

Размери и позиции

width: число и мерни единици

ширина на елемента

height: число и мерни единици

височина на елемента

position: вид позициониране

Стойности за *вида на позициониране*:

✓ absolute

абсолютно позициониране в страницата;

✓ relative

относително позициониране;

✓ fixed

абсолютно позициониране в *прозореца* (при скролиране на страницата елементът остава **неподвижен**).

Задаване на позициите

left: число и мерни единици

top: число и мерни единици

right: число и мерни единици

bottom: число и мерни единици

z-index: число

позиция по Z-координатата (ако два елемента се припокриват, видим е този, който има по-голям **z-index**)

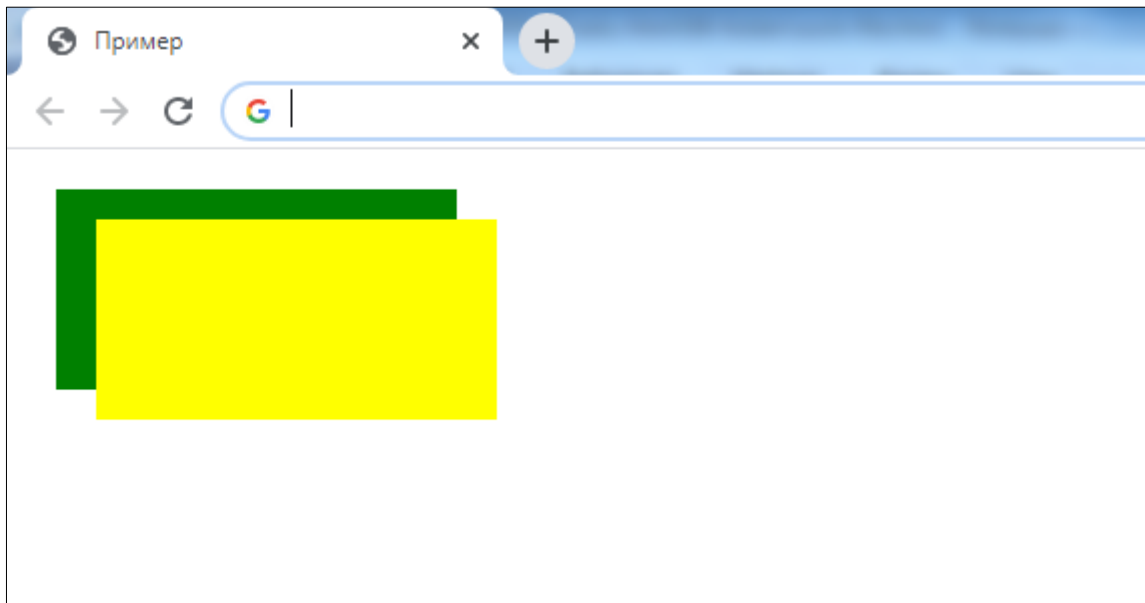
Пример:

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>Пример</title>
  <style>
    #div1
    {
      position:absolute;
      left:25px;
      top:20px;
      width:200px;
      height:100px;
      z-index:0;
      background:green;
    }

    #div2
    {
      position:absolute;
      left:45px;
      top:35px;
      width:200px;
      height:100px;
      z-index:1;
      background:yellow;
    }

  </style>
</head>
<body>
  <div id="div1">
  </div>
  <div id="div2">
  </div>
</body>
</html>
```

Резултат:



Отрязване

clip:rect(отрязване отгоре, отрязване отдясно,
отрязване отдолу, отрязване отляво)

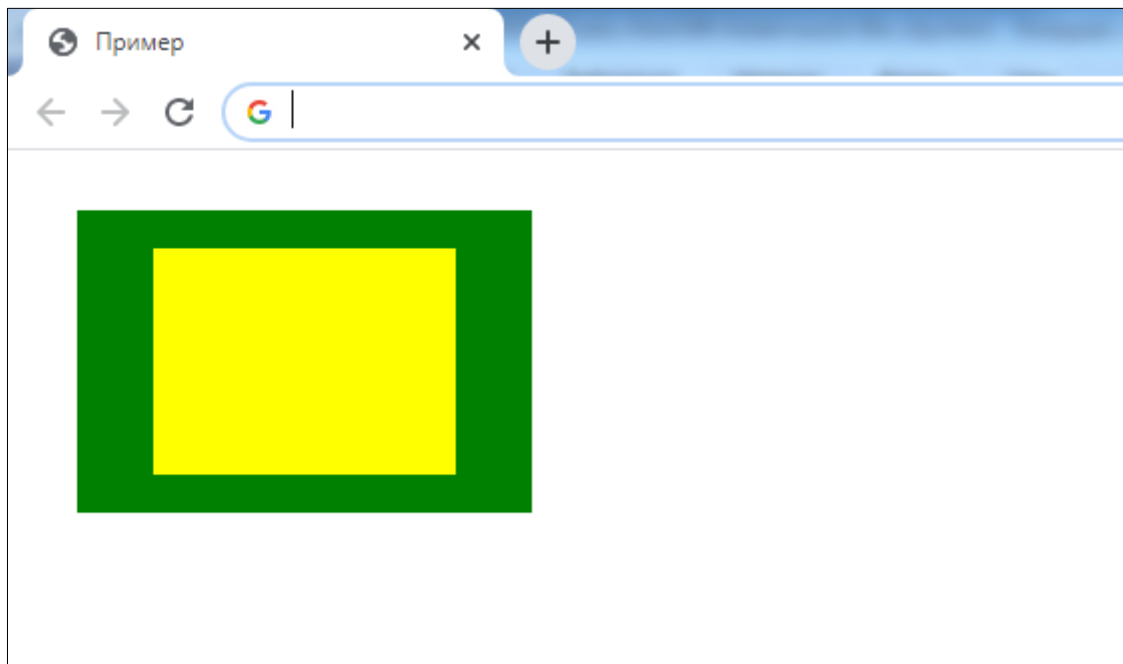
Отрязванията се задават с числа и мерни единици *относно горния ляв ъгъл* на обекта.

Пример:

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>Пример</title>
  <style>
    #div1
    {
      position:absolute;
      left:35px;
      top:30px;
      width:6cm;
      height:4cm;
      z-index:0;
```

```
        background:green;
    }
    #div2
    {
        position:absolute;
        left:35px;
        top:30px;
        width:6cm;
        height:4cm;
        z-index:1;
        background:yellow;
    }
    #div2:hover{clip:rect(0.5cm, 5cm, 3.5cm, 1cm)}
</style>
</head>
<body>
    <div id="div1">
    </div>
    <div id="div2">
    </div>
</body>
</html>
```

Резултат при *hover*:



Препълване

Блок с определени размери може да съдържа елементи, които излизат извън неговите граници, т.е. може да има **препълване**. Този проблем се решава с помощта на свойство **overflow**.

overflow

Стойности:

✓ visible

визуализират се и елементите извън границите на блока;

✓ hidden

елементите извън границите на блока остават скрити;

✓ scroll

в блока се добавят скролери (плъзгачи);

✓ auto

автоматично се добавят скролери (плъзгачи) в случай на препълване.

Аналогично, само по **хоризонтала** и съответно по **вертикала**:

overflow-x

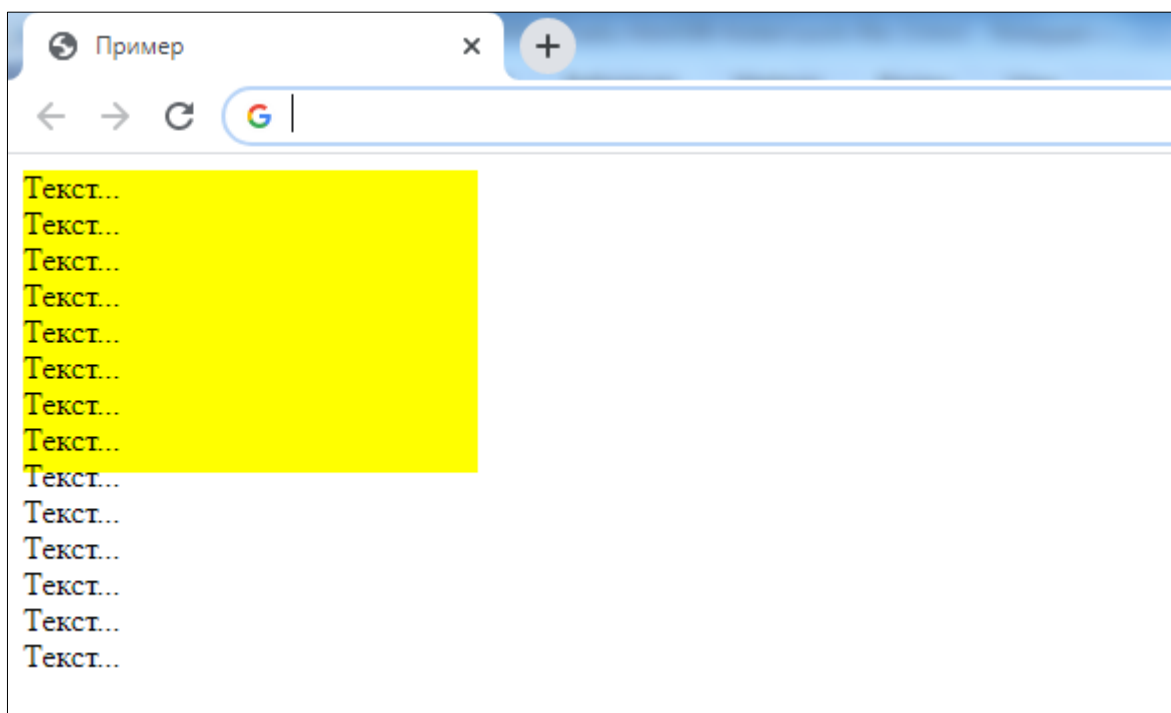
overflow-y

Пример:

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>Пример</title>
  <style>
    #div1
    {
      position:relative;
      width:6cm;
      height:4cm;
      background:yellow;
```

```
        overflow:hidden;
    }
    #div1:hover{ overflow:visible }
</style>
</head>
<body>
    <div id="div1">
        Текст...<br>
        Текст...<br>
        Текст...<br>
        Текст...<br>
        Текст...<br>
        Текст...<br>
        Текст...<br>
        Текст...<br>
        Текст...<br>
        Текст...<br>
        Текст...<br>
        Текст...<br>
        Текст...<br>
        Текст...<br>
        Текст...<br>
    </div>
</body>
</html>
```

Резултат при *hover*:



Видимост на елементите

Свойство:

`visibility`

Стойности:

✓ `visible`

елементът е видим (това е и стойността по подразбиране);

✓ `hidden`

елементът става невидим, но запазва мястото си на страницата.

Свойство:

`display`

Стойности:

✓ `inline`

елементът се визуализира в страницата;

✓ `block`

елементът се визуализира в страницата като блоков елемент;

✓ `none`

елементът се скрива и *освобождава мястото си в страницата*.

Свойство:

`opacity`:

задава степен на *прозрачност* на елемента (0-пълна прозрачност, 1-пълна плътност).

Блокова структурирана страницата

Посредством свойството ***float*** може да се задава разположение на елементите (блоковете) един спрямо друг.

Стойности:

ляво подравняване, т.е. елементът (блокът) ще бъде ляво подравнен до предходния или до лявата граница на страницата;

дясно подравняване.

Пример за блокова структура на страница:

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Блокова структура на страницата</title>
<!-- В общия случай стиловете за блоковете се записват в отделен CSS-
файл, който се използва от всяка страница в един сайт. -->
<style>
#top
{
    height:80px;
    width:100%;
    float:left;
    background:#aaaaaa;
    border-bottom:1px solid black;
}
#left
{
    width:15%;
    float:left;
    height:320px;
    background:#cccccc;
}
#content
{
    width:85%;
    float:left;
    height:320px;
    background:#ffaa11;
}
```



```
</style>
</head>
<body>
<div id="top">
  <h1 align="center">Горна секция ("шапка" на страницата)</h1>
</div>
<div id="left">
  <p style="text-indent:0.75cm; color:#0000ff; cursor:pointer">
    <u>Лавата секция (блок) обикновено се използва за меню от
    линкове.</u>
  </p>
</div>
<div id="content">
  <h1 style="text-indent:1cm">
    Основна секция (съдържание на страницата)
  </h1>
</div>
</body>
</html>
```

Резултат:

