

## План урока к восьмой паре

### ● Позиционирование. Верстка web-страниц блоками

Свойство **position**:

- рассмотрение позиционирования: **relative** и **absolute**;

#### Типы позиционирования

- Свойство **position** позволяет точно задать новое местоположение блока относительно того места, где он находился бы в нормальном потоке документа. По умолчанию все элементы располагаются последовательно один за другим в том порядке, в котором они определены в структуре **HTML**-документа. Свойство не наследуется.

#### Position

Значение:

**static** - Значение по умолчанию, означает отсутствие позиционирования. Элементы отображаются последовательно один за другим в том порядке, в котором они определены в **HTML**-документе. Используется для очистки любого другого значения позиционирования.

**relative** - Относительно позиционированный элемент сдвигается со своего обычного места в разных направлениях относительно границ родительского контейнера, а пространство, которое он занимал, не исчезает. При этом такой элемент может перекрывать другое содержимое на странице.

Если для относительно позиционированного элемента одновременно задать свойства **top** и **bottom** или **left** и **right**, то в первом случае сработает только **top**, во втором — **left**.

Относительное позиционирование позволяет задавать **z-index** для элемента, а также абсолютно позиционировать дочерние элементы внутри блока.

**absolute** - Абсолютно позиционированный элемент полностью удаляется из потока документа и позиционируется относительно границ его блока-контейнера (другого элемента или окна браузера). Блок-контейнер для абсолютно позиционированного элемента — ближайший элемент-предок, значение свойства **position** которого не равно **static**.

Местоположение краёв элемента определяется с помощью свойств смещения. Пространство, которое занимал такой элемент, схлопывается, как будто элемента не существовало на странице. Абсолютно позиционированный элемент может перекрывать другие элементы или быть перекрытым ими (за счёт свойства **z-index**). Любой абсолютно позиционированный элемент генерирует блок, то есть принимает значение **display: block;**

**fixed** - Фиксирует элемент в указанном месте страницы. Блоком-контейнером фиксированного элемента является окно просмотра, то есть элемент всегда фиксируется относительно окна браузера и не меняет своего положения во время прокрутки страницы. Сам элемент при этом полностью удаляется из основного потока документа и создаётся в новом потоке документа.

**initial** - Устанавливает значение свойства в значение по умолчанию.

**inherit** - Наследует значение свойства от родительского элемента.

**sticky** - Это сочетание относительного и фиксированного позиционирования. Элемент рассматривается как позиционированный относительно, пока он не пересекает определённый порог, после чего рассматривается как фиксированный. Обычно применяется для фиксации заголовка на одном месте, пока содержимое, к которому относится заголовок, прокручивается на странице.

- свойства **top, left, bottom, right**.

- Свойства описывают смещение относительно ближайшей стороны блока-контейнера. Задаются для элементов, для которых значение свойства **position** не равно **static**. Могут принимать как положительные, так и отрицательные значения. Не наследуются.

**top, right, bottom, left**

Значение:

**auto** - Значение по умолчанию. Вычисляемое значение свойства равно нулю.

**длина** - Смещение задаётся в единицах длины.

**%** - Процентные значения вычисляются относительно высоты блока-контейнера для **top** и **bottom** и ширины блока-контейнера для **right** и **left**.

**initial** - Устанавливает значение свойства в значение по умолчанию.

**inherit** - Наследует значение свойства от родительского элемента.

- Для свойства **top** положительные значения перемещают верхний край позиционируемого элемента ниже, а отрицательные — выше верхнего края его блока-контейнера. Для свойства **left** положительные значения сдвигают край позиционируемого элемента вправо, а отрицательные значения — влево. То есть, положительные значения смещают элемент внутрь блока-контейнера, а отрицательные — за его пределы.

#### 1. Свойства **visibility**, **overflow**.

- Для вёрстки страниц часто используются два основных инструмента — позиционирование (**positioning**) и свободное перемещение (**floating**). CSS-позиционирование позволяет указать, где появится блок элемента, а свободное перемещение перемещает элементы к левому или правому краю блока-контейнера, позволяя остальному содержимому «обтекать» его.

**CSS display** и **visibility** или видимость элементов

**visibility** — определяет видимость или невидимость элемента:

**visibility**: inherit | visible | hidden;

visibility: **inherit** - наследует видимость родительского элемента;

visibility: **visible** - делает элемент страницы видимым (по умолчанию);

visibility: **hidden** - делает невидимым.

**HTML:**

```
<div>
  <p>Some Text Goes Here ...</p>
</div>
```

**CSS:**

```
div {
  visibility: visible;
}
```

**HTML:**

```
<div>
  <p>Some Text Goes Here ...</p>
</div>
```

**CSS:**

```
div {
  visibility: hidden;
}
```

**display** — определяет способ отображения элементов страницы:

**display:** *inline* | *block* | *none* | *inline-block* | *list-item*

**display: inline** - значение по умолчанию;

**display: block** - наделяет элементы свойствами блока;

**display: none** - элемент не отображается;

**display: inline-block** - аналогично **block**;

**display: list-item** - аналогично **block**, но при этом элемент страницы считается позицией списка.

войства **CSS overflow** или полосы прокрутки

**overflow** — определяет будет ли элемент страницы иметь полосу прокрутки:

**overflow:** *visible* | *scroll* | *hidden* | *auto*

**overflow: visible**

- границы элемента расширяются по его содержимому (по умолчанию);

**overflow: scroll** - если содержимое элемента не вмещается в его границы – добавляются полосы прокрутки;

**overflow: hidden** - обрезает содержимое элемента по установленным размерам;

**overflow: auto** - полосы прокрутки отображаются если в этом есть необходимость.

## 2. Практика.

### HTML:

```
<div class="success left-abs"></div>
```

```
<div class="warning center-abs"></div>
```

```
<div class="danger right-abs">
```

```
  <p>{ position: absolute; }</p>
```

```
</div>
```

```
<div class="success left-rel"></div>
```

```
<div class="warning center-rel"></div>
```

```
<div class="danger right-rel">
```

```
  <p class="rel">{ position: relative; }</p>
```

```
</div>
```

CSS:

**body {**

**color:** *aliceblue;*

**font-family:** *"segoe ui", helvetica, verdana, sans-serif;*

**}**

**div {**

**width:** *250px;*

**height:** *200px;*

**text-align:** *center;*

**}**

**p {**

**font-size:** *20px;*

**vertical-align:** *middle;*

**padding-top:** *65px;*

**}**

**p.rel {**

**padding-top:** *85px;*

**}**

**success {**

**background:** *#206E3C;*

**}**

**.warning {**

**background:** *#D38E3B;*

**}**

**.danger {**

**background:** *#7C2020;*

**}**

**.left-abs {**

**position:** *absolute;*

**top:** *0px;*

**left:** *0px;*

**}**

**.center-abs {**

**position:** *absolute;*

**top:** *50px;*

**left:** *50px;*

**}**

```
.right-abs {  
    position: absolute;  
    top: 100px;  
    right: 650px;  
}  
.left-rel {  
    position: relative;  
    top: -10px;  
    left: 750px;  
}  
.center-rel {  
    position: relative;  
    top: -160px;  
    left: 700px;  
}  
.right-rel {  
    position: relative;  
    top: -330px;  
    right: -650px;  
}
```

### 3. Основы верстки блоками. Правила верстки:

- вложение блоков;

Верстка **<div>** (старая)

#### HTML:

```
<body>  
    <!-- Header -->  
    <div id="headerInner">  
        <!-- Navigation -->  
        <div class="bottomMenu">  
            <!-- ..... -->  
        </div><!-- / Navigation -->  
    </div><!-- / Header -->
```

```
<div id="wrapper"><!-- Main block -->
  <div id="colLeft"><!-- Block with post -->
    <!-- Post start -->
    <div class="postBox">
      <!-- ..... -->
    </div><!-- / Post end -->
    <!-- Post start -->
    <div id="colRight"> <!-- Right col block -->
      <!-- ..... -->
    </div><!-- / Right col block end -->
  </div> <!-- / Block with post -->
</div>
<!-- Footer -->
<div id="footerInner">
  <!-- ..... -->
</div><!-- / Footer end -->
</body>
```

Сначала давайте рассмотрим теги, которые используются в первую очередь, другими словами основные:

**<header>** — данный тег задаёт верхнюю часть сайта, или шапку сайта. В него может входить меню, логотип, и другие элементы которые используются в верхней части сайта.

**<section>** — данный тег задаётся для важных разделов на сайте, например для таких как статьи, или же важные моменты, которые Вы хотите выделить.

**<article>** — обычно в этом теге содержаться разнообразные записи на сайте, новости, документы. Например посты на блоге.

**<aside>** — задаётся для боковых колонок на сайте (Сайтбара). Не важно где он будет находится слева или справа.

**<footer>** — в этот тег должен входить низ Вашего сайта (Футер)

**<hgroup>** — в данный тег входят группы заголовков сайтов, например **<h1><h6>**.

**<nav>** — данный тег включает в себя всю навигацию на сайте (меню).

Хочу сказать, что эти все теги обязательно должны закрываться, например Если это тег **<article>** он должен иметь закрывающий тег **</article>** и так далее.