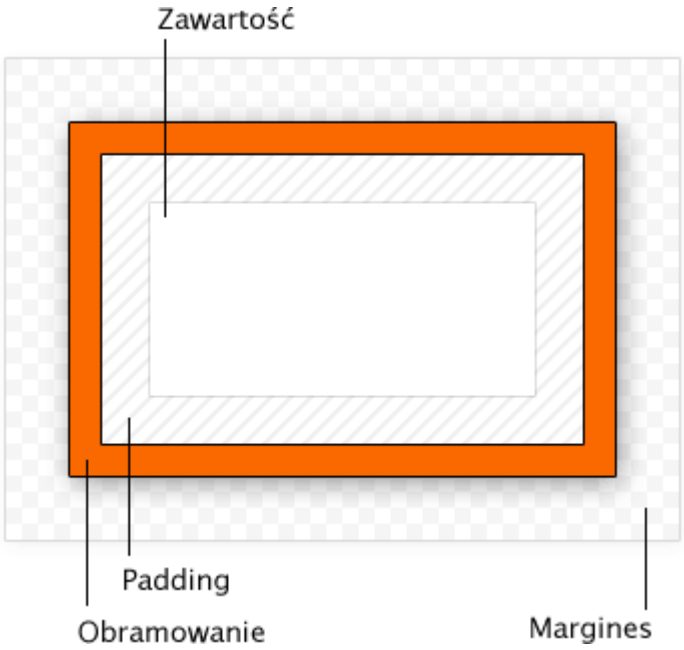
Box Model

Wielkość

Standardowo szerokość i wysokość ustalane w CSS odnoszą się do wielkości **zawartości** danego elementu. Całkowita wielkość elementu jest większa o **padding**, **obramowanie** (border) i niewidoczny **margines** (margin).



Typy boksów

CSS generalnie dzieli elementy na **blokowe** (block) – div i **liniowe** (inline) - span. Elementy blokowe są zawsze prostokątne i mogą mieć nadane wymiary (np. akapit, nagłówek). Elementy liniowe są częścią tekstu i mogą być przełamane przy przenoszeniu do nowej linii (np. pogrubienie, odnośnik).

Szczególnym przypadkiem w CSS są tabele. Ze względu na kompatybilność z HTML tabele nieco inaczej reagują na nadane im wymiary, obramowanie i mają pewne ograniczenia co do pozycjonowania.

Style boksów — Wymiary

Wysokość i szerokość w CSS ma odmienne zachowanie. Strony mają tendencję do wypełniania całej szerokości ekranu, ale nie wysokości, dlatego bardziej skomplikowane jest umieszczenie kilku elementów obok siebie w poziomie, niż w pionie oraz trudniejsze jest rozciąganie elementów na całą wysokość, niż szerokość.

Szerokość: `width`

Ustala szerokość zawartości elementów. Wyjątek stanowią elementy **inline**, którym nie można nadać wymiarów. Jeśli zawartość elementu nie zmieści się w podanych wymiarach nastąpi przepełnienie.

Wartości:

absolutne wartości

: bez niespodzianek — np. `10px` oznacza, że element będzie miał 10 pikseli szerokości.

auto

: ma różne znaczenie w zależności od typu obiektu.
Dla elementów pływających i absolutnie pozycjonowanych oznacza wielkość dopasowaną do zawartości.
Dla zwykłych (statycznych) elementów **blokowych** oznacza całą dostępną szerokość pomniejszoną o margines, obramowanie i padding (przy szerokości 100% padding i ramka by wystawały poza rodzica).

procenty

: procentowa szerokość elementu nadrzędnego (lub elementu zawierającego dla elementów absolutnie pozycjonowanych).

Elementom, które zawierają tekst najlepiej nadawać wielkość w jednostkach `em` — zależnej od wielkości tekstu.

Wysokość: `height`

Ustala wysokość zawartości elementów. Wyjątek stanowią elementy **inline**, którym nie można nadać wymiarów.

Jeśli zawartość elementu nie zmieści się w podanych wymiarach nastąpi przepełnienie.

absolutne wartości

bez niespodzianek — np. `20em` to 20 linii.

auto

minimalna wysokość zawartości (to jest domyślna wartość)

procenty

procentowa wysokość elementu nadrzędnego (lub elementu zawierającego dla elementów absolutnie pozycjonowanych). Działa tylko w określonych przypadkach — zobacz poniżej.

Wysokość w procentach

Wysokość podana w procentach działa tylko w prostych przypadkach:

```
<div style="height:250px">
 <div style="height:100%" />
</div>
```

Powyższy przykład zadziała, bo wiadomo, że wysokość wewnętrznego `<div>` to `100%` z `250px`.

```
<div>
 <div style="height:100%" />
</div>
```

Ten przykład nie zadziała, bo wiadomo tylko, że chodzi o `100%`, ale nie wiadomo z jakiej konkretnej wysokości. Paradoks bardziej widoczny jest w takim przypadku:

```
<div>
 <div style="height:100%" />
 <div style="height:100%" />
</div>
```

Zewnętrzny `<div>` ma mieć wysokość jak jego zawartość, ale jego zawartość jest dwa razy większa od niego. W rezultacie wychdzi na to, że `<div>` ma być dwa razy większy od siebie samego...

Minimalna i maksymalna wielkość

Można ograniczyć zakres działania `width` i `height` za pomocą `min-width`, `max-width`, `min-height` i `max-height`.

```
width: 50%;
max-width: 600px;
min-width: 20ex;
```
Element, który ma 50% szerokości, ale nie więcej niż 600 pixeli i nie mniej, niż ok. 20 znaków.

`min-width`/`min-height` ma pierwszeństwo nad `max-width`/`max-height`

Nadmiar: `overflow`

Jeśli element ma nadane konkretne wymiary i jego zawartość się w nim nie mieści, to następuje przepełnienie. Może ono być widoczne na różne sposoby:

`visible`

Zawartość będzie wystawać poza element. To jest domyślne zachowanie.

`hidden`

Zawartość zostanie przycięta do wymiarów elementu. Nadmiar będzie niewidoczny.

`scroll` i `auto`

Element będzie miał paski przewijania, które pozwolą zobaczyć nadmiarową treść. W przypadku wartości `auto` paski przewijania pojawią się tylko wtedy, gdy to konieczne.

Jeśli chcesz, żeby element się rozciągał do wielkości zawartości, to nadaj mu `min-width`/`min-height` zamiast `width`/`height`.

Style boksów — odstępy i obramowanie

Poniższe właściwości są dostępne dla czterech boków każdego boksu. Boki wybiera się dodając `-top` (góra), `-right` (prawo), `-bottom` (dół) lub `-left` (lewo) do nazwy właściwości. Można też użyć skróconego zapisu podając jedną (wszystkie boki), dwie (góra/dół, boki), trzy (góra, boki, dół) lub cztery (od góry wg wskazówek zegara) właściwości na raz.

```
margin: 0 0 15px
```
15 pixeli na dole, góra i boki wyzerowane

`padding: 5px 3em`
5 pixeli na górze i na dole, a 3 wysokości linii tekstu po bokach.

`border-top: 0`
Bez górnego obramowania

`padding`

`padding` czasem jest tłumaczony jako *dopełnienie* lub *margines wewnętrzny*

Ustala odstęp między treścią, a obramowaniem elementu. Nie odsuwa tła elementu, dzięki czemu jest przydatna do zapobiegania zasłaniania brzegów tła (obrazka) elementu przez jego tekst.

Górny i dolny `padding` na elementach **inline** jest widoczny, ale nie wpływa na wysokość linii tekstu. Jeśli wysokość ma się zmienić, to użyj `line-height` lub zmień `display` na np. `inline-block`.

Jeśli elementowi z `width` lub `height` ustawiasz `padding`, a chcesz, żeby wizualnie zachował te same wymiary, to musisz pomniejszyć wartość `width` i `height` o `padding`.

Marginesy: `margin`

wcięcia akapitowe robi `text-indent`

Marginesów używa się do kontrolowania odstępów między elementami oraz do robienia wcięć na *całą wysokość elementu*.

Odstępy między słowami i literami robi się przez `word-spacing` i `letter-spacing`

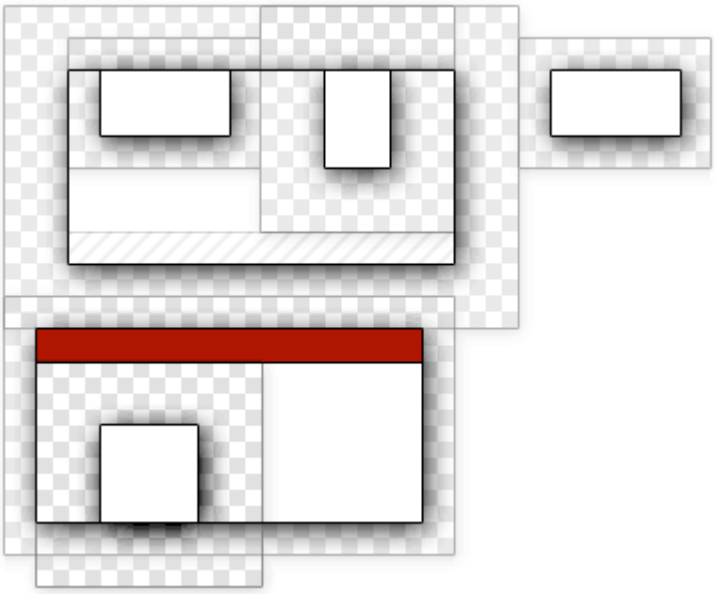
Można zmieniać pionowe i poziome marginesy elementów blokowych. W przypadku elementów **inline** widoczne będą tylko poziome marginesy.

Zapadanie się marginesów

Górne i dolne marginesy elementów blokowych *zapadają* (nakładają) się. Oznacza to, że ustalony margines jest minimalną odległością między danym elementem, a sąsiednimi (odstęp między górnymi/dolnymi krawędziami dwóch elementów będzie równy większemu z ich marginesów, a nie ich sumie).

Marginesy zapadają się nie tylko dla sąsiednich elementów, ale nawet jeśli jeden element jest w drugim *i między marginesami nie ma obramowania, ani paddingu*.

Lewe i prawe marginesy po prostu rezerwują określoną odległość z boku elementu — zapadanie ich nie dotyczy. Odległość między bokami sąsiednich elementów będzie równa sumie ich marginesów.



Na ilustracji marginesy są zaznaczone kratkowanym polem, padding kreskowany, a obramowanie kolorem. Zauważ, że zapadnięte pionowe marginesy nakładają się.

Teoria stojąca za marginesami wygląda bardzo skomplikowanie, ale w praktyce okazuje się intuicyjna i marginesy nie sprawiają większych problemów. Gdyby sprawiały, to masz do czynienia z ich zapadaniem — ustawienie `border` lub `padding` na nadrzędnym elemencie rozwiąże „problem”.

Centrowanie: `margin:auto`

Marginesy poziome przyjmują wartość `auto`, która powoduje wycentrowanie całego obiektu, jeśli ma on nadaną szerokość.

Więcej możliwości wyrównania do bocznych krawędzi daje właściwość `float`.

Jeśli ustawione są oba, element jest centrowany. Jeśli tylko lewy jest `auto`, element będzie wyrównany do prawej.

```
margin: 1em auto;
width: 50%
```

Górne i dolne marginesy na wysokość jednego wiersza, a boczne marginesy automatyczne, centrujące element.

Obramowanie: `border`

Dzieli się na właściwości: `border-color` (kolor), `border-style` (styl) i `border-width` (grubość) z których każda z osobna przyjmuje skrócony (od 1 do 4) zapis parametrów w analogiczny sposób do margin/padding lub `border-top`, `border-right`, `border-bottom` i `border-left`, i samo `border`, które przyjmują na raz grubość, styl i kolor, w dowolnej kolejności.

Styl obramowania: `border-style`

solid

jednolite obramowanie

dotted

złożone z kropek

dashed

z kresek

double

podwójna linia

inset, outset

wgłębione i wypukłe

ridge, groove

podwójne wgłębione, podwójne wypukłe

Nie można precyzyjnie sterować odcieniami kolorów użytymi do `inset`, `outset`, `ridge` i `groove`. Dwa pierwsze style da się zastąpić przez `solid` i odpowiednio dobrane kolory.

```
border: 1px solid red
```

Jednopixelowa czerwona ramka wokół całego elementu

```
border-left: gray 2px dotted
```

Szara dwupixelowa kropkowana lewa krawędź

```
border: dashed blue; border-width: 5px 0
```

Kreskowane niebieskie krawędzie na górze i na dole elementu (grubość 0 usuwa obramowanie). Można w ten sposób składać właściwości CSS dzięki kaskadzie.

```
border: 3px solid; border-color: white black black white;
```

Element mający wygląd wypukłego przycisku

Są też dodatkowe cechy i wartości dla obramowania komórek tabel.

Obwódka: `outline`

Ma składnię analogiczną do `border`.

Dodaje obwódkę wokół całego elementu, ale *nie wypływa na jego wielkość*. Jest zawsze poza nim i ew. nachodzi na elementy sąsiednie.

Jest to przydatne do wyróżnienia elementu — np. do zaznaczenia aktywnego elementy formularza czy przy tworzeniu arkuszy stylów, gdy chce się zobaczyć wielkość elementu bez burzenia precyzyjnych układów.

```
input:focus {outline: 1px solid red;}
```

Może być tylko jedna obwódka dla całego elementu. *Nie ma* właściwości jak `outline-left`, jest tylko jeden ogólny `outline`.

Tło

Najskuteczniejszym sposobem upiększania elementów za pomocą CSS jest nadawanie im tła. Ta właściwość jest używana bardzo często, nie tylko do tła strony, ale tła najróżniejszych elementów (np. graficzne przyciski, elementy menu, zaokrąglone rogi, czasem nawet wypunktowania list).

Tło wypełnia elementy do zewnętrznej krawędzi obramowania.

Tło jednolite: `background-color`

Przyjmuje dowolny kolor, którym zostanie wypełnione wnętrze elementu. Wartość `transparent` oznacza przezroczyste tło.

Najlepiej używać skróconego zapisu np.:

```
background: red; color: yellow
```

Nada stronie czerwone tło i żółty tekst.

Grafika

O ile to możliwe, ustawiaj kolor tła zawsze, gdy ustawiasz grafikę. W ten sposób element będzie miał widoczne tło zanim załaduje się grafika.

Każdy element może mieć ustawioną grafikę jako tło. Grafika nie wyklucza jednolitego koloru tła — jest rysowana nad nim.

Plik graficzny: `background-image`

Wskazuje (za pomocą funkcji `url()`) ścieżkę do pliku graficznego. Ścieżka jest zawsze względna do lokalizacji arkusza stylów.

```
background-image: url(ciapki.png)
```

W obecnej wersji CSS jeden element może mieć tylko jedno tło. Jeśli chcesz nałożyć na siebie więcej grafik (żeby zrobić np. zaokrąglone rogi w elemencie o zmiennej wielkości), musisz stworzyć dodatkowe zagnieżdżone elementy w XHTML (np. `<div>`) i nadać tło każdemu z nich.

Powtarzanie: `background-repeat`

`repeat`

: Grafika jest powtarzana, aby wypełniła całą powierzchnię tła. To jest domyślne zachowanie.

`repeat-x`

: Grafika jest powtarzana tylko w poziomie. Wypełnia całą szerokość, ale nie wysokość.

`repeat-y`

: Grafika jest powtarzana tylko w pionie. Wypełnia całą wysokość, ale nie szerokość.

`no-repeat`

: Grafika jest wstawiana tylko w jednym miejscu.

W zależności od powtarzania, grafika może mieć bardzo różne zastosowanie.

Można wstawić tylko pojedynczą grafikę (`no-repeat`) i zarezerwować na nią miejsce za pomocą `padding` — tekst nie będzie na nią wchodził, jakby to był zwykły obrazek.

Można robić ozdobne krawędzie i cienie za pomocą `repeat-x` i `repeat-y`.

```
background-image: url(zaokraglony-rozek.png);
background-repeat: no-repeat;
```

Pozycja grafiki: `background-position`

Kolejność współrzędnych `background-position` jest odwrotna, niż w skróconym zapisie właściwości takich, jak `margin` i `padding`.

Ustala na raz poziomą i pionową pozycję grafiki. Lewy górny róg jest domyślną wartością.

Dla wartości absolutnych (w pikselach) oznacza odsunięcie lewej górnej krawędzi grafiki od lewej górnej krawędzi tła.

Dla wartości w procentach oznacza odsunięcie o *x* procent punktu, który jest w *x* procent wymiaru grafiki. Dla 0% wyrównuje lewą krawędź grafiki z lewą krawędzią tła, dla 50% równo środkuje, a dla 100% wyrównuje prawą krawędź grafiki do prawej krawędzi tła.

```
background-position: 0 0;
```
Lewy górny róg. To są wartości domyślne.

```
background-position: 100% 100%;
```
Prawy dolny róg.

```
background-position: 10px 5px;
```
Grafika odsunięta o 10 pikseli od lewej i o 5 pikseli od góry.

```
background-position: 50% 0;
```
Grafika umieszczona na górze i wyśrodkowana w poziomie.

Nie da się odsunąć grafiki o określoną ilość pikseli od prawej lub dolnej krawędzi. Jeśli potrzebujesz uzyskać taki efekt, wstaw odpowiednią ilość pustego miejsca do grafiki i użyj `100%` jako pozycji.

Przesuwanie grafiki: `background-attachment`

position pozwala unieruchomić całe elementy

fixed

Tło pozostanie nieruchome względem okna przeglądarki

scroll

Tło przesuwa się razem ze stroną. To jest domyślne zachowanie.

Skrócony zapis właściwości tła: `background`

Wszystkie powyższe właściwości najlepiej ustawiać wszystkie na raz za pomocą właściwości `background`:

```
background: red url(foo.png) 5px 100% no-repeat;
```

Nie trzeba podawać wszystkich parametrów. Kolejność jest dowolna. Pominięte parametry przyjmą wartości domyślne.