

Znaczniki i atrybuty

Znacznik (tag) — element hipertekstowego języka znaczników. To jest najmniejszy element składowy każdej strony internetowej. Każdy tag oznacza jakąś jednostkę: nagłówek, listę, akapit tekstu, obraz.

Aby zaznaczyć znaczniki w tekście dokumentu, stosowane są nawiasy ostre, w których jest określona nazwa znacznika i jego atrybuty.

```
<nazwa_znacznika>...</nazwa_znacznika>
```

Znacznik otwierający wskazuje, gdzie zaczyna się element, a znacznik zamykający gdzie kończy się. Znacznik zamykający jest tworzony przez dodanie ukośnika (/) przed nazwą znacznika. Zawartość znacznika – treść – znajduje się między otwierającym a zamykającym znacznikiem.

Przykłady znaczników.

```
<section>Sekcja</section>  
<p>Akapit</p>  
<a>Link</a>  
<button>Przycisk</button>
```

Komentarze

Komentarze służą do tego, żeby pozostawiać objaśnienia, uwagi w kodzie źródłowym, tymczasowo skomentować fragment kodu itp.

```
<!-- To jest komentarz, jego treść nie będzie wyświetlana na stronie -->  
  
<p>To jest akapit tekstu, on będzie wyświetlany na stronie.</p>  
  
<!--  
  Komentarz może być wielowierszowy.  
  Jest to wygodne w przypadku bardziej obszernych opisów.  
-->
```

Atrybuty

Atrybuty to dodatkowe ustawienia znaczników, za pomocą których można zmieniać właściwości i zachowanie elementu. Każdy znacznik ma obowiązkowe i nieobowiązkowe atrybuty, może ich być kilka lub nie być wcale.

Atrybuty są zapisywane wewnątrz znacznika otwierającego, a ich wartości znajdują się w podwójnych cudzysłowach. Kilka atrybutów oddzielamy spacją.

```
<a href="https://google.com" class="link">...</a>



<input type="text" name="user_name" />

<button type="submit">...</button>

<p class="text">...</p>
```

Przyjrzyjmy się niektórym atrybutom znaczników `<a>`.

```
<a href="http://google.com" target="_blank" title="Wyszukiwarka internetowa Google">
  Google.com
</a>
```

- `href` - adres strony, na którą przejdzie użytkownik po kliknięciu w tekst linku.
- `target` - wskazuje, w której zakładce strona zostanie otwarta po kliknięciu w link.
- `title` - dodaje podpowiedź do tekstu linku.

Przyjrzyjmy się niektórym atrybutom znaczników ``.

```

```

- `src` - adres obrazka.
- `alt` - tekst alternatywny, który będzie wyświetlony, jeśli obraz nie zostanie załadowany.

POMOCNE

Aby poznać dostępne atrybuty znaczników oraz ich przeznaczenie, można przejść do dokumentacji lub przewodników, na przykład htmlreference.io.

Znaczniki sparowane

Składają się z otwierających i zamykających znaczników, które zawierają treść, umożliwiając zmianę jego właściwości lub sposobu wyświetlania, grupowanie według znaczenia. Wewnątrz sparowanych znaczników można zagnieżdżać inne znaczniki, podobnie do matryoski.

```
<!-- Artykuł z tytułem oraz akapitem -->
<article>
  <h1>Krótko o znacznikach sparowanych</h1>
  <p>
    Większość znaczników jest sparowana. Ich zawartość znajduje się między znacznikiem
    otwierającym a zamykającym.
  </p>
```

```
</p>
</article>
```

Znaczniki pojedyncze

Składają się tylko z znacznika otwierającego, nie zawierają treści tekstowej i otrzymują treść lub zachowanie na podstawie atrybutów. Służą do zmiany właściwości dokumentu, połączenia plików itp.

```
<!-- Kodowanie metainformacji -->
<meta charset="utf-8" />

<!-- Pole tekstowe -->
<input type="text" />

<!-- Obrazek -->

```

i INTERESUJĄCO

W poprzednich standardach HTML pojedyncze znaczniki były konieczne zapisywane z odwrotnym ukośnikiem `/` przed nawiasem zamykającym. Na przykład `` albo `<input />`. W nowoczesnym standardzie jest to opcjonalne i na nic nie wpływa.

Zagnieżdżanie znaczników

Zgodnie z pewnymi zasadami znaczniki można zagnieżdżać w sobie. Podczas zagnieżdżania kolejność ich zamykania należy zachować zgodnie z zasadą matryoszki.

```
<znacznik1>
  <znacznik2>
    <znacznik4>...</znacznik4>
    <znacznik5>...</znacznik5>
  </znacznik2>
  <znacznik3>
    <znacznik6>...</znacznik6>
  </znacznik3>
</znacznik1>
```

Oto są ważne znaczniki akapitu z linkiem i zaznaczonym tekstem, a także lista zalet wraz z opisem.

```
<p>
  <a href="squoosh.app">Squoosh</a> - serwis <em>optymalizacja</em> obrazków.
</p>

<p>Zalety</p>
<ul>
  <li>Działa w przeglądarce</li>
  <li>Nie ładuje systemu</li>
</ul>
```

```
<li>Wygodny interfejs</li>
</ul>
```

Ale ta wersja ma problemy (wiersze są podświetlone).

```
<!-- Kolejność zamykania znaczników nie jest przestrzegana -->
<p><a href="squoosh.app">Squoosh</a> - serwis <em></a>optymalizacja</p> obrazków.</em>

<!-- Nie są przestrzegane szczegółowe zasady dotyczące zagnieżdżania znaczników -->
<ul>
  <p>Zalety</p>
  <li>Działa w przeglądarce</li>
  <li>Nie ładuje systemu</li>
  <li>Wygodny interfejs</li>
</ul>
```

Specyfikacja HTML

Specyfikacja [HTML Living Standard](#) - główny dokument opisujący standardy, możliwości i przyszły rozwój języka HTML.

Aby przestudiować layout, przede wszystkim są ważne sekcje opisujące [jakie są rodzaje elementów](#), [co oznacza każdy z nich](#), oraz rozumienie tego, jak można je zagnieżdżać w sobie.

POMOCNE

Element HTML - jest to termin opisujący pewną semantyczną całość w standardzie HTML. Na przykład akapit, nagłówek, listę lub link. W dokumencie HTML element jest reprezentowany przez znacznik (sparowany lub pojedynczy) – składnię języka HTML.

Zagnieżdżanie znaczników

Algorytm określania czy znacznik można zagnieżdżać jest dość prosty.

- Sięgamy po [specyfikację](#) i znajdujemy potrzebny element.
- Sprawdzamy model treści elementu (Content model), do którego zagnieżdżamy.
- Sprawdzamy kategorie elementu (Categories), który zagnieżdżamy.
- Jeśli kategoria jest odpowiednia, a ograniczenia nie zabraniają, można zagnieżdżać, inaczej jest niemożliwe.

Po kilku dniach używania HTML przyzwyczaiacie się do prawidłowego zagnieżdżania elementów w sobie, ponieważ zasady zagnieżdżania będą intuicyjnie zrozumiałe po zapoznaniu się z głównymi znacznikami.

Przewodniki

Oprócz specyfikacji warto mieć pod ręką przewodniki i ściągawki. Zapamiętywać wszystko na pamięć nie jest konieczne, rzeczy bardziej specyficzne lub rzadko używane zawsze można podpatrzeć.

- [Przewodnik znaczników HTML](#)
- [Przewodnik właściwości CSS](#)