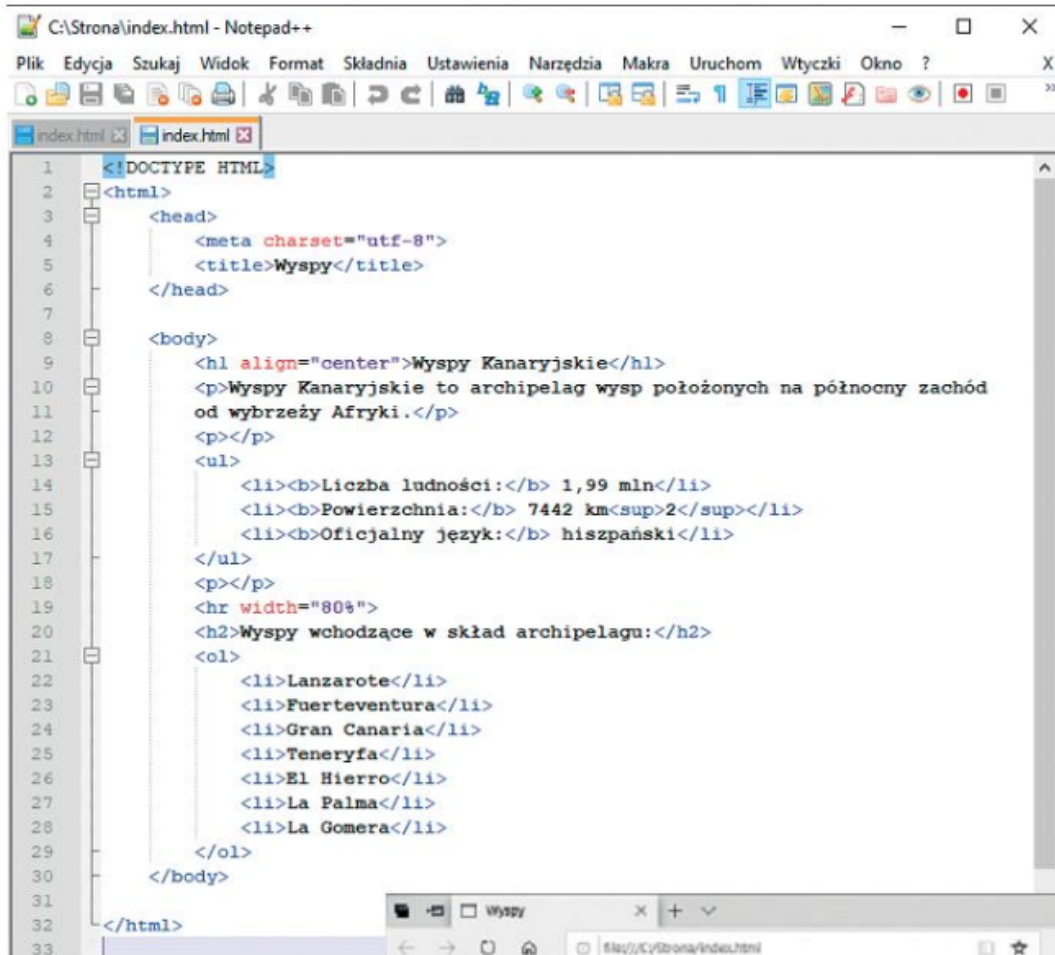
HTML pozwala na tworzenie list uporządkowanych (numerowania) lub nieuporządkowanych (wypunktowania). Przy tworzeniu list numerowanych komputer wstawia automatycznie kolejne numery punktów. Listy można też zagnieżdżać wewnątrz siebie.

Lista uporządkowana ma postać:

```
<ol>
  <li>Element listy</li>
  <li>Element listy</li>
  ...
</ol>
```

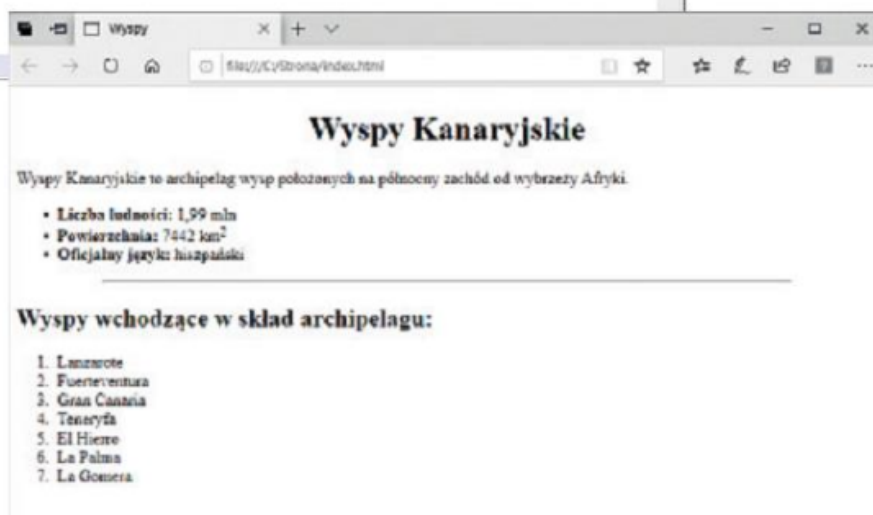
Lista nieuporządkowana ma postać:

```
<ul>
  <li>Element listy</li>
  <li>Element listy</li>
  ...
</ul>
```



```
1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Wyspy</title>
6   </head>
7
8   <body>
9     <h1 align="center">Wyspy Kanaryjskie</h1>
10    <p>Wyspy Kanaryjskie to archipelag wysp położonych na północny zachód
11    od wybrzeży Afryki.</p>
12    <p></p>
13    <ul>
14      <li><b>Liczba ludności:</b> 1,99 mln</li>
15      <li><b>Powierzchnia:</b> 7442 km<sup>2</sup></li>
16      <li><b>Oficjalny język:</b> hiszpański</li>
17    </ul>
18    <p></p>
19    <hr width="80%">
20    <h2>Wyspy wchodzące w skład archipelagu:</h2>
21    <ol>
22      <li>Lanzarote</li>
23      <li>Fuerteventura</li>
24      <li>Gran Canaria</li>
25      <li>Tenerify</li>
26      <li>El Hierro</li>
27      <li>La Palma</li>
28      <li>La Gomera</li>
29    </ol>
30  </body>
31 </html>
```

Rys. 3a. Kod źródłowy strony widoczny w oknie programu Notepad++ – ćwiczenie 4.



Rys. 3b. Strona widoczna w przeglądarce – ćwiczenie 4.



Ćwiczenie 4. Dodajemy do strony listę nieuporządkowaną, listę uporządkowaną i poziomą linię

1. Otwórz plik *index.html*, zapisany w ćwiczeniu 3. w folderze *Strona*. Dodaj do strony o Wyspach Kanaryjskich:
 - listę nieuporządkowaną, zawierającą dane o liczbie ludności, powierzchni i języku oficjalnym (urzędowym),
 - listę uporządkowaną, zawierającą nazwy wysp wchodzących w skład archipelagu.Listy oddziel od siebie linią poziomą o szerokości 80% strony.
2. Zapisz plik pod tą samą nazwą.

2.2. Tworzenie łączy hipertekstowych

Najważniejszą cechą języka HTML, decydującą o jego popularności i wszechstronności, jest możliwość tworzenia **hiperłączy**. Przez hiperłącze rozumiemy wyróżniony w pewien sposób obiekt (fragment tekstu, rysunek), po którego wybraniu (np. przez kliknięcie myszą) w oknie przeglądarki zostanie wyświetlony nowy dokument.

Do tworzenia łączy hipertekstowych służy znacznik `<a>`. Istnieją dwie różne składnie tego polecenia, o odmiennych zastosowaniach.

Tworzenie hiperłącza do innego dokumentu

Znacznik przyjmuje postać:

```
<a href="obiekt_docelowy">Treść hiperłącza</a>
```

Przez treść hiperłącza należy rozumieć tekst lub rysunek aktywujący hiperłącze (po kliknięciu myszą). Natomiast *obiekt_docelowy* to nazwa bądź adres URL pliku, który ma zostać otwarty przez przeglądarkę.

Dodatkowy atrybut `target="nazwa_ramki"` umożliwia otwarcie strony docelowej w innym oknie (na innej karcie) przeglądarki.



Przykład 2. Tworzenie hiperłączy

```
<a href="index2.html">Dalej</a>
```

Po kliknięciu tekstu *Dalej* w oknie przeglądarki zostanie wyświetlona zawartość pliku *index2.html*.

```
<a href="https://www.gov.pl/web/edukacja" target="_blank">Ministerstwo Edukacji Narodowej</a>
```

Po kliknięciu tekstu *Ministerstwo Edukacji Narodowej* otworzy się nowe okno (nowa karta) przeglądarki, w którym zostanie wyświetlona strona Ministerstwa Edukacji Narodowej.

```
<a href="index2.html"></a>
```

Aktywowanie hiperłącza następuje poprzez kliknięcie rysunku *rys.gif*.

Tworzenie kotwic wewnętrznych

Przy długich dokumentach zdarza się, że potrzebujemy wyświetlić określone miejsce takiego dokumentu (np. jeżeli tekst podzielony jest na rozdziały – początek rozdziału).

Język HTML pozwala na zaznaczanie wybranych miejsc za pomocą kotwic (ang. *anchor*). Kotwicę tworzymy poleceniem: `<a name="nazwa_kotwicy"></a>`, gdzie *nazwa_kotwicy* to dowolny, wybrany przez nas ciąg znaków.

Aby następnie przejść do zaznaczonego w ten sposób miejsca, tworzymy hiperłącze poleceniem: `<a href="plik.html#nazwa_kotwicy">Treść hiperłącza</a>`, gdzie *plik.html* to nazwa pliku, w którym wstawiliśmy kotwicę.

Jeżeli odnośnik ma przenieść nas do kotwicy w tym samym dokumencie, nazwę pliku można pominąć: `<a href="#nazwa_kotwicy">Treść hiperłącza</a>`.

2.3. Tworzenie podstrony

Na stronie głównej umieścimy ogólne informacje o Wyspach Kanaryjskich, a szczegółowe o poszczególnych wyspach przedstawimy na podstronach. Na stronie głównej musi być umieszczone hiperłącze prowadzące do podstrony, a na podstronie – hiperłącze przenoszące z powrotem do strony głównej. Podstronę najlepiej zapisać w pliku z rozszerzeniem *html* w tym samym folderze, co strona główna.



Ćwiczenie 5. Tworzymy podstronę i dodajemy hiperłącze do strony głównej

1. Utwórz w programie Notepad++ podstronę o wybranej wyspie, np. Lanzarote. Umieść na stronie w nagłówku tytuł, a pod nim trzy zdania opisujące wyspę (możesz utworzyć trzy akapity lub zastosować listę).
2. Zapisz podstronę w pliku pod nazwą *lanzarote.html* w folderze *Strona*.
3. Na stronie głównej umieść hiperłącze na nazwie „Lanzarote”, prowadzące do podstrony *lanzarote.html*, a na podstronie – w lewym górnym rogu ikonę z linkiem powrotu do strony głównej.
4. Zapisz obydwa pliki: stronę główną i podstronę. Odśwież stronę w przeglądarce. Sprawdź poprawność działania hiperłączy.

2.4. Wstawianie obrazów

Podstawowe formaty obrazów wykorzystywanych na stronach WWW to: GIF, PNG, JPEG.



Aby umieścić na stronie rysunek, należy zastosować znacznik

``. Znacznik ten nie ma postaci zamykającej.

Na przykład: ``

Na stronie można także umieścić dowolny rysunek z Internetu, pod warunkiem, że znamy jego adres URL. Wykorzystując na swojej stronie obrazy pochodzące z innych stron internetowych, należy pamiętać o przestrzeganiu praw autorskich (temat D1).

Znacznik `<img>` posiada wiele dodatkowych atrybutów, które można wykorzystać do określenia położenia obrazu na stronie. Najważniejsze z nich zebrano w tabeli 4.

Nazwa atrybutu	Funkcja
<code>align="wyrównanie"</code>	Określa położenie rysunku względem strony. Możliwe wartości: <code>left</code> , <code>right</code> , <code>center</code> .
<code>alt="tekst"</code>	Określa tekst pokazywany w sytuacji, gdy niemożliwe jest wyświetlenie obrazu.
<code>width="szerokość"</code>	Określa szerokość rysunku. Szerokość powinna być podana w pikselach.
<code>height="wysokość"</code>	Określa wysokość rysunku. Wysokość może być podana w pikselach (względem wysokości oryginalnego rysunku).

Tabela 4. Wybrane atrybuty znacznika `<img>`

Na przykład:

```

```

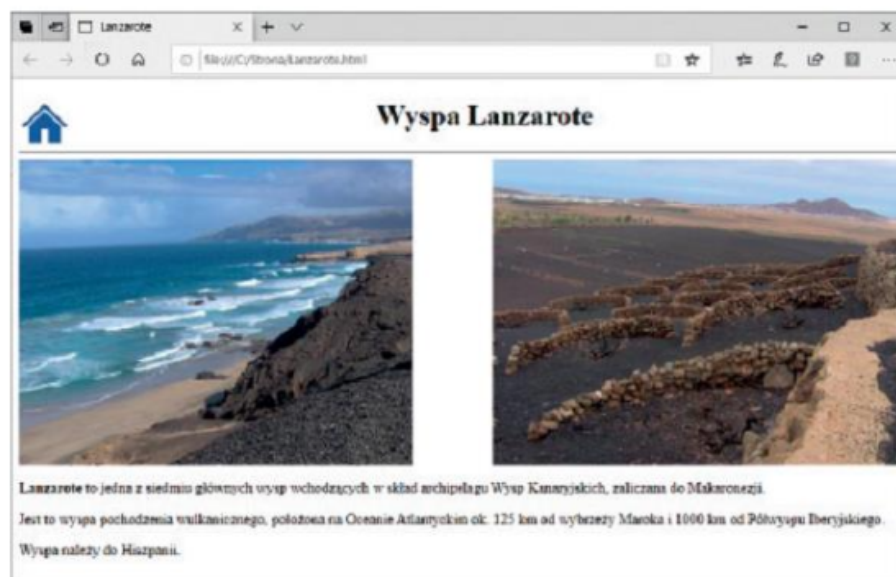
wyświetla obraz wyrównany do lewej strony, o szerokości 300 i wysokości 200 pikseli.



Ćwiczenie 6. Dodajemy obrazy do podstrony

1. Otwórz plik *lanzarote.html*, zapisany w ćwiczeniu 5. Dodaj do strony przed tekstem opisującym wyspę dwa wybrane obrazy (rysunki, zdjęcia).
2. Zapisz plik pod tą samą nazwą.

Wskazówka: Aby mieć pewność, że tekst będzie zaczynał się pod obrazem, należy poprzedzić tekst znacznikiem `<br>` z atrybutem `clear="all"`:
`<br clear="all">`.



Rys. 4. Przykładowy wygląd podstrony z wstawionymi zdjęciami i ikoną powrotu do strony głównej

2.5. Wstawianie tabel

W języku HTML **tabele** składają się z **wierszy**, a wiersze – z **komórek**. Granice tabeli określają znaczniki `<table> ... </table>`, granice wierszy tabeli – znaczniki `<tr> ... </tr>`, komórek tabeli – znaczniki `<td> ... </td>`.



Przykład 3. Tworzenie tabeli

Utworzymy tabelę składającą się z dwóch wierszy – każdy z dwóch komórek.

```
<table>
  <tr>
    <td>Wiersz 1, komórka 1</td>
    <td>Wiersz 1, komórka 2</td>
  </tr>
  <tr>
    <td>Wiersz 2, komórka 1</td>
    <td>Wiersz 2, komórka 2</td>
  </tr>
</table>
```

Dodatkowe atrybuty pozwalają określić sposób wyświetlania tabeli. Najważniejsze z nich zebrano w tabeli 5.

Nazwa atrybutu	Funkcja
Znacznik <table>	
align="wyrównanie"	Określa położenie tabeli względem strony. Możliwe wartości: left, right, center.
width="szerokość"	Określa szerokość tabeli. Szerokość może być podana w pikselach lub procentowo, względem szerokości elementu nadrzędnego (np. całej strony).
border="grubość"	Określa grubość (w pikselach) obramowania tabeli i komórek tabeli. Jeżeli atrybut nie zostanie użyty lub będzie miał wartość 0, linie nie będą wyświetlane.
cellspacing="odstęp"	Określa odstęp (w pikselach) pomiędzy komórkami tabeli.
cellpadding="odstęp"	Określa odstęp (w pikselach) pomiędzy krawędziami komórek tabeli a zawartością komórek.
Znacznik <td>	
align="wyrównanie"	Określa położenie zawartości komórki względem komórki. Możliwe wartości: left, right, center, justify.
width="szerokość"	Określa szerokość komórki. Szerokość może być podana w pikselach lub procentowo (względem szerokości tabeli). Raz określona szerokość komórki obowiązuje dla kolejnych wierszy.
colspan="liczba_komórek"	Powoduje połączenie komórki z komórkami po prawej stronie komórki; liczba_komórek określa całkowitą liczbę połączonych komórek.
rowspan="liczba_komórek"	Powoduje połączenie komórki z komórkami umieszczonymi poniżej komórki; liczba_komórek określa całkowitą liczbę połączonych komórek.

Tabela 5. Atrybuty znaczników <table> i <td>

Dane statystyczne Lanzarote		
Powierzchnia	845,9 km ²	
Populacja	Gęstość zaludnienia	227,6 os/km ²
	Liczba ludności	142 132

Rys. 5. Przykładowy wygląd tabeli na podstronie



Ćwiczenie 7. Dodajemy tabelę do podstrony

1. Otwórz plik podstrony *lanzarote.html*, zapisany w ćwiczeniu 6. Dodaj tabelę, w której wierszach umieścisz podstawowe informacje o wyspie: dane dotyczące powierzchni, populacji itp. – jak pokazano na rysunku 5.
2. Zapisz plik z podstroną pod tą samą nazwą.

Wskazówka: Aby zapisać km^2 , użyj znacznika `<sup> ... </sup>`: `km<sup>2</sup>`.

2.6. Stosowanie kolorów

Wielu elementom strony WWW można nadawać odrębne kolory. Kolor można określić albo za pomocą angielskiej nazwy, albo podając jego wartość w systemie RGB. Na wartość w systemie RGB składają się, poprzedzone krzyżykiem, trzy liczby zapisane w systemie szesnastkowym, odpowiadające składowym koloru (czerwonej, zielonej i niebieskiej). W ten sposób można opisać dowolną z ponad 24 milionów barw dostępnych na ekranie monitora.



Przykład 4. Stosowanie kolorów

```
<font color="red">czerwony tekst</font>
<font color="#ff0000">czerwony tekst</font>
```



Ćwiczenie 8. Stosujemy kolory na stronie

1. Znajdź w Internecie w specyfikacji języka HTML pełen wykaz kolorów i opis tworzenia kolorów w systemie RGB.
2. Na stronie głównej o Wyspach Kanaryjskich, zapisanej w ćwiczeniu 5. i na podstronie zapisanej w ćwiczeniu 7., zmień według własnego pomysłu kolory nagłówków. Zapisz pliki pod tymi samymi nazwami.

3. Kaskadowe arkusze stylów CSS

Kaskadowe arkusze stylów pozwalają opisywać sposób wyświetlania stron WWW. Stanowią uzupełnienie znaczników języka HTML, udostępniając jednocześnie znacznie szerszy zakres możliwości w zakresie kształtowania wyglądu strony.

Przykładowa reguła zapisana w języku CSS może mieć następującą postać:

```
p { font-family: Verdana; font-size: 11px; }
```

Przed nawiasami klamrowymi umieszczamy **selektor**, określający elementy strony, których dotyczy dana definicja. Jeżeli definicja ma dotyczyć kilku różnych typów elementów, ich nazwy oddzielamy przecinkami. W nawiasach klamrowych podajemy nazwy **atrybutów** i ich wartości (po wartości atrybutu należy umieścić średnik). Podana przykładowa reguła dotyczy akapitów (znaczniki `<p>`) i oznacza, że będą one wyświetlane z użyciem czcionki Verdana o rozmiarze 11 pikseli. Przykład 5. pokazuje, w jaki sposób można dołączyć podaną wyżej regułę do dokumentu HTML.



Przykład 5. Dołączanie stylów CSS do dokumentów HTML

Istnieją dwa sposoby dołączania stylów CSS do dokumentów HTML. Pierwszy z nich polega na umieszczeniu stylów bezpośrednio w dokumencie, wewnątrz znacznika `<style>`, dodanego w nagłówku dokumentu HTML.

```
<head>
  <style type="text/css">
    p { font-family: Verdana; font-size: 11px; }
  </style>
</head>
```

Drugi sposób polega na umieszczeniu reguł CSS w osobnym pliku. W nagłówku dokumentu HTML dodajemy wtedy odwołanie do pliku stylów CSS.

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

W przypadku takiego odwołania reguły CSS powinny być umieszczone w pliku o nazwie *style.css*, zapisanym w tym samym katalogu co dokument HTML. Nazwa i położenie pliku z arkuszem stylów mogą być dowolne – określamy je za pomocą atrybutu `href` w podobny sposób, w jaki wskazywaliśmy elementy docelowe hiperłączy.



Reguły CSS zawarte wewnątrz znacznika `<style>` umieszczonego w dokumencie mają pierwszeństwo przed regułami dołączanymi z zewnętrznych plików za pomocą znacznika `<link>`.



Ćwiczenie 9. Stosujemy style do zmiany parametrów czcionki

1. Używając jednego ze sposobów opisanych w przykładzie 5., dodaj do strony o Wyspach Kanaryjskich regułę CSS określającą czcionkę akapitów. Wypróbuj różne wielkości i kroje czcionek. Następnie zmodyfikuj regułę tak, aby zamiast akapitów dotyczyła całego dokumentu.
2. Zapisz plik pod tą samą nazwą.

Wskazówki:

- Aby zmienić czcionkę w akapitach i listach, można wpisać polecenie w taki sposób:

```
p, ul, ol { font-family: Arial; font-size: 12 px; }
```

Polecenie to zmieni czcionkę w akapitach, listach uporządkowanych i nieuporządkowanych na Arial o rozmiarze 12 pikseli.

- Aby zmienić czcionkę w całym dokumencie, należy dodać regułę dotyczącą znacznika `<body>`.

3.1. Najczęściej wykorzystywane atrybuty CSS i sposoby określania ich wartości

Nazwa atrybutu	Funkcja
<code>width: szerokość</code>	Określa szerokość elementu.
<code>height: wysokość</code>	Określa wysokość elementu.
<code>margin: wielkość_marginesów</code>	Określa wielkość marginesów dla elementu, w kolejności: górny, prawy, dolny, lewy.
<code>border: grubość styl kolor</code>	Określa grubość, styl i kolor obramowania.
<code>font-family: rodzina_krojów</code>	Określa rodzinę krojów pisma dla tekstu.
<code>font-size: wielkość</code>	Określa wielkość pisma dla tekstu.
<code>font-style: lista_atrybutów</code>	Określa atrybuty tekstu. Możliwe wartości: <code>normal</code> , <code>italic</code> , <code>oblique</code> .
<code>font-weight: pogrubienie</code>	Określa pogrubienie tekstu. Możliwe wartości: <code>normal</code> , <code>bold</code> , <code>bolder</code> , <code>lighter</code> lub liczba określająca grubość pisma.
<code>color: kolor</code>	Określa kolor elementu.
<code>background-color: kolor_tła</code>	Określa kolor tła elementu.

Tabela 6. Najczęściej wykorzystywane atrybuty CSS

Sposób określania wartości atrybutów	
px	Określa wielkość w pikselach. Przykład: { font-size: 10px; }
pt	Określa wielkość w punktach. Przykład: { font-size: 12pt; }
%	Określa wielkość w procentach (względem elementu nadrzędnego). Przykład: { width: 50%; }
in, cm, mm	Określa wielkość odpowiednio w calach, centymetrach lub milimetrach. Przykład: { margin: 2cm 4cm 3cm 4cm; }
em	Określa wielkość pisma relatywnie do aktualnie ustawionej wielkości pisma. Przykład: { font-size: 2em; }

Tabela 7. Sposoby określania wartości atrybutów

3.2. Formatowanie hiperłączy

Jednym z elementów, których wygląd często modyfikuje się za pomocą stylów CSS, jest hiperłącze. Arkusze stylów CSS pozwalają na osiągnięcie efektów niemożliwych do uzyskania przy zastosowaniu jedynie znaczników HTML.



Przykład 6. Zmiana koloru hiperłączy

Arkusze stylów CSS pozwalają łatwo zmienić wygląd hiperłączy znajdujących się na stronie. Aby zmienić kolory hiperłączy, wystarczy dodać do arkusza stylów następujące trzy reguły:

```
a { color: #000080; }
a:hover { color: #ff0000; }
a:visited { color: #808080; }
```

Oznaczają one, że domyślnie hiperłącza będą koloru granatowego (#000080). Po najechaniu (ang. *hover*) na hiperłącze kursorem jego kolor zmieni się na czerwony (#ff0000). Odwiedzone (ang. *visited*) hiperłącza będą koloru ciemnoszarego (#808080).



Przykład 7. Ustawianie podświetlania hiperłączy po najechaniu kursorem myszy

Ciekawy efekt można uzyskać, zmieniając kolor tła hiperłącza, na które najedzie kursor. Na przykład reguła:

```
a:hover { background-color: #ffecc; }
```

oznacza, że gdy kursor najedzie na hiperłącze, będzie ono podświetlane na żółto (#ffecc).



Ćwiczenie 10. Zmieniamy wygląd hiperłączy

1. Korzystając z przykładów 6. i 7., zmień wygląd hiperłączy na stronie głównej o Wyspach Kanaryjskich. Dobierz dla hiperłączy wybrane przez siebie kolory.
2. Zapisz plik pod tą samą nazwą.

3.3. Tło strony



Przykład 8. Zmienianie tła strony

Aby zmienić tło strony na jasnoniebieskie, dodajemy do arkusza stylów CSS definicję:

```
body { background-color: #d5f3ff; }
```

Możliwe jest także ustawienie jako tła strony obrazka.

```
body { background-image: url(obrazek.gif); }
```

W podanej formule `obrazek.gif` to nazwa obrazka, który ma zostać użyty jako tło. Warto zauważyć, że jeżeli obrazek tła jest mniejszy od okna przeglądarki, będzie on powielany w pionie i poziomie.



Ćwiczenie 11. Zmieniamy tło strony

1. Zmień tło strony głównej o Wyspach Kanaryjskich na wybrany przez siebie kolor lub obrazek.
2. Zapisz plik pod tą samą nazwą.

3.4. Klasy elementów

Przy tworzeniu stron internetowych często zachodzi potrzeba zmiany wyglądu jedynie wybranych elementów strony – na przykład, gdy chcemy, aby niektóre akapity były zapisane inną czcionką od pozostałych. W takiej sytuacji jednym z rozwiązań jest zastosowanie znacznika `<font>`, jednak znacznik ten musielibyśmy powtarzać dla każdego akapitu, co spowodowałoby wydłużenie kodu HTML i utrudniłoby jego późniejszą modyfikację. Znacznie lepszym rozwiązaniem jest zastosowanie klas CSS.



Przykład 9. Stosowanie klas CSS

W arkuszu stylów CSS definiujemy wygląd obiektów danej klasy, tworząc regułę, w której jako selektor podajemy kropkę i nazwę klasy, na przykład:

```
.mojaKlasa { font-size: 11px; font-weight: bold; }
```

Od tej pory wszystkie elementy tej klasy będą używały pogrubionej czcionki o rozmiarze 11 pikseli.

Aby określić klasę elementu, dodajemy do znacznika określającego element atrybut `class="nazwa_klasy"`, na przykład:

```
<p class="mojaKlasa">Ten akapit będzie używał czcionki określonej  
w definicji klasy mojaKlasa.</p>
```

Ta sama klasa może być na stronie używana wielokrotnie do formatowania elementów różnych typów. Aby sformatować jedynie fragment tekstu, zawarty na przykład wewnątrz akapitu, można użyć znacznika `<span> ... </span>`:

```
<p>Ten <span class="mojaKlasa">fragment tekstu</span> będzie używał  
czcionki określonej w definicji klasy mojaKlasa.</p>
```

4.

Elementy dynamiczne na stronie – skrypty w języku JavaScript

Strony WWW nie muszą być statycznymi dokumentami wyświetlającymi tekst. Dzięki zastosowaniu programów komputerowych (skryptów) możliwe jest uzyskanie wielu interesujących efektów, takich jak dynamiczne generowanie treści (np. na podstawie informacji pobranych z bazy danych) czy dodanie interaktywnych elementów reagujących na działania użytkownika.

Wyróżniamy dwa rodzaje skryptów:

- **skrypty działające po stronie serwera** – skrypty takie uruchamiane są na serwerze za każdym razem, gdy użytkownik chce otworzyć stronę; zajmują się generowaniem kodu strony, przetwarzaniem danych przekazanych przez użytkownika w formularzach itp.;
- **skrypty działające po stronie przeglądarki** – umożliwiają dodanie do strony elementów interaktywnych, np. przycisków zmieniających wygląd po najechnięciu na nie kursorem myszy, a także modyfikację treści i wyglądu strony w zależności od działań użytkownika.

Najczęściej używane języki skryptowe, działające po stronie serwera, to: VBScript (technologia ASP), Visual Basic, C# (technologia ASP.NET), PHP, Python, JavaScript (technologia Node.js).

Językiem skryptowym działającym po stronie przeglądarki jest JavaScript. Składnia poleceń tego języka jest, w podstawowym zakresie, zbliżona do składni języka C++.

W przykładzie 10. pokazujemy przykład skryptu działającego po stronie przeglądarki, napisanego w języku JavaScript. Wyświetla on aktualną datę i godzinę przy każdym otwarciu strony.



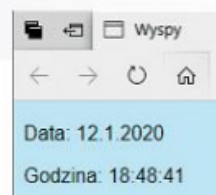
Przykład 10. Tworzenie skryptu w języku JavaScript

Aby wyświetlić aktualną datę i godzinę, korzystamy z następującego skryptu:

- za pomocą słowa kluczowego `var` tworzymy zmienną o nazwie `data` (zmienne w języku JavaScript, podobnie jak w języku Python, nie posiadają z góry określonego typu),
- za pomocą instrukcji przypisania = przypisujemy zmiennej `data` aktualną datę i godzinę,
- używamy instrukcji wyprowadzania danych `document.writeln` do wypisania w treści strony tekstu (ujętego w pojedyncze lub podwójne cudzysłowy) albo wartości: aktualnego dnia, miesiąca, roku (`date.getDate()`, `date.getMonth()`, `date.getFullYear()`) oraz aktualnej godziny, minuty i sekundy (`date.getHours()`, `date.getMinutes()`, `date.getSeconds()`),
- każdą instrukcję kończymy średnikiem.

```
<script type="text/javascript">
var data = new Date();
document.writeln('<p>Data: ');
document.writeln(data.getDate() + '.' + (data.getMonth()+1) + '.' +
data.getFullYear());
document.writeln('</p>');
document.writeln('<p>Godzina: ');
document.writeln(data.getHours() + ':' + data.getMinutes() + ':' +
data.getSeconds());
document.writeln('</p>');
</script>
```

Podany skrypt wyświetli datę i godzinę w następującej postaci:





Ćwiczenie 12. Dodajemy do strony wyświetlanie daty i godziny

1. Dodaj do strony głównej *index.html* o Wyspach Kanaryjskich skrypt wyświetlający aktualną datę i godzinę.
2. Zmodyfikuj skrypt podany w przykładzie 10., aby data i godzina wyświetlane były w jednej linii, rozdzielone przecinkiem.
3. Zapisz plik pod tą samą nazwą.

Wskazówka: Skrypt umieść w sekcji `<body>`.

Język JavaScript posiada wszelkie cechy typowe dla języków programowania, np. możliwość zaprogramowania iteracji.



Przykład 11. Wyświetlenie kwadratów dwudziestu początkowych liczb naturalnych

```
<script type="text/javascript">
  for(var i=1; i<=10; i++)
    document.writeln(i*i);
</script>
```



Ćwiczenie 13. Dodajemy do strony wyświetlanie daty i godziny

1. Sprawdź działanie skryptu z przykładu 11., tworząc prostą stronę w programie Notepad++.
2. Utwórz folder *Matematyka* i zapisz w nim stronę w pliku pod nazwą *index.html*. Wyświetl stronę w przeglądarce.
3. Dodaj skrypt wyświetlający w drugim akapicie wartość sześciu początkowych liczb naturalnych. Zapisz plik pod tą samą nazwą i odśwież stronę.

5. Publikowanie i promowanie stron internetowych w Internecie

5.1. Publikowanie strony WWW



Aby nasza witryna była widoczna dla innych, musimy składające się na nią pliki umieścić na serwerze WWW.

Mamy do wyboru cztery rozwiązania:

1. możemy uruchomić serwer WWW na naszym własnym komputerze (na przykład korzystając z bezpłatnego oprogramowania Apache HTTP, dostępnego między innymi dla systemów Windows i Linux);
2. jeżeli pracujemy w sieci lokalnej podłączonej do Internetu i serwer tej sieci jest odpowiednio skonfigurowany, możemy opublikować witrynę na tym serwerze;
3. możemy umieścić witrynę na powszechnie dostępnym, darmowym serwerze WWW;
4. możemy umieścić witrynę na komercyjnym serwerze WWW.

Każde z tych rozwiązań ma swoje wady i zalety (tabela 8.).

1.	2.	3.	4.
+ Zalety			
+ możliwość samodzielnego skonfigurowania serwera w zależności od potrzeb + objętość strony WWW ograniczona jedynie pojemnością dysku twardego + łatwe zarządzanie plikami strony internetowej (znajdują się na lokalnym dysku komputera) + brak konieczności ponoszenia kosztów utrzymania strony WWW	+ w większości przypadków brak konieczności ponoszenia kosztów utrzymania strony WWW + wygodny dostęp do serwera (połączenie przez sieć lokalną) + witryna dostępna przez 24 godziny na dobę	+ rozwiązanie darmowe + witryna dostępna przez 24 godziny na dobę + stały adres strony	+ witryna dostępna przez 24 godziny na dobę + stały adres strony + możliwość korzystania z pomocy technicznej dostawcy usług + szybkie serwery, zwykle z gwarancją dostępności + możliwość korzystania z zaawansowanych funkcji serwera (np. bazy danych, skrypty uruchamiane po stronie serwera)
- Wady			
- witryna dostępna jest tylko wtedy, gdy nasz komputer jest włączony - konieczność posiadania publicznego, najlepiej stałego, adresu IP - konieczność samodzielnego nadzorowania zainstalowanego oprogramowania serwera (wymagana jest wiedza fachowa) - pracujący serwer obciąża łącze, procesor i pamięć naszego komputera	- objętość witryny ograniczona ilością miejsca udostępnionego przez administratora serwera - łącza w sieciach lokalnych są zwykle w ciągu dnia mocno obciążone przez użytkowników, przez co dostęp do strony WWW może być utrudniony - najczęściej brak gwarancji bezawaryjnego działania serwera	- funkcjonalność darmowych serwerów WWW jest zwykle ograniczona (np. brak obsługi skryptów uruchamianych po stronie serwera) - objętość witryny ograniczona jest zwykle do kilkudziesięciu megabajtów - ograniczony transfer danych, który może generować nasza witryna - brak gwarancji bezawaryjnego działania serwera - wielu dostawców darmowych usług wymaga umieszczania reklam na naszych stronach	- koszty utrzymania strony (zwykle kilkaset złotych rocznie) - ograniczony transfer danych, który może generować nasza witryna (jeżeli witryna generuje duży ruch, zwykle trzeba ponieść dodatkowe koszty)

Tabela 8. Wady i zalety czterech sposobów publikowania strony w Internecie

5.2. Promowanie strony WWW

Jeżeli uważamy, że nasza witryna została profesjonalnie przygotowana i może zainteresować innych, warto rozreklamować ją w Internecie. W tym celu można podjąć następujące działania:

- dodać witrynę do jednego z internetowych katalogów stron WWW;
- wykonać optymalizację witryny pod kątem wyszukiwarek internetowych (opisać poszczególne strony za pomocą słów kluczowych, odpowiednio dobrać tytuły stron i zmodyfikować publikowane treści tak, aby zawierały słowa kluczowe związane z tematyką strony, często wyszukiwane przez internautów);
- jeżeli witryna jest interesująca i nowatorska, można wysłać informacje o niej do mediów;
- promować stronę w mediach społecznościowych;
- wykupić płatne reklamy, na przykład bannery czy odnośniki w wyszukiwarkach.

Jeżeli nasza witryna cieszy się popularnością, można spróbować na niej zarabiać. Najprostsza metoda to uczestnictwo w jednym z programów reklamowych (np. Google AdSense). W ramach tego rodzaju programów zgadzamy się na prezentowanie na naszej witrynie materiałów reklamowych, w zamian za co otrzymujemy niewielką kwotę za każdą odsłonę lub kliknięcie reklamy.



Ćwiczenie 14. Sprawdzamy możliwość dodania strony do katalogu internetowego

Znajdź kilka katalogów internetowych. Sprawdź, jak są w nich zorganizowane kategorie – do której z nich pasowałaby najlepiej strona o Wyspach Kanaryjskich? Dla wybranego katalogu sprawdź również, jakie dane o stronie trzeba podać, jeśli chcemy ją dodać do katalogu, i czy dodawanie stron jest płatne.



Warto zapamiętać

- W Internecie możemy publikować materiały, tworząc własne strony internetowe.
- Do opisu stron stosujemy język znaczników hipertekstowych HTML.
- Aby utworzyć prostą witrynę, wystarczy edytor tekstu, edytor grafiki oraz program umożliwiający przesyłanie na serwer WWW (klient FTP) plików składających się na witrynę.
- Arkusze stylów CSS pozwalają w wygodny sposób modyfikować wygląd dokumentów HTML, skracając jednocześnie długość kodu i eliminując konieczność wielokrotnego powtarzania tych samych znaczników formatujących.
- Dzięki zastosowaniu skryptów jest możliwe m.in. dynamiczne generowanie treści. Wyróżniamy dwa rodzaje skryptów: działające po stronie serwera i po stronie przeglądarki.
- W języku JavaScript możemy tworzyć skrypty działające po stronie przeglądarki.
- W języku JavaScript możemy tworzyć skrypty realizujące iterację. Język ten umożliwia również obsługę zdarzeń.
- Aby wysłać pliki na zewnętrzny serwer WWW, korzystamy najczęściej z protokołu FTP.
- Aby wypromować stworzoną stronę WWW, można, między innymi: dodać informacje o stronie do katalogów internetowych, wymienić się odnośnikami z innymi witrynami o podobnej tematyce czy zoptymalizować witrynę pod kątem wyszukiwarek internetowych.



Pytania i polecenia

1. Za pomocą jakich narzędzi można tworzyć strony WWW?
2. Czym charakteryzuje się język HTML?
3. Jaka jest ogólna struktura pliku w języku HTML?
4. Czym się różni tworzenie stron internetowych za pomocą języka HTML od tworzenia stron z wykorzystaniem innych narzędzi?
5. Do czego służą arkusze stylów CSS?
6. Jakie są sposoby dołączania arkuszy stylów CSS do dokumentów HTML?
7. Jakie korzyści płyną ze stosowania skryptów? Jakie wyróżniamy rodzaje skryptów?
8. Jakie możliwości oferuje język JavaScript?
9. Jakie są możliwości umieszczenia własnej strony w Internecie? Porównaj ich wady i zalety.



Zadania

1. Dodaj do strony o Wyspach Kanaryjskich podstrony opisujące pozostałe sześć wysp i połącz je hiperłączami ze stroną główną. Zapisz wszystkie pliki w folderze *Strona*.
2. Przygotuj stronę internetową o miejscu, które chciałbyś odwiedzić podczas wakacji. Strona powinna przypominać stronę o Wyspach Kanaryjskich utworzoną w tym temacie.
3. Korzystając z wybranego przez siebie narzędzia, zaprojektuj i utwórz stronę internetową związaną z twoimi zainteresowaniami (np. zespoły muzyczne, zwierzęta świata, ulubiona dyscyplina sportowa, modelarstwo, moda). Odszukaj w Internecie podobne strony i postaraj się, aby twoja strona wyróżniała się oryginalnymi i błyskotliwymi treściami (własnymi), ciekawymi ilustracjami (np. częściowo wykonanymi samodzielnie, a nie tylko pobranymi z Internetu). Pamiętaj o zasadach korzystania z cudzych materiałów (temat D1).
4. Odszukaj w Internecie informacje o tym, w jaki sposób za pomocą arkusza stylów CSS można wyłączyć wyświetlanie podkreśleń pod hiperłączami umieszczonymi na stronie.

Dla zainteresowanych

5. Wraz z kolegami i koleżankami stwórzcie stronę internetową o waszej miejscowości (dzielnicy, osiedlu). Strona powinna zawierać informacje o geografii i historii miejsca, ciekawostkach z nim związanych, sławnych ludziach z regionu. Zadbajcie o szatę graficzną strony – możecie np. zamieścić samodzielnie wykonane zdjęcia ciekawych obiektów.
- ~~6. Napisz skrypt, który będzie wyświetlał na stronie aktualną godzinę. Wykorzystaj funkcję `setTimeout()` do cyklicznego wywoływania funkcji aktualizującej wyświetlany czas.~~
7. Zapoznaj się z informacjami na temat formularzy w języku HTML. Utwórz stronę wykorzystującą język JavaScript, która będzie umożliwiać obliczanie pierwiastków równania kwadratowego $ax^2 + bx + c = 0$ dla podanych przez użytkownika współczynników a , b i c .
8. Znajdź w Internecie informacje na temat języka PHP.