

Wstęp do tworzenia stron WWW

Lekcja 14. Sterownie wyświetlaniem elementów strony za pomocą stylu CSS.

1. Kolejność wyświetlania elementów witryny

Cecha *z-index* pozwala określić, jaka ma być kolejność wyświetlania elementów pozycjonowanych, które całkowicie lub częściowo zajmują ten sam obszar strony WWW. Domyślnie kolejność nakładania (zakrywania) zależy od kolejności wpisywania elementów w kodzie źródłowym dokumentu, tzn. te elementy, które zostały wpisane później lub głębiej w strukturze strony mają pierwszeństwo przed poprzednimi. Dzięki właściwości *z-index* można dowolnie kontrolować kolejność wyświetlania elementów witryny.

```
selektor { position: rodzaj; przesunięcie; z-index: numer }
```

Selektorem może znacznik poddany pozycjonowaniu.

Jako "numer" należy wpisać liczbę, określającą, kolejność nakładania pozycjonowanych elementów. Można również wpisać określenie auto, co ustali wartość automatyczną.

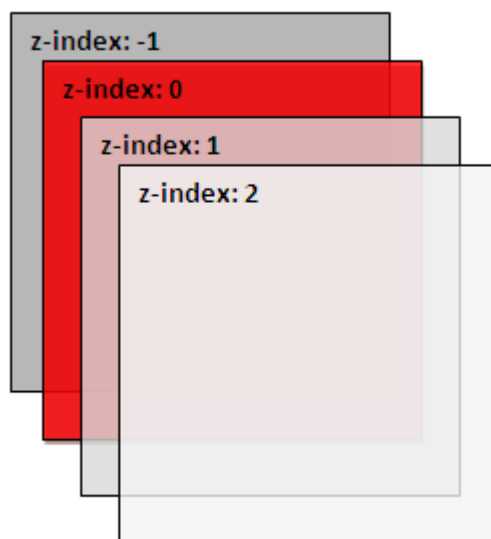
Zasada działania właściwości *z-index* jest taka, że im większą wartość przyjmuje, tym priorytet (pierwszeństwo) wyświetlania jest większe. Parametr auto powoduje, że wartość *z-index* będzie taka sama jak w elemencie nadrzędnym.

UWAGA!

Polecenie *z-index* możemy zastosować, tylko jeśli element witryny jest poddany pozycjonowaniu.

Cecha *z-index* może przyjmować wartości ujemne. Na przykład, dla wartości *z-index: -1*, pozycjonowany element nie zakryje tekstu czy innych znaczników, które znajdowały się wcześniej w danym miejscu, lecz zostanie wyświetlony pod nimi (zwykły tekst i inne elementy domyślnie mają "*z-index: 0*"). Kolejność nakładania schematycznie przedstawia poniższy rysunek:

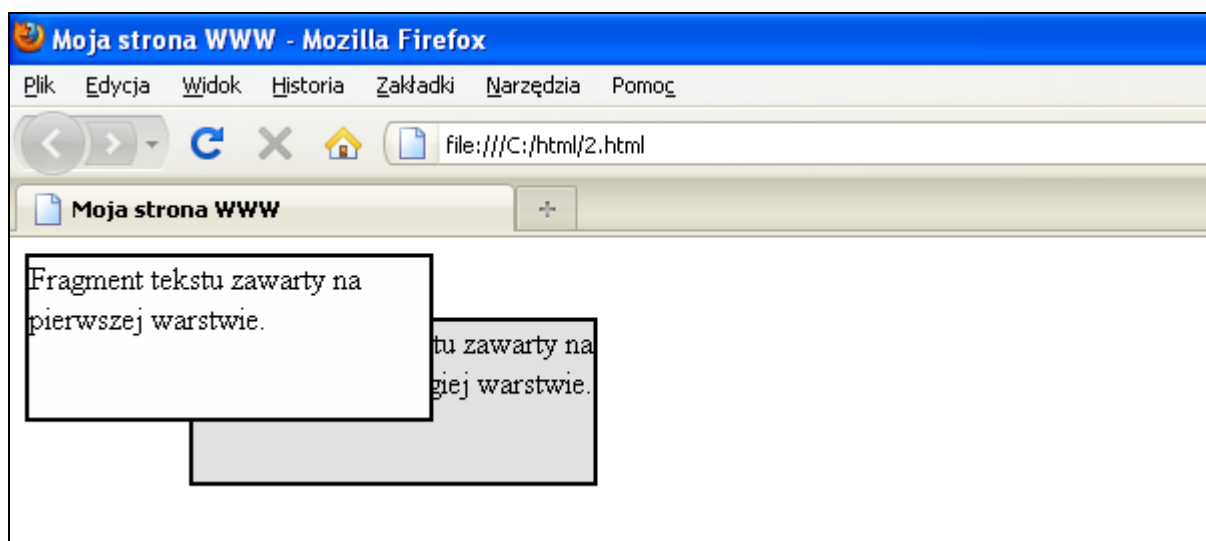
Rysunek 1. Kolejność nakładania elementów w zależności od właściwości *z-index*.



Listing 1. Przykład zastosowania cechy *z-index*.

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Moja strona WWW</title>
<style type="text/css">
  div{
    width:200px;
    height:80px;
    border:2px solid black;
  }
  #warstwa1{
    position:relative;
    background-color:#FEFEFE;
    z-index:2;
  }
  #warstwa2{
    position:absolute;
    background-color:#E1E1E1;
    top:40px;
    left:90px;
    z-index:1;
    text-align:right;
  }
</style>
</head>
<body>
  <div id="warstwa1">Fragment tekstu zawarty na pierwszej
warstwie.</div>
  <div id="warstwa2">Fragment tekstu zawarty na drugiej
warstwie.</div>
</body>
```

Rysunek 2. Przykład zastosowania cechy *z-index*.



2. Kadrowanie elementów

Polecenie *clip* pozwala na określenie, która część elementu ma być widoczna, podczas gdy pozostały obszar zostanie ukryty. Ta właściwość jest zazwyczaj wykorzystywana w przypadku obrazów, ale można w ten sposób przyciąć dowolny element np. akapit.

```
selektor { position: absolute; clip: rect(pgY, pgX, ldY, ldX) }
```

Selektorem może być praktycznie każdy element pozycjonowany metodą bezwzględną.

Właściwość *clip* przyjmuje wartość *auto* (wyświetlanie standardowe) lub wartości w postaci:

```
rect(pgY, pgX, ldY, ldX)
```

Powyższe polecenie oznacza prostokąt, natomiast *pgY* i *pgX* określają współrzędne prawego górnego rogu, a *ldY* i *ldX* współrzędne lewego dolnego rogu. Obecnie dostępne jest jedynie kadrowanie za pomocą prostokąta.

Ukrywanie elementów

Cecha *visibility* steruje wyświetlaniem elementu na ekranie. Pozwala ukryć dowolny element, a w jego miejsce wprowadzić pustą przestrzeń o takich samych rozmiarach, jakie miałby element, gdyby tylko był widoczny.

```
selektor { visibility: typ }
```

Selektorem może być praktycznie dowolny znacznik.

Wartości cechy *visibility* są następujące:

- visible - element widoczny (wartość domyślna),
- hidden - element ukryty (w miejscu elementu pojawi się pusta przestrzeń o takich samych rozmiarach jak element),
- collapse – ukrycie wiersza lub kolumny tabeli, dla innych elementów działa tak samo jak wartość hidden..

Wyświetlanie

Cecha *display* pozwala określić sposób wyświetlania elementów witryny. Właściwość pozwala zmienić domyślny sposób wyświetlania.

```
selektor { display: sposób }
```

Selektorem może być praktycznie dowolny znacznik.

W miejsce „sposób” można wpisać następujące wartości:

- block - element będzie wyświetlony w bloku (odstęp z góry i z dołu),
- inline - element będzie wyświetlony w linii (sąsiadując z innymi),
- list-item - element wykazu: `...`,
- none - element nie będzie wyświetlany,



- inline-block - element zachowuje się podobnie jak element zastępowany (Firefox 3, Opera 7, Chrome, Konqueror),
- run-in - jeśli po elemencie następuje brat będący blokiem, element zostanie sformatowany jako "inline" (w linii) w ramach treści brata. W przeciwnym razie element zostanie sformatowany jako "block" (Opera 6, Chrome),
- table - tabela blokowa: <table>...</table> ,
- inline-table - tabela "inline": <table>...</table>
- table-row-group - grupa wierszy tabeli: <tbody>...</tbody> ,
- table-header-group - nagłówek tabeli: <thead>...</thead> ,
- table-footer-group - stopka tabeli: <tfoot>...</tfoot> ,
- table-row - wiersz tabeli: <tr>...</tr> ,
- table-column-group - grupa kolumn tabeli: <colgroup>...</colgroup> ,
- table-column - kolumna tabeli: <col /> ,
- table-cell - komórka tabeli: <td>...</td> , <th>...</th> ,
- table-caption - tytuł (podpis) tabeli: <caption>...</caption> .

Polecenie to pozwala określić sposób, w jaki będzie wyświetlony na ekranie dany element. Jest to bardzo przydatna funkcja, jeśli chcemy zrezygnować z domyślnego sposobu wyświetlania. Na przykład: wszystkie tytuły (<Hn>...</Hn>) są wyświetlane domyślnie w bloku, tzn. automatycznie są dodawane linijki przerwy pomiędzy danym tytułem a sąsiadującymi z nim elementami. Lecz dzięki poleceniu *"display: inline"* możliwe jest ustawienie dwóch sąsiednich tytułów w jednej linii.

Kształty kursora

Cecha cursor pozwala określić jaki będzie kształt kursora myszy, gdy znajduje się on nad danym elementem witryny.

```
selektor { cursor: rodzaj }
```

Selektorem może być praktycznie dowolny znacznik.

Możliwe do zastosowania wartości zostały przedstawione poniżej:

- auto - automatyczny wybór kształtu kursora (zależy od typu elementu, na który wskazujemy - tryb domyślny),
- default - kursor domyślny (strzałka),
- none - brak kursora,
- help - kursor pomocy,
- pointer - wskaźnik (wskazuje odsyłacz),
- hand – dłoń,
- progress - "postęp",
- wait - "poczekaj",
- crosshair – krzyżyk,
- text - kursor tekstowy,
- vertical-text - pionowy kursor tekstowy,
- alias – skrót,
- copy – kopia,
- move - "przesuń",
- no-drop - wskazuje, że przeciągany element nie może zostać upuszczony w tej pozycji,
- not-allowed - żądana akcja nie zostanie przeprowadzona,
- ne-resize - "północny wschód",
- e-resize - "wschód",
- se-resize - "południowy wschód",
- s-resize - "południe",
- col-resize - pozioma zmiana szerokości kolumny,
- row-resize - pionowa zmiana wysokości wiersza,
- all-scroll - przewijanie w dowolnym kierunku.

Istnieje możliwość wczytania dowolnego kursora z pliku z obrazem kursora. W tym celu należy skorzystać z poniższej konstrukcji:

```
selektor { cursor: url(ścieżka dostępu), rodzaj }
```

W miejsce ścieżka dostępu należy wpisać adres do pliku z obrazem kursora, a rodzaj zastąpić wartością określającą kształt kursora.