

Przydatne strony

<http://www.kurshtml.edu.pl/css/>

<http://www.e-zoner.pl/css>

<http://cpw.net.pl/css>

<http://www.gajdaw.pl/html/uklad-witryny-gwz/print.html>

<http://kodcss.pl/>

<http://www.barelyfitz.com/screencast/html-training/css/positioning/>

Identyfikatory i klasy

Dokument HTML składa się z wielu różnych elementów. Część z nich, jak te podstawowe, a więc między innymi p, h1 czy ul, często będą formatowane za pomocą CSS. Jednak w przypadku korzystania z selektora, którym jest nazwa elementu, automatycznie style przypisywane są do wszystkich elementów danego typu występujących w dokumencie. Jest to problematyczne w przypadku, gdy chcemy wyróżnić tylko jeden lub kilka elementów danego typu spośród wszystkich występujących w dokumencie. W takim przypadku przychodzą z pomocą identyfikatory i klasy.

Identyfikatory - id

Identyfikatorów używanych w elementach dotyczą dwie zasady:

- każdy element może mieć tylko jeden identyfikator,
- każda strona może mieć tylko jeden element z takim identyfikatorem.

Z powyższych zasad wynika zatem, że każdy identyfikator jest unikalny i może wystąpić tylko raz na danej stronie. W kodzie strony do elementu HTML dodaje się atrybut id.

Identyfikatory - id

W kodzie strony do elementu HTML dodaje się atrybut id.

```
HTML <h1 id="identyfikator"></h1>
```

Tworząc selektor w oparciu o atrybut id, należy wartość atrybutu poprzedzić znakiem #.

```
CSS #identyfikator {color: red;}
```

Identyfikator może być poprzedzony nazwą elementu na który wskazuje.

```
CSS h1#identyfikator {color: red;}
```

```
<body>
<h1>temat 1</h1>
<p>In publishing and graphic design, lorem ipsum
is a filler text commonly used to demonstrate the
graphic elements of a document or visual
presentation. </p>
<h1>temat 2</h1>
<p id="wazne">In publishing and graphic design,
lorem ipsum is a filler text commonly used to
demonstrate the graphic elements of a document or
visual presentation. </p>
<h1>temat 3</h1>
<p>In publishing and graphic design, lorem ipsum
is a filler text commonly used to demonstrate the
graphic elements of a document or visual
presentation. </p>
<h1>temat 4</h1>
<p>In publishing and graphic design, lorem ipsum
is a filler text commonly used to demonstrate the
graphic elements of a document or visual
presentation. </p>
<h1>temat 5</h1>
<p>In publishing and graphic design, lorem ipsum
is a filler text commonly used to demonstrate the
graphic elements of a document or visual
presentation. </p>
</body>
```

```
p{
  background-color:#00ffff;
}
#wazne{
  background-color:#1E90FF;
  color: red;
  font-size:30px;
  font-style: italic;
  font-weight:bold;
}
```

Przykład

temat 1

In publishing and graphic design, lorem ipsum is a filler text commonly used to demonstrate the graphic elements of a document or visual presentation.

temat 2

In publishing and graphic design, lorem ipsum is a filler text commonly used to demonstrate the graphic elements of a document or visual presentation.

temat 3

In publishing and graphic design, lorem ipsum is a filler text commonly used to demonstrate the graphic elements of a document or visual presentation.

temat 4

In publishing and graphic design, lorem ipsum is a filler text commonly used to demonstrate the graphic elements of a document or visual presentation.

temat 5

In publishing and graphic design, lorem ipsum is a filler text commonly used to demonstrate the graphic elements of a document or visual presentation.

HTML Block Elements

Most HTML elements are defined as **block level** elements or as **inline** elements.

Block level elements normally start (and end) with a new line when displayed in a browser.

Examples: `<h1>`, `<p>`, ``, `<table>`

HTML Inline Elements

Inline elements are normally displayed without starting a new line.

Examples: ``, `<td>`, `<a>`, ``

The HTML `<div>` Element

The HTML `<div>` element is a block level element that can be used as a container for grouping other HTML elements.

The `<div>` element has no special meaning. Except that, because it is a block level element, the browser will display a line break before and after it.

When used together with CSS, the `<div>` element can be used to set style attributes to large blocks of content.

Another common use of the `<div>` element, is for document layout. It replaces the "old way" of defining layout using tables. Using `<table>` elements for layout is not the correct use of `<table>`. The purpose of the `<table>` element is to display tabular data.

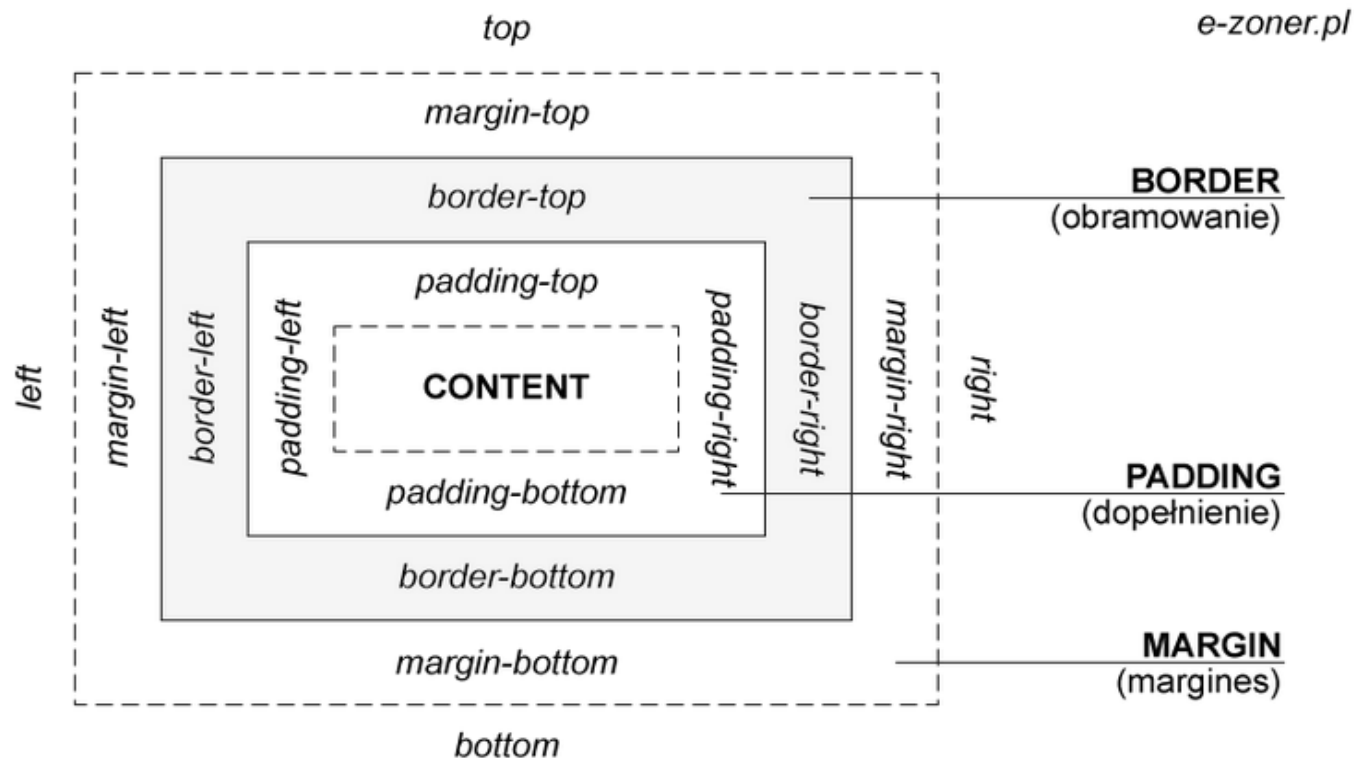
The HTML `` Element

The HTML `` element is an inline element that can be used as a container for text.

The `` element has no special meaning.

When used together with CSS, the `` element can be used to set style attributes to parts of the text.

Każdy element blokowy posiada obszar zawartości, mogący być na przykład tekstem, obrazkiem czy choćby innymi elementami. Element taki opcjonalnie może być otoczony dopełnieniem (**padding**), obramowaniem (**border**) oraz marginesem (**margin**).



W modelu blokowym wartości szerokości i wysokości elementu, odnoszą się wyłącznie do zawartości. Oznacza to, że aby obliczyć całkowitą szerokość lub wysokość bloku, należy zsumować w poziomie lub pionie wszystkie wartości: dopełnienia, obramowania, marginesu oraz zawartości.

```

<body>
<div id="jeden">
<h1>temat 1</h1>
<p>In publishing and graphic design, lorem ipsum is a
filler text commonly used to demonstrate the graphic
elements of a document or visual presentation. </p>
<h1>temat 2</h1>
<p id="wazne">In publishing and graphic design, lorem
ipsum is a filler text commonly used to demonstrate the
graphic elements of a document or visual presentation. </p>
</div>
<div>
<h1>temat 3</h1>
<p>In publishing and graphic design, lorem ipsum is a
filler text commonly used to demonstrate the graphic
elements of a document or visual presentation. </p>
<h1>temat 4</h1>
<p>In publishing and graphic design, lorem ipsum is a
filler text commonly used to demonstrate the graphic
elements of a document or visual presentation. </p>
<h1>temat 5</h1>
<p>In publishing and graphic design, lorem ipsum is a
filler text commonly used to demonstrate the graphic
elements of a doc . . . </p>
</div>
</body>

```

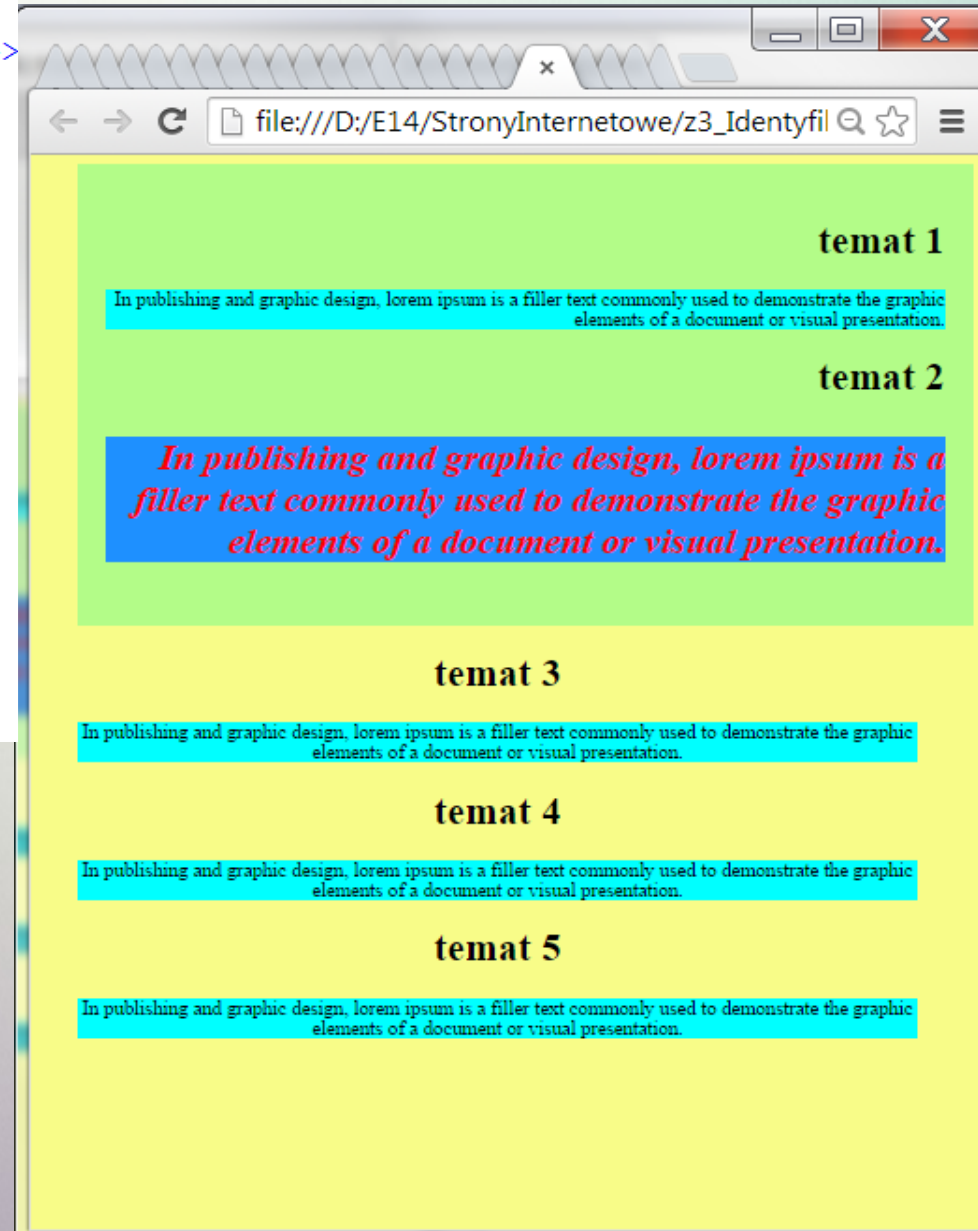
```

body{
background-color:#F8FC88;
}
div{
text-align:center;

width:90%;
margin-left:30px;
}
div#jeden{
background-color:#B3FC88;
text-align:right;
padding:3%;
}

```

Przykład - div



Klasy - class

Klas używanych w elementach dotyczą dwie zasady:

- ta sama klasa może być użyta w wielu elementach,
- w elemencie może być użyte wiele klas.

Z powyższych zasad wynika zatem, że jedna klasa może być użyta do opisania więcej, niż jednego elementu jak również, że więcej klas może opisywać jeden element. W kodzie strony do elementu HTML dodaje się atrybut class.

Klasy - przykład

```
<body>
<div id="jeden">
<h1>temat 1</h1>
<p class="zielony">In publishing and graphic design, lorem ipsum is a
filler text commonly used to demonstrate the graphic elements of a document
or visual presentation. </p>
<h1>temat 2</h1>
<p id="wazne">In publishing and graphic design, lorem ipsum is a filler text
commonly used to demonstrate the graphic elements of a document or visual
presentation. </p>
</div>
<div>
<h1 class="zielony">temat 3</h1>
<p>In publishing and graphic design, lorem ipsum is a filler text commonly
used to demonstrate the graphic elements of a document or visual
presentation. </p>
<h1>temat 4</h1>
<p>In publishing and graphic <span>design, lorem ipsum is
commonly used to demonstrate the </span> graphic elements
visual presentation. </p>
<h1>temat 5</h1>
<p class="zielony">In publishing and graphic design, lor
filler text commonly used to demonstrate the graphic elem
or visual presentation. </p>
</div>
</body>
```

```
.zielony{
  color: green;
  font-size: 40px;
}
span, h1.zielony{
  text-transform: uppercase;
}
```

temat 1

In publishing and graphic design, lorem ipsum is a filler text commonly used to demonstrate the graphic elements of a document or visual presentation.

temat 2

In publishing and graphic design, lorem ipsum is a filler text commonly used to demonstrate the graphic elements of a document or visual presentation.

TEMAT 3

In publishing and graphic design, lorem ipsum is a filler text commonly used to demonstrate the graphic elements of a document or visual presentation.

temat 4

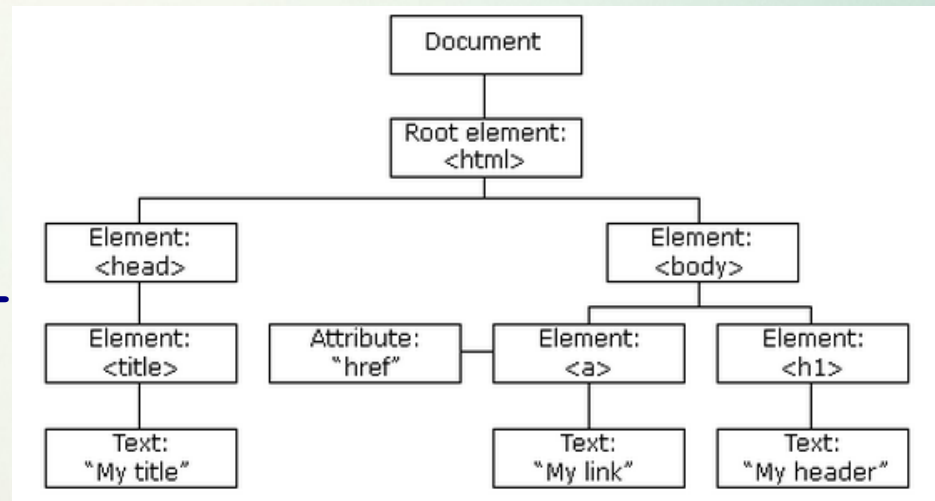
In publishing and graphic DESIGN, LOREM IPSUM IS A FILLER TEXT COMMONLY USED TO DEMONSTRATE THE graphic elements of a document or visual presentation.

temat 5

In publishing and graphic design, lorem ipsum is a filler text commonly used to demonstrate the graphic elements of a document or visual presentation.

Model DOM(Document Object Model) - Hierarchia dokumentu

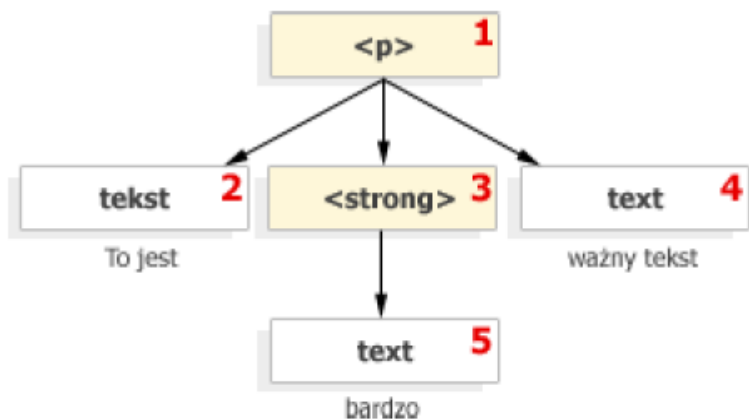
Nasz dokument możemy rozrysować jako hierarchiczne drzewo. Na samej górze jest document HTML, a tuż pod nim znajdują się jego "dzieci" (inna nazwa to korzenie - nody) - czyli elementy znajdujące się w document.



Każdy element na stronie tworzy tak zwany node czyli pojedynczy korzeń drzewa. Takimi nodami są nie tylko Tagi, ale także tekst w nich zawarty, ich atrybuty itp.

Rozpiszmy przykładowy akapit, który posiada w sobie pogrubiony tekst:

```
<p id="paragraf">To jest <strong>bardzo</strong> ważny tekst</p>
```



- Node 1 jest rodzicem (parent) nodów 2,3 i 4
- Nody 2, 3 i 4 są dziećmi (children) **noda 1**
- Node 2 jest pierwszym dzieckiem (first child) **noda 1**
- Node 4 jest ostatnim dzieckiem (last child) **noda 1**
- Node 3 jest kolejnym elementem po Nodzie 2
- Node 3 jest poprzednim elementem przez Nodem 4
- Node 5 nie jest dzieckiem (child) **noda 1** - jest za to dzieckiem Noda 3
- Nody 2, 4 i 5 są typu tekstowego

Pozycjonowanie elementów

Pozycję elementu można określić za pomocą właściwości **position**. Możliwe jest ustalenie pozycji w relacji do poprzedzającego elementu, pozycji absolutnej względem elementu przodka lub pozycji ustalonej względem okna przeglądarki.

`position: static | relative | absolute | fixed | initial | inherit ;`

Wartości właściwości position:

Wartość	Opis
static	Elementy ładowane są w takim porządku, jak wynika to z układu w dokumencie. Właściwości <code>top</code> , <code>right</code> , <code>bottom</code> , <code>left</code> nie mają tutaj zastosowania. Wartość domyślna.
relative	Element jest umieszczany względem swojego normalnego położenia. Na przykład określając właściwość <code>top</code> na 10 pikseli, element zostanie umieszczony w pozycji przesuniętej o 10 pikseli w dół względem normalnego położenia.
absolute	Element jest umieszczany w pozycji absolutnej, określanej względem pierwszego elementu przodka, posiadającego inną metodę pozycjonowania, niż <code>static</code> .
fixed	Element jest umieszczany względem okna przeglądarki.
initial	Ustawia wartość właściwości na domyślną.
inherit	Wartość właściwości jest dziedziczona z elementu rodzica.

Pozycjonowanie względne (relative)

Element pozycjonowany względnie jest umieszczany w swojej pozycji w układzie normalnym, a następnie przesuwany względem tej pozycji.

Za pomocą właściwości `top`, `right`, `bottom`, `left` określa się o jaką wartość zostanie przesunięty element względem normalnej pozycji. Przesunięcie następuje względem krawędzi pozycjonowanego pudełka. Tak więc zastosowanie na przykład właściwości `left` z dodatnią wartością bezwzględną, doda do lewej krawędzi pudełka margines o podanej wartości. Zastosowanie właściwości `left` z ujemną wartością bezwzględną, doda do lewej krawędzi margines ujemny, czyli przesunie całe pudełko w lewo o podaną wartość.

Pozycjonowanie bezwzględne (absolute)

Element pozycjonowany bezwzględnie jest umieszczany w pozycji określonej przez właściwości przesuwające (top, right, bottom, left), względem pierwszego elementu nadrzędnego, który ma przypisaną właściwość position z inną wartością niż static.

Brak określenia właściwości position (z wartością inną niż static), w elementach nadrzędnych elementu pozycjonowanego, spowoduje że element ten będzie pozycjonowany względem elementu body. Określenie w przynajmniej jednym elemencie nadrzędnym właściwości position, spowoduje że pozycja elementu zostanie obliczona względem tego elementu.

Pozycjonowanie ustalone (fixed)

Element posiadający ustaloną pozycję jest pozycjonowany względem okna przeglądarki i zachowuje swoją pozycję w trakcie przewijania strony.

position: fixed

Kolejność nakładania warstw - z-index

Korzystanie z właściwości `position` i przesunięć elementów względem układu normalnego, pociąga za sobą konieczność określenia kolejności ich nakładania. Związane jest to z faktem, że przemieszczenie elementu w inne miejsce, niż wynika to z normalnego układu, powoduje iż elementy następujące, mogą go częściowo lub w pełni zasłonić.

Kolejność występowania elementów w dokumencie, w układzie normalnym, jest wytyczną dla przeglądarki jak elementy te mają być ustawione. Inaczej jest w przypadku elementów pozycjonowanych z przesunięciami, gdzie kolejność nakładania warstw można wymusić przy użyciu właściwości `z-index`. Dzięki tej właściwości, możliwe jest nakazanie przeglądarce, aby element występujący wcześniej w dokumencie, nakładał się na element występujący w tym dokumencie później.

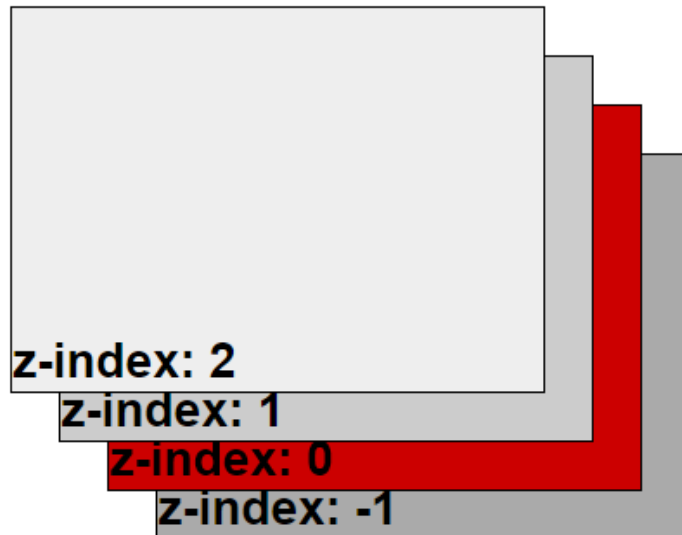
Kolejność nakładania warstw - z-index

z-index: auto | liczba | initial | inherit ;

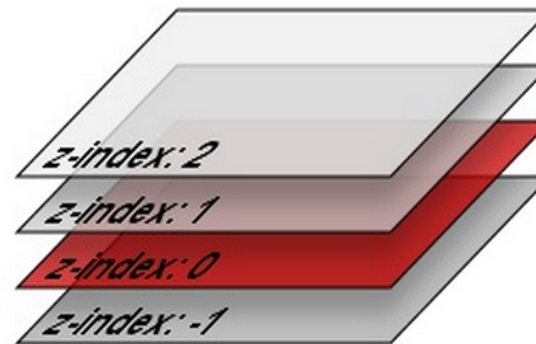
Wartość	Opis
auto	Ustawia kolejność w stosie równą pozycji rodzica. Wartość domyślna.
<i>liczba</i>	Ustawia liczbę wykorzystywaną do obliczenia, jak elementy mają być ułożone w stosie. Wartościami są liczby całkowite. Liczby mogą być ujemne.
initial	Ustawia wartość właściwości na domyślną.
inherit	Wartość właściwości jest dziedziczona z elementu rodzica.

Kolejność nakładania warstw - z-index

Interpretowane są również wartości ujemne. Dzięki podaniu np. "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"). Wartości bardziej mniejsze od zera (-2, -3, -4,...) mają zastosowanie, jeśli chcemy wyświetlić pod tekstem więcej elementów w określonej kolejności nakładania:



co można przedstawić w trzech wymiarach następująco:



Kolejność nakładania warstw - z-index

Skąd w ogóle wzięła się nazwa "z-index"? *Index (indeks)*, to po prostu numer porządkowy określający kolejność. Natomiast przedrostek "z" odnosi się do trzeciego wymiaru układu współrzędnych, który swój początek ma w lewym-górnym rogu ekranu:

1. oś "x" - biegnie poziomo w prawo
2. oś "y" - biegnie pionowo w dół
3. oś "z" - jest skierowana w stronę użytkownika patrzącego na ekran



Kolejność nakładania warstw - z-index



HTML

```
<br />

```

CSS

```
#rys1{
width: 100px;
height: 75px;
position: relative;
}
#rys2{
position: relative;
left: 84px;
bottom: 91px;
}
```



HTML

```
<br />

```

CSS

```
#rys1{
width: 100px;
height: 75px;
position: relative;
z-index: 2;
}
#rys2{
position: relative;
left: 84px;
bottom: 91px;
z-index: 1;
}
```