1. **Elementy blokowe i liniowe**

Elementy języka HTML podzielone są na dwie grupy:

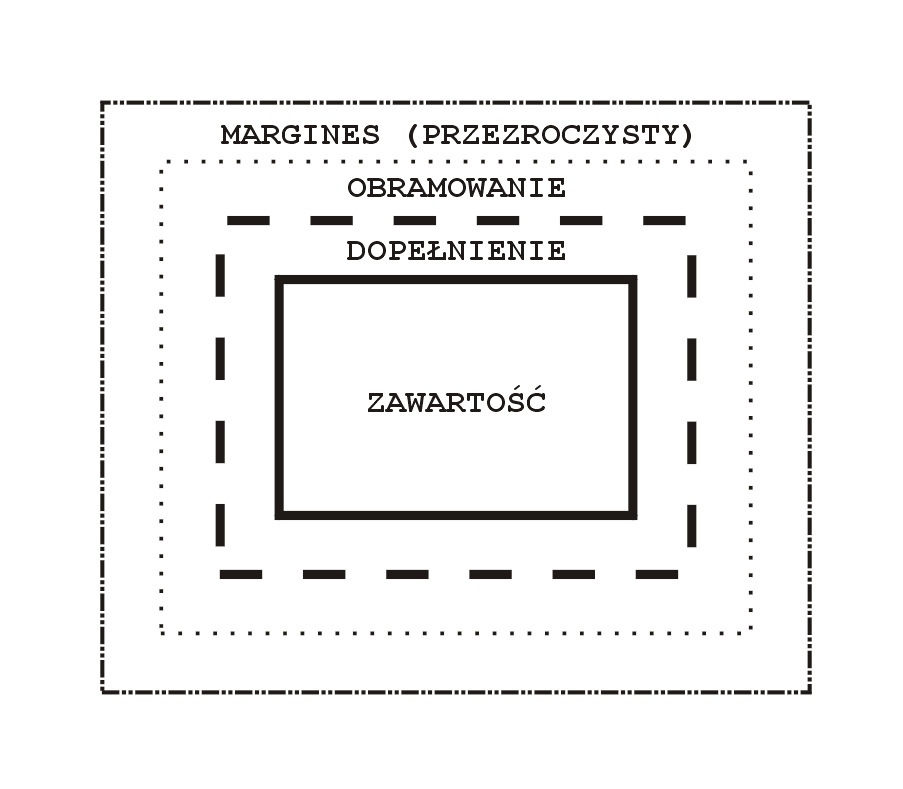
* Blokowe (np. div, p, h1..h6, ul, ol, figure) – domyślnie wyświetlane w trybie block, zajmują całą dostępną dla nich szerokość, a ich wysokość jest tak ustalona, by cała mieściła się w elemencie
* Liniowe (np. em, strong, span, br, a, link) – domyślnie wyświetlane w trybie inline. Szerokość i wysokość elementu jest tak dobrana by element pomieścił całą swoja zawartość.

Tryb wyświetlania każdego elementu możemy ustalić, wykorzystując rozszerzenia przeglądarek WWW.

1. **Obszar zajmowany przez element**

Każdy z elementów wyświetlanych przez przeglądarkę w obrębie treści strony jest zawarty w prostokątnym obszarze. Prostokąt jest podzielony na cztery obszary:

* Margines (zawsze przezroczysty - *margin*)
* Obramowanie (*border*)
* Dopełnienie (*padding*)
* Zawartość (*content*)



Do ustalenia wymiarów elementu służą właściwości **height** i **width (mogą być stosowane tylko do elementów blokowych!!!)**

Jeśli w elemencie **body** umieścisz element **div** o zawartości:

**<body>**

**<div>**

**<p>**

**Lorem ipsum**

**</p>**

**</div>**

**</body>**

I sformatujesz go stylami:

body {

margin: 0;

}

div {

margin: 100px;

border: 5px solid black;

padding: 40px;

width: 250px;

height: 150px;

}

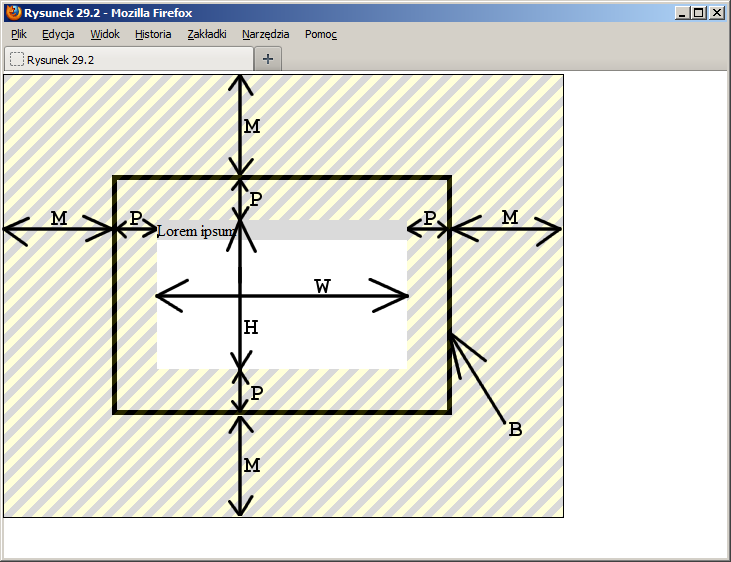
p {

background: #dadada;

margin: 0;

}

To otrzymasz efekt jak na rysunku:



Na element DIV musimy zatem przewidzieć prostokąt o wymiarach 540 x 440

1. **Właściwość display –** korzystając z tej właściwości każdy element możemy wyświetlić wedle uznania jako liniowy lub blokowy
2. **Wyśrodkowanie elementu blokowego –** należy skorzystać z właściwości **margin**. Przypisanie wartości auto lewemu oraz prawemu marginesowi spowoduje wyśrodkowanie elementu wewnątrz dostępnej przestrzeni

Div {margin: 0 auto;

}

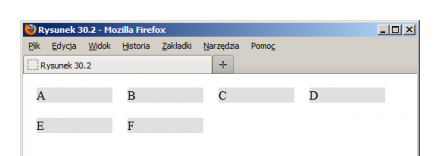
1. **Łączenie marginesów pionowych –** margines pionowy oddzielający elementy jest równy większej z podanych wartości, a nie ich sumie
2. **Wymiary minimalne i maksymalne –** służą do ustalenia min i max wymiarów elementu. Czynnikami które wpłyną na wymiary elementu, będą szerokość okna przeglądarki oraz ilość zawartego w nim tekstu:

* Min-width
* Max-width
* Min-height
* Max-height

1. **Elementy pływające**

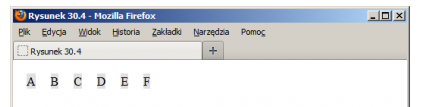
Elementy blokowe niezagnieżdżone są domyślnie pozycjonowane jeden pod drugim, a każdy z nich zajmuje całą dostępną szerokość.

1. **Właściwość float** – pozwala na tworzenie układów wielokolumnowych - float: left, right;



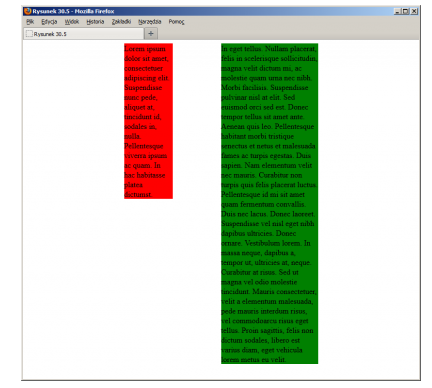
1. div {
2. width: 100px;
3. float: left;
4. background: gray;
5. margin: 10px;
6. }
7. ...
8. <div>A</div>
9. <div>B</div>
10. <div>C</div>

Jeżeli usuniemy właściwość **width** to efekt będzie taki jak na rysunku. Wniosek jest taki, że szerokość elementów pływających, dla których nie określono właściwości **width** jest minimalizowana.



1. **Układy kolumnowe**

Stosując elementy pływające, możemy w prosty sposób przygotować układy wielokolumnowe. Wystarczy wewnątrz pojemnika div o zadanej szerokości umieścić elementy pływające, których suma szerokości nie przekracza pojemnika

1. <div id="pojemnik">
2.     <div id="lewa">
3. Lorem ipsum...
4. </div>
5. <div id="prawa">
6. In eget tellus...
7. </div>
8. </div>

#pojemnik {

width: 400px;

margin: 0 auto;

}

#lewa {

width: 100px;

float: left;

background: red;

}

#prawa {

width: 200px;

float: right;

background: green;

}

Sprawdź co się stanie gdy szerokości kolumn przekroczą szerokość pojemnika!!!!

**Zadania**

1. **Wykonaj stronę jak na rysunkach:**

# Efekt rollover w oparciu o elementy div i style CSS

# <script type="text/javascript" src="rollover.js"></script>

### Wstępne pobieranie obrazów

Element **body** wzbogacamy o obsługę zdarzenia **onload**. W ten sposób obrazy wykorzystane w menu będą pobierane z serwera przy wejściu na stronę, a nie dopiero po najechaniu kursorem myszki na opcję, co mogłoby powodować — przy słabym łączu osoby odwiedzającej lub przeciążeniu sieci — niemiły efekt mrugnięcia lub wyświetlenie ikony brakującego obrazu.

<body onload="preload();">

W pliku **rollover.js** ustalamy liczbę wymieniamy obrazów (w opisywanym zadaniu wymianie podlegają trzy obrazy) oraz wpisujemy ich nazwy:

var rollovers\_count = 3;

var filenames\_off = new Array;

filenames\_off[0] = "g-1-0.png";

filenames\_off[1] = "g-1-1.png";

filenames\_off[2] = "g-1-2.png";

var filenames\_on = new Array;

filenames\_on[0] = "g-1-0-on.png";

filenames\_on[1] = "g-1-1-on.png";

filenames\_on[2] = "g-1-2-on.png";

****

główna:

<h1>Lorem</h1>

<p>

Aenean quis leo.

Vestibulum lorem.

Sed ut magna vel odio molestie tincidunt.

Pellentesque id mi sit amet quam fermentum convallis.

Curabitur non turpis quis felis placerat luctus.

Curabitur at risus.

<img src="img/foto/lorem-foto-1.jpg" alt="" class="l" />

Mauris consectetuer, velit a elementum malesuada, pede mauris interdum risus, vel commodoarcu risus eget tellus.

Duis nec lacus.

Suspendisse vel nisl eget nibh dapibus ultricies.

Vestibulum lorem.

Suspendisse vel nisl eget nibh dapibus ultricies.

Curabitur non turpis quis felis placerat luctus.

Aenean quis leo.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

Duis nec lacus.

Sed euismod orci sed est.

Mauris consectetuer, velit a elementum malesuada, pede mauris interdum risus, vel commodoarcu risus eget tellus.

In hac habitasse platea dictumst.

Nullam placerat, felis in scelerisque sollicitudin, magna velit dictum mi, ac molestie quam urna nec nibh.

Aenean quis leo.

Nullam placerat, felis in scelerisque sollicitudin, magna velit dictum mi, ac molestie quam urna nec nibh.

Suspendisse vel nisl eget nibh dapibus ultricies.

<img src="img/foto/lorem-foto-2.jpg" alt="" class="r" />

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

Suspendisse vel nisl eget nibh dapibus ultricies.

Pellentesque viverra ipsum ac quam.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Suspendisse vel nisl eget nibh dapibus ultricies.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

<img src="img/foto/lorem-foto-3.jpg" alt="" class="l" />

Mauris consectetuer, velit a elementum malesuada, pede mauris interdum risus, vel commodoarcu risus eget tellus.

In hac habitasse platea dictumst.

Suspendisse vel nisl eget nibh dapibus ultricies.

Vestibulum lorem.

Nam elementum velit nec mauris.

In eget tellus.

Vestibulum lorem.

Suspendisse vel nisl eget nibh dapibus ultricies.

Donec tempor tellus sit amet ante.

Aenean quis leo.

Vestibulum lorem.

Sed ut magna vel odio molestie tincidunt.

Pellentesque id mi sit amet quam fermentum convallis.

Curabitur non turpis quis felis placerat luctus.

Curabitur at risus.

Aenean quis leo.

Vestibulum lorem.

Sed ut magna vel odio molestie tincidunt.

Pellentesque id mi sit amet quam fermentum convallis.

Curabitur non turpis quis felis placerat luctus.

Curabitur at risus.

<img src="img/foto/lorem-foto-4.jpg" alt="" class="r" />

Duis sapien.

Donec ornare.

In hac habitasse platea dictumst.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Aenean quis leo.

Pellentesque viverra ipsum ac quam.

Curabitur at risus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Suspendisse vel nisl eget nibh dapibus ultricies.

Curabitur non turpis quis felis placerat luctus.

Aenean quis leo.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

Pellentesque id mi sit amet quam fermentum convallis.

Vestibulum lorem.

Nam elementum velit nec mauris.

Aenean quis leo.

Nullam placerat, felis in scelerisque sollicitudin, magna velit dictum mi, ac molestie quam urna nec nibh.

Aenean quis leo.

Vestibulum lorem.

Sed ut magna vel odio molestie tincidunt.

Pellentesque id mi sit amet quam fermentum convallis.

Curabitur non turpis quis felis placerat luctus.

Curabitur at risus.

Aenean quis leo.

Vestibulum lorem.

Sed ut magna vel odio molestie tincidunt.

Pellentesque id mi sit amet quam fermentum convallis.

Curabitur non turpis quis felis placerat luctus.

Curabitur at risus.

Nullam placerat, felis in scelerisque sollicitudin, magna velit dictum mi, ac molestie quam urna nec nibh.

Aenean quis leo.

</p>

Jedfne z bardziej popularnych zdarzeń JS to onmouseover i onmouseout. Pierwsze z nich powstaje, kiedy kursor myszy znajdzie się nad danym elementem strony, drugie — kiedy opuści obszar tego elementu. Pozwala to dynamicznie reagować na poczynania użytkownika.

<img src="img/lorem-off.png" alt="Lorem" id="img0" onmouseout="rollover\_off('img0', 0);" onmouseover="rollover\_on('img0', 0);" />