

Блочная модель

HTML

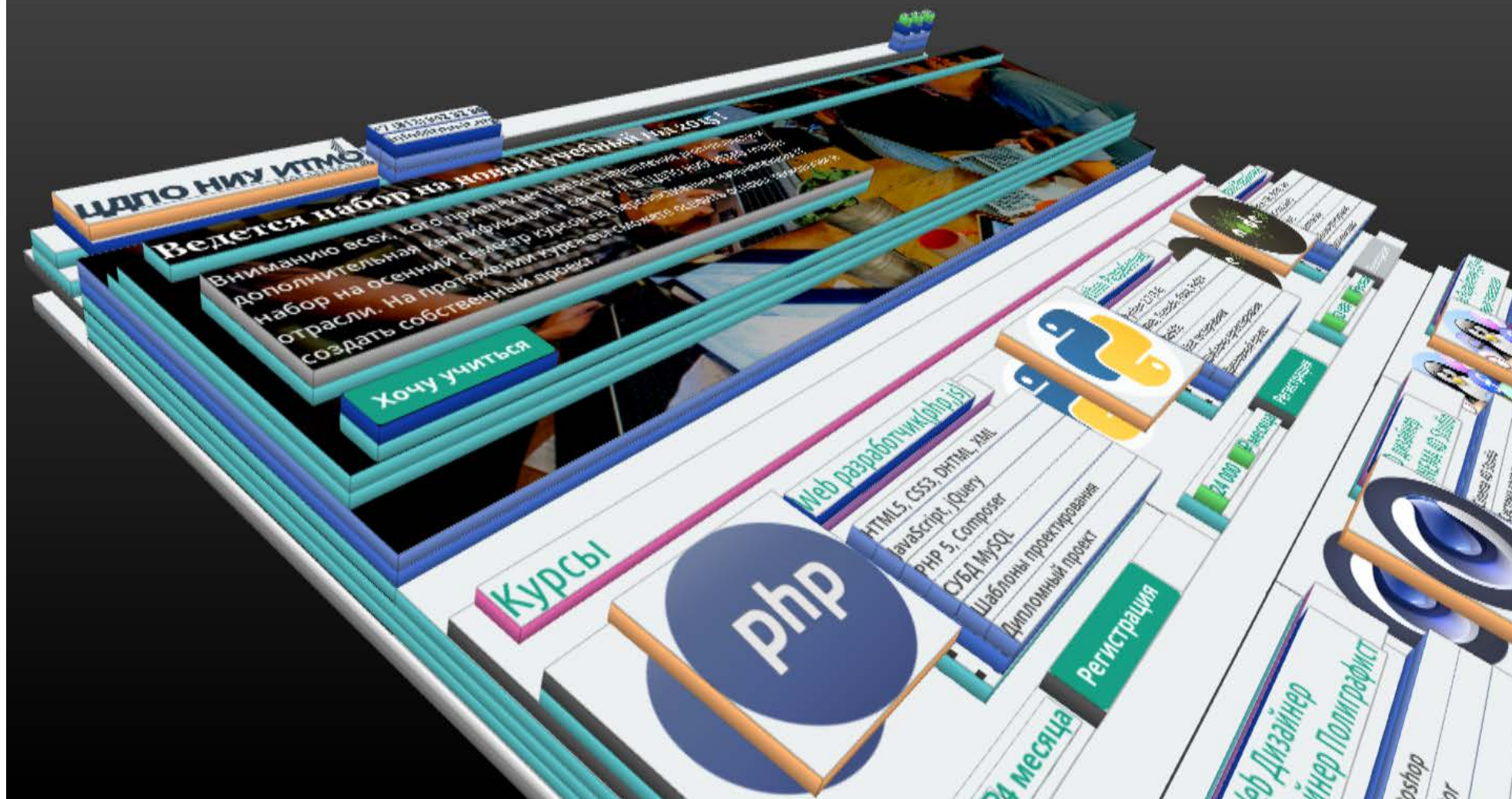


CSS

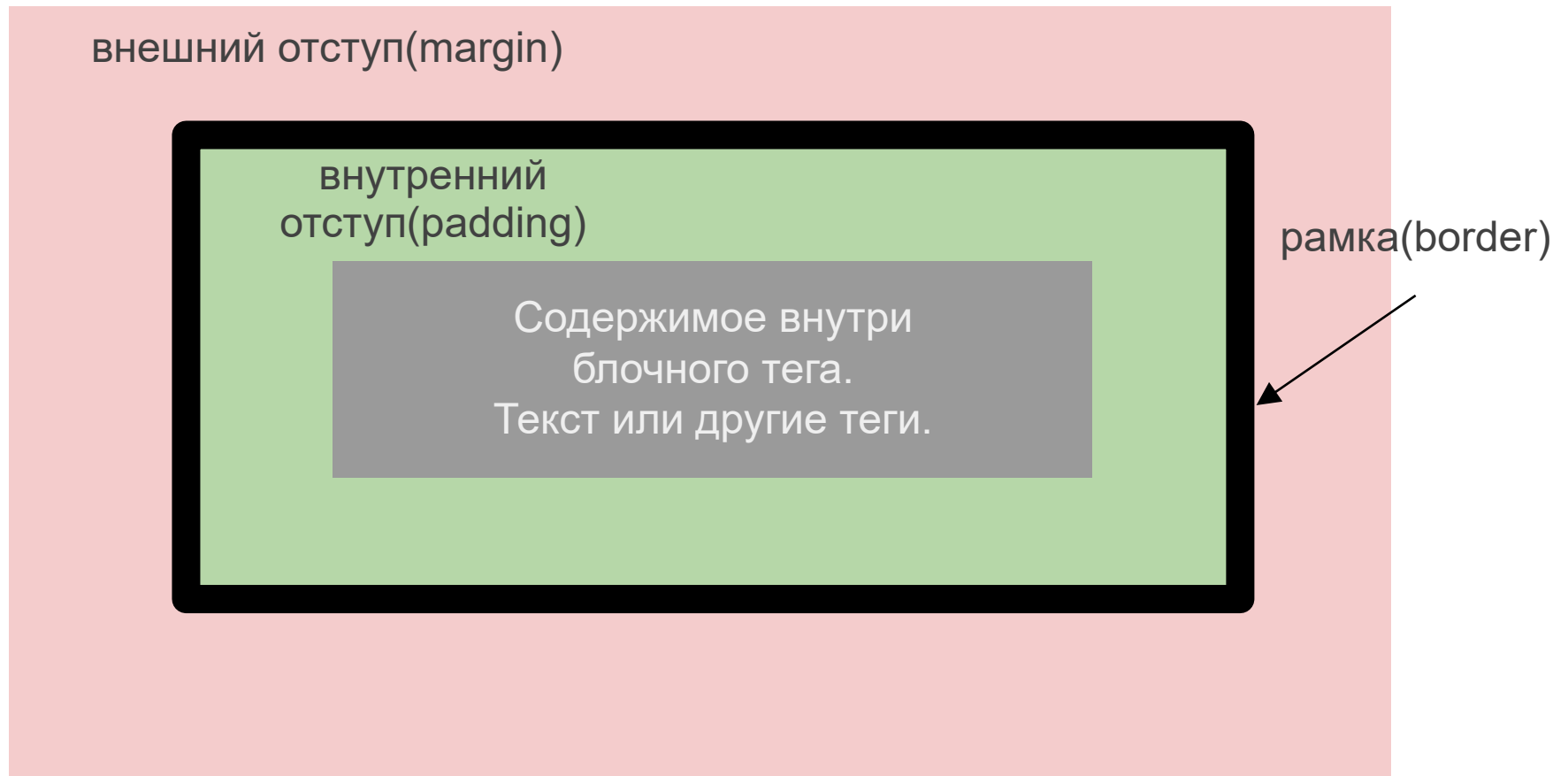


Блочные элементы

Иллюстрация того, как браузер рендерит контейнеры элементов.



Как выглядит блок



Основной частью каждого блока элемента является **область содержимого**.

Область содержимого окружена произвольным количеством **отступов (padding)**, **рамок (border)** и **полей (margin)**.

Как задавать поля

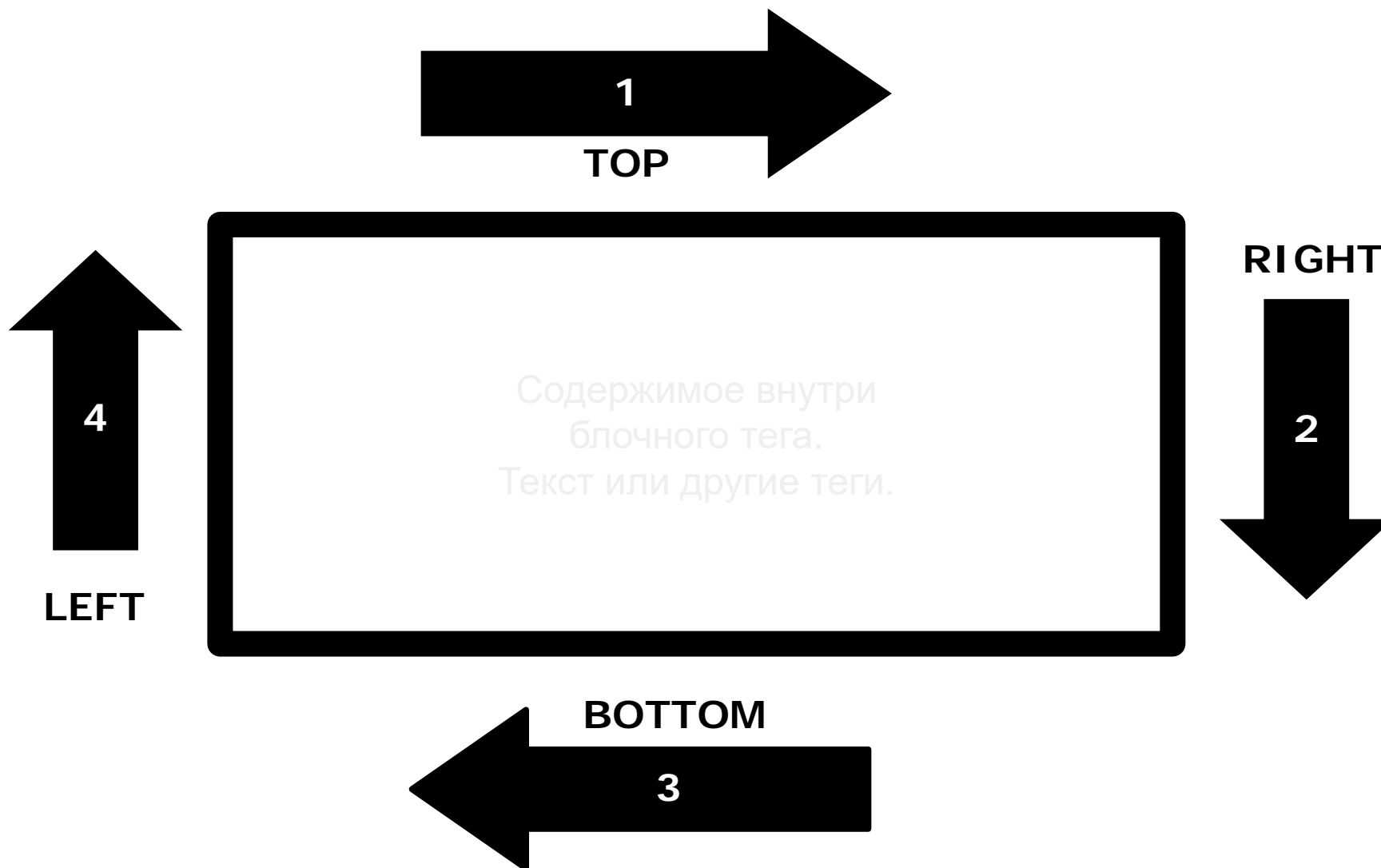
```
.block {  
  margin-top: 20px;  
  margin-right: 10px;  
  margin-bottom: 20px;  
  margin-left: 10px;
```

/ универсальное свойство */*

```
margin: 10px;  
margin: 20px 10px;      /* top/bottom - right/left */  
margin: 20px 10px 20px; /* top - right/left - bottom */  
margin: 20px 10px 20px 10px;
```

```
}
```

Порядок установки значений margin и padding


























Как задавать отступы

```
.block {  
  padding-top: 20px;  
  padding-right: 10px;  
  padding-bottom: 20px;  
  padding-left: 10px;  
  
  /* универсальное свойство */  
  
  padding: 10px;  
  padding: 20px 10px;           /* top/bottom - right/left */  
  padding: 20px 10px 20px;     /* top - right/left - bottom */  
  padding: 20px 10px 20px 10px;  
}
```



Как задавать рамку

```
.block {  
  /* универсальное свойство → border: ширина стиль цвет */  
  border: 1px solid #000;  
  border: 3px dotted #ccc;  
}
```

1 пиксел	3 пиксела	5 пикселей	7 пикселей
			
			
			
			
			
			
			
			

Как вычисляются размеры блока

width: значение | проценты | auto | inherit
height: значение | проценты | auto | inherit

1. **Фактическая ширина блока по стандарту равна:**

$$\text{fullWidth} = \text{width} + \text{padding-left} + \text{padding-right}$$
$$+ \text{border-left} + \text{border-right}$$
$$+ \text{margin-left} + \text{margin-right}$$

2. **Фактическая высота блока по стандарту равна:**

$$\text{fullHeight} = \text{height} + \text{padding-top} + \text{padding-bottom}$$
$$+ \text{border-top} + \text{border-bottom}$$
$$+ \text{margin-top} + \text{margin-bottom}$$

```
.block {  
  border: 1px solid #000;  
  padding: 20px 10px;  
  height: 100px;      /* Чему равна высота блока? */  
  width: 200px;       /* Чему равна ширина блока? */  
}
```

Как изменить алгоритм расчета размеров блока

box-sizing: content-box | border-box | padding-box | inherit

1. **Фактическая ширина блока с border-box равна:**
$$\text{fullWidth} = \text{width} + \text{margin-left} + \text{margin-right}$$
2. **Фактическая высота блока с border-box равна:**
$$\text{fullHeight} = \text{height} + \text{margin-top} + \text{margin-bottom}$$

```
.block {  
  border: 1px solid #000;  
  margin: 10px;  
  padding: 10px;  
  
  box-sizing: border-box;  
  
  height: 100px;      /* Чему равна высота блока? */  
  width: 200px;       /* Чему равна ширина блока? */  
}
```

Чем блочные элементы отличаются от строчных

1. **в блочные** элементы можно вкладывать **любые элементы**;
2. **строчные** элементы могут содержать **только данные** или другие **строчные элементы**;
3. **блочные** элементы всегда **начинаются с новой строки**;
4. **блочные** элементы **занимают всю** доступную **ширину**;
5. **ширина строчных** элементов равна их **содержимому** плюс значения отступов, полей и границ.

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Asperiores tempora, nam provident, dolor quam rem accusamus, officiis beatae sunt velit harum natus placeat dolore quae, voluptatum ullam at molestiae minus.

Cupiditate quisquam ab qui explicabo quae ea eum voluptas, nesciunt earum doloremque fugit repellat nemo! Velit, corporis. Architecto fugit quam aut, blanditiis, at excepturi rem omnis doloribus animi cupiditate nostrum!

Lorem ipsum dolor. Maxime nisi, facere! Minus, mollitia, totam? Ut, voluptas, nemo. Eligendi nesciunt, accusantium. Amet, similique, dolor. Harum, fugiat, nulla? Nemo, eaque, velit. Velit, cumque, optio! Eveniet, quis, eligendi. Assumenda, debitis esse. Necessitatibus, quos ex. Dolorum, error, tempora! Illum, odit, animi! Quis, maiores voluptatem! Error, adipisci, nobis! Repellendus, perferendis doloremque. Ipsa, mollitia, quas. Porro, eveniet, iure? Mollitia, amet, dolorem. Nam, nemo, repellat. Possimus ad, facere! Perspiciatis, magni, debitis. Vel, assumenda illum. Fugit, ea, beatae.

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Corporis ea iure necessitatibus enim cupiditate saepe, quod iusto quaerat. Cum rerum nihil eligendi voluptas pariatur est veniam doloribus atque, dolorem facere.

Ex perspiciatis dolore animi aut doloribus esse error deserunt. Harum dolores, atque quis dolorum, quae ullam vitae, deserunt doloremque unde consequuntur accusantium sed vel. Voluptates minus explicabo voluptatum blanditiis repudiandae.

Lorem ipsum dolor. Quasi, fuga, ipsa. Soluta doloremque, modi! Cumque, fugiat ipsum. Aut, deleniti, cupiditate. Aspernatur, unde, nostrum! Fugiat, reiciendis nemo. Officia, vero, placeat. Ipsum, mollitia, suscipit! Adipisci, reiciendis, odio? Modi praesentium, fugit? A, eveniet, quidem! Dolor, magni, voluptates. Nemo, fugiat, quisquam. Minima, ab, dolore. Ipsa, natus, exercitationem. Aliquam, ut nisi. Officiis consequatur, illum. Quod, amet laborum! Neque, ipsam, incidunt. Ex, quas esse. Ipsam, temporibus, aspernatur! Ratione, vel, aspernatur! Impedit, molestiae, a. Ea, aliquid nostrum.

Как изменить вид отображения элемента

display: block | inline | inline-block | none | flex ...

Полный список значений на htmlbook.ru¹.

```
span {  
  display: block; /* Отобразить строчный элемент как блочный */  
}  
  
div {  
  display: inline; /* Отобразить блочный элемент как строчный */  
}  
  
div {  
  display: table-cell; /* Отобразить элемент как ячейку таблицы */  
}
```

¹. все остальные допустимые значения поддерживаются браузерами выборочно.

Как скрыть элемент

visibility: visible | hidden | collapse | inherit

- выделяется место под элемент — псевдо-прозрачный
p.s. display none → полностью скрывает элемент;
- не использовать для вывода элементов в одно место.

Плавающие элементы, поток

float: left | right | none | inherit

Плавающий элемент:

- становится блочным (display: block);
- размер определяется объемом его содержимого;
- не взаимодействует с соседними блочными элементами;
- обтекается строчными элементами;
- не может плавать по центру.

```
.col-6 {  
  float: left;  
  padding: 0 10px;  
  width: 50%;  
}
```

Клиринг

clear: none | left | right | both | inherit

```
<div class="sidebar" style="float: left">This is Sidebar</div>  
<div class="content" style="float: left">This is Content</div>  
<div class="footer">This is Footer</div>
```



```
.footer {  
  clear: none;  
}
```

```
.footer {  
  clear: right;  
}
```



```
.footer {  
  clear: both;  
}
```

```
.footer {  
  clear: left;  
}
```


Если контенту мало места

overflow: auto | hidden | scroll | visible | inherit

- без overflow браузеры различно обрабатывают "переполнение" блок

overflow-x: auto | hidden | scroll | visible

overflow-y: auto | hidden | scroll | visible

Верстка сеткой

col-1	col-2	col-3	col-4	col-5	col-6	col-7	col-8	col-9	col-10	col-11	col-12
.col-1											
.col-2											
.col-3											
.col-4				.col-4				.col-4			
		.col-4.offset-2						.col-3.offset-2			

clearfix

```
<div class="wrapper">
  <div class="row">
    <header class="col-12">
      <h1>CSS Grid</h1>
    </header>
  </div>
  <div class="row">
    <div class="col-9"></div>
    <div class="col-3"></div>
  </div>
  <div class="row">
    <footer class="col-12"></footer>
  </div>
</div>
```

```
/* clearfix */
.row:after {
  clear: both;
  content: ".";
  display: block;
  height: 0;
  visibility: hidden;
}
```

Позиционирование элементов

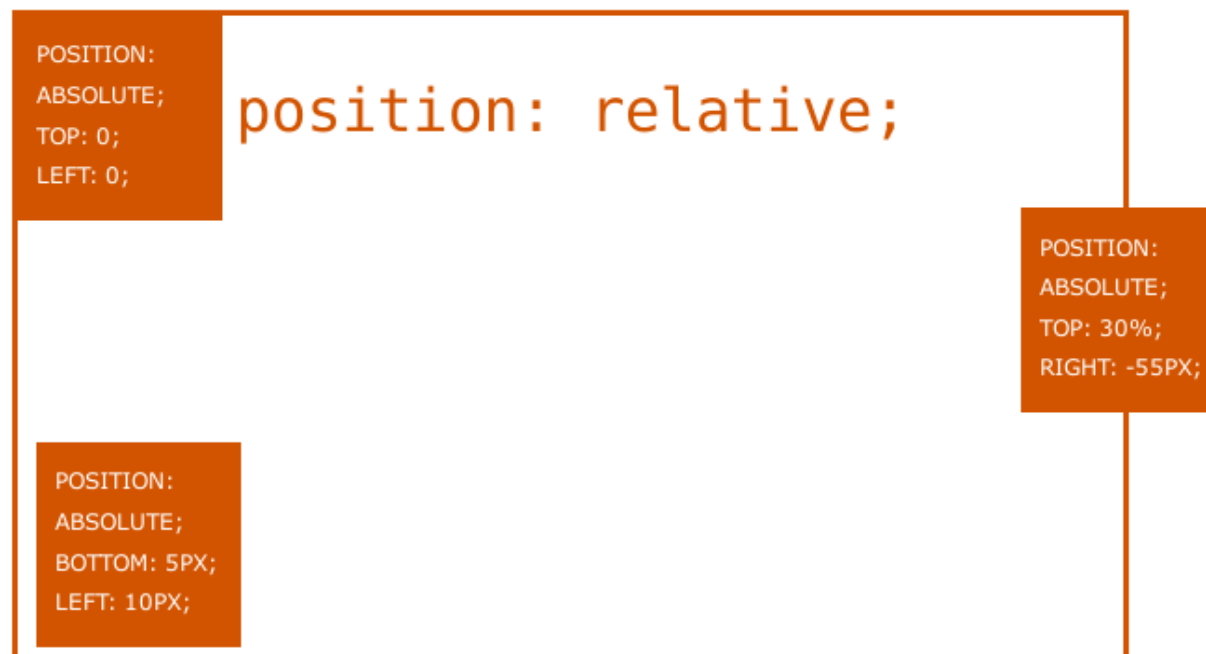
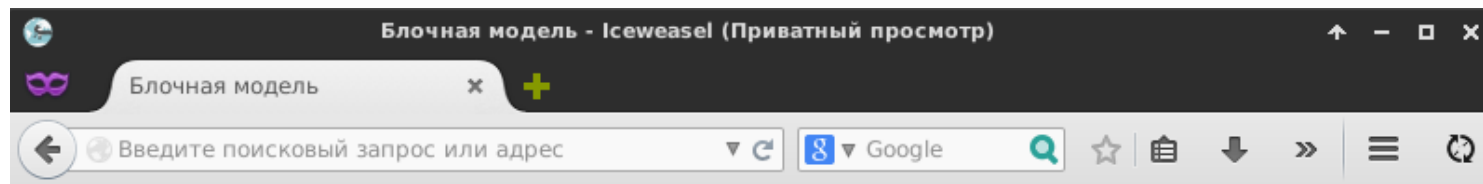
position: **absolute** | **fixed** | **relative** | **static** | **inherit**

- **static:** по-умолчанию – нормальный поток
- **relative:** относительное – от места расположения элемента
- **absolute:** абсолютное – от ближайшего элемента с position: relative
- **fixed:** фиксированное – относительно окна браузера (видимой области)

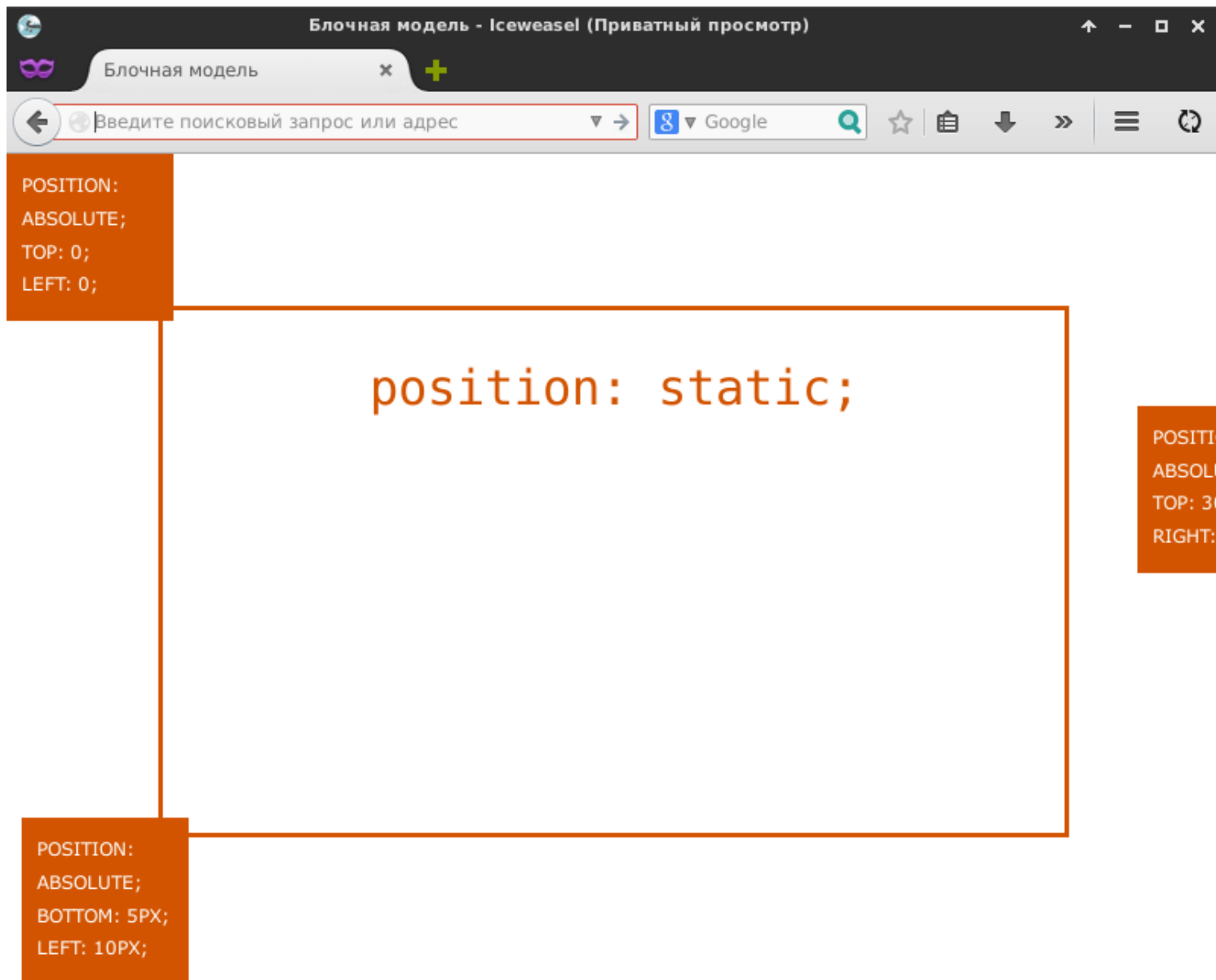
Элементам с position absolute:

- обязательно задавать координаты;
- становятся блочным (display: block);
- размер определяется объемом заключенного в него содержимого;
- не участвуют в margin-collapse (схлопывающие отступы);
- другие элементы не видят элементов с position: absolute.

Относительное позиционирование



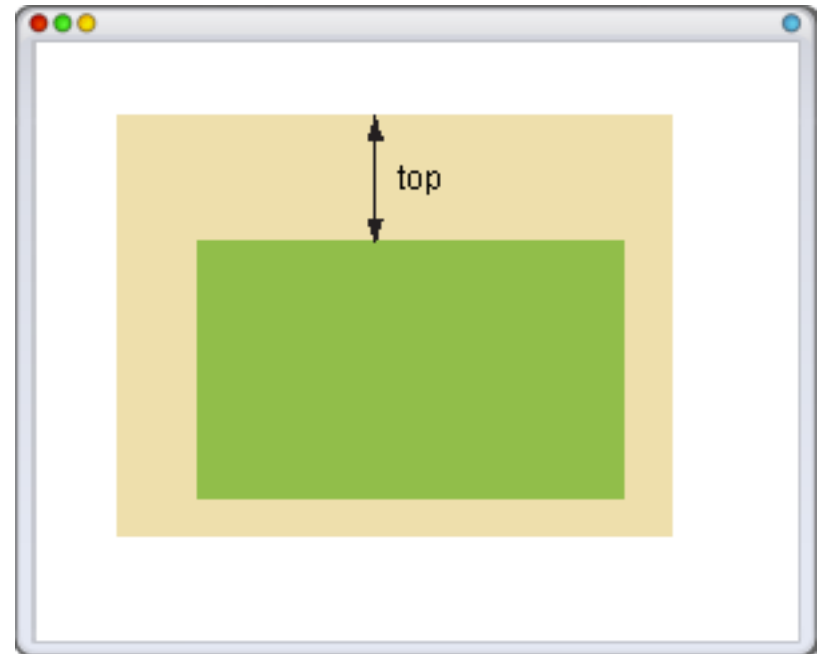
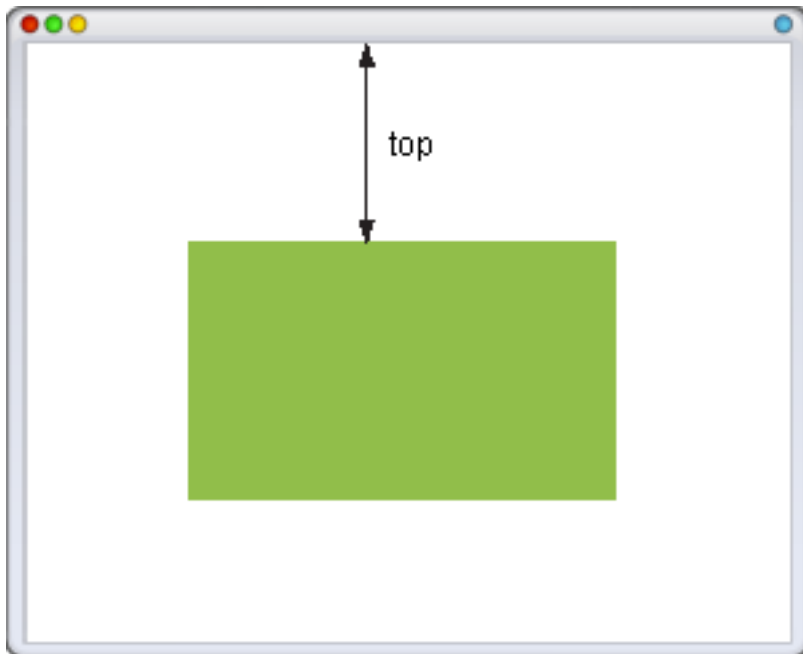
Позиционирование по-умолчанию



Как элементу задать координаты

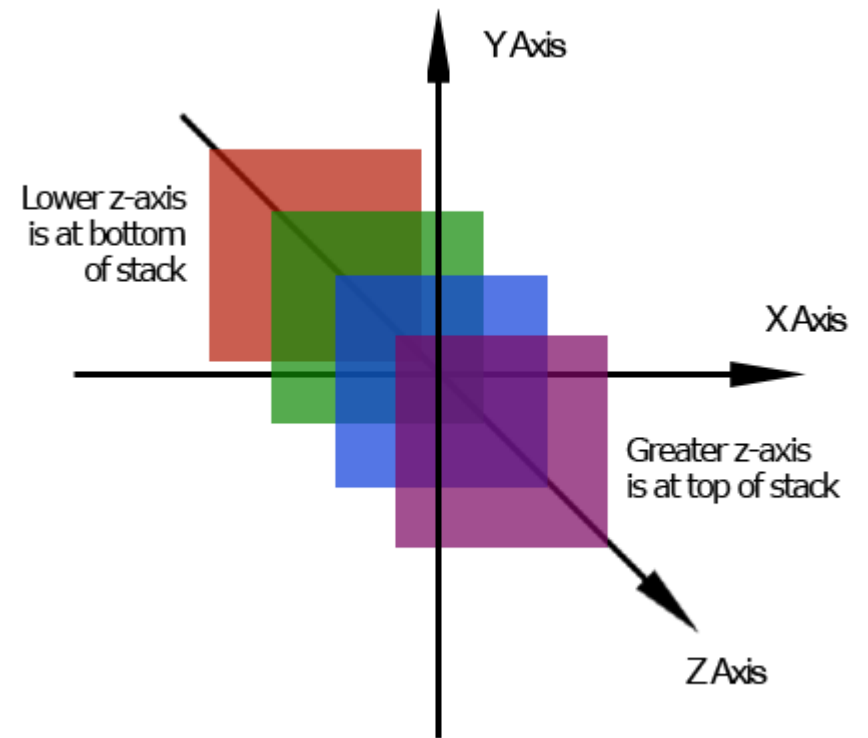
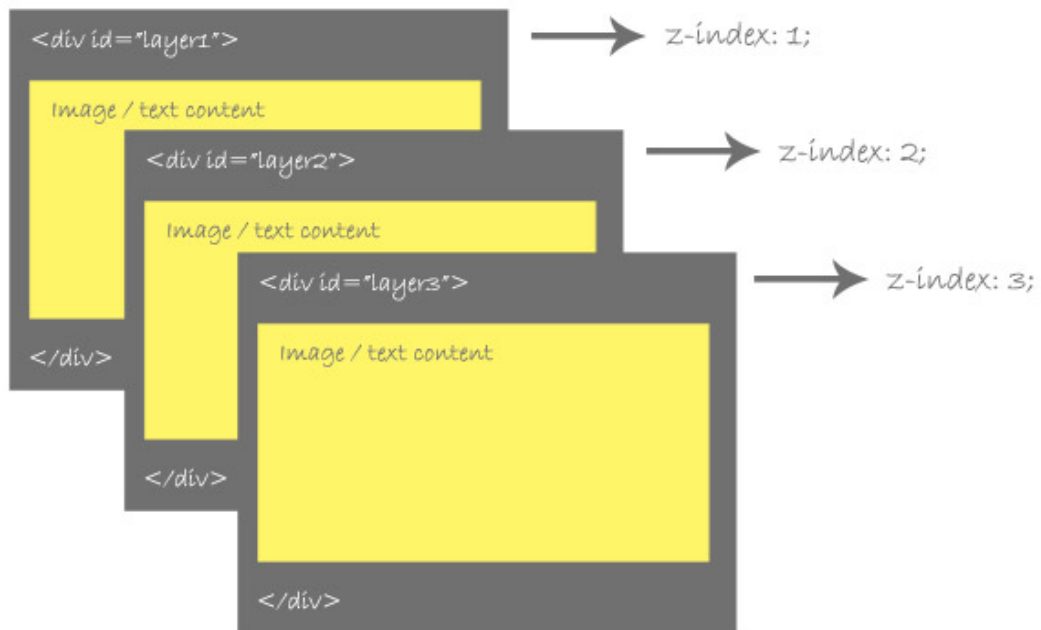
top: value | percent | auto | inherit

- **top:** от верхнего края родительского элемента
- **right:** от правого края
- **bottom:** от нижнего края
- **left:** от левого края



z-index

z-index: число | auto | inherit



Как показать часть элемента

`clip: rect(Y1, X1, Y2, X2) | auto | inherit`

- что не влезло → обрезается
- только прямоугольная область
- **только для абсолютно позиционированных элементов**

