Как CSS Visibility улучшает веб-дизайн с помощью скрытых драгоценных камней

25.05.2023

У нас много свойств CSS, и освоение всех может оказаться непростой задачей. Видимость CSS – одно из важных свойств, которым вы должны овладеть, если хотите стать квалифицированным веб-разработчиком. В этой статье я дам определение CSS Visibility, объясню его важность, а также перечислю и объясню различные значения CSS visibility.

Что такое видимость CSS?

Свойство CSS visibility скрывает или показывает элемент на веб-странице. Например, вы можете разместить на веб-странице четыре поля и использовать свойство видимости, чтобы определить, как они будут отображаться. Все элементы будут отображаться на странице, если вы установите свойство видимости как visible. Однако если элемент скрыт, он будет по-прежнему занимать место, но будет скрыт от конечного браузера/экрана.

CSS Видимость важна в следующих случаях:

- **Контроль видимости:** Вы можете контролировать то, что должно отображаться в зависимости от текущего пользователя. Вы можете установить видимость элемента, чтобы он был виден только тогда, когда пользователь активирует его с помощью определенного действия. Например, наведение курсора или нажатие на кнопку.
- **Сохранение макета:** Хорошее приложение должно сохранять свой макет и содержимое независимо от размера экрана. Когда вы устанавливаете видимость элемента как скрытую,

он по-прежнему будет занимать место, но не будет виден конечным пользователям. Такой подход позволяет поддерживать последовательную компоновку.

- **Оптимизация производительности:** Браузеру не нужно постоянно пересчитывать макет, когда свойство видимости установлено как visibility:hidden. Однако при использовании свойства display:none браузеру приходится пересчитывать макет всякий раз, когда вы решаете снова показать элемент.
- **Создание динамичного и интерактивного пользовательского интерфейса:** Вы можете комбинировать свойство CSS visibility с другими свойствами, такими как opacity, для создания эффектов затухания, анимации и плавных переходов.

Различные значения видимости CSS

CSS visibility имеет пять возможных значений. Я буду подробно описывать их с помощью блоков кода и скриншотов. Если вы намерены следовать за мной,

- Создайте папку на вашей локальной машине.
- Добавьте два файла: index.html и styles.css.
- Откройте проект в вашем любимом редакторе кода (я использую VS Code).

Вы можете использовать эту команду:

```
mkdir -p CSS-visibility && cd CSS-visibility && touch index.html styles.css
```

Далее необходимо соединить файлы **index.html** и **styles.css**. В разделе **<head>** файла index.html добавьте следующее:

```
<link rel="stylesheet" href="styles.css">
```

Теперь в вашем редакторе кода должно быть что-то похожее на

это:

Visible

Когда вы устанавливаете значение элемента как visibility: visible, он будет отображаться на веб-странице. Эта видимость присутствует по умолчанию. Чтобы лучше понять эту концепцию, мы можем сделать три поля в нашем HTML-документе. В разделе **<body>** вашего **index.html** добавьте следующее:

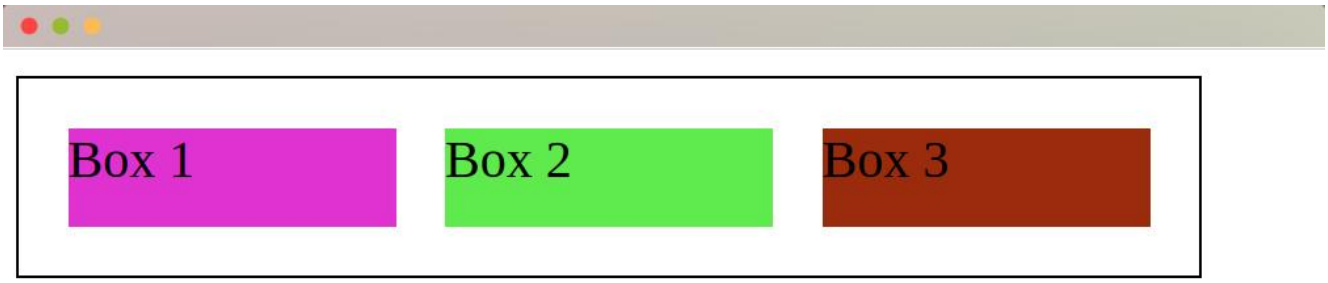
```
<div class="container">

      <div class="box1"> Box 1 </div>

      <div class="box2"> Box 2 </div>

      <div class="box3"> Box 3 </div>

    </div>
```

Теперь мы можем стилизовать наши divs (поля) с помощью следующего кода CSS:

```
.container {

    padding: 15px;

    width: max-content;

    display: flex;

    border: 1px solid black;

  }

  .box1,
```

```
  .box2,
  .box3 {
    height: 30px;
    width: 100px;
  }
  .box1 {
    background-color: rgb(224, 49, 209);
    margin-right: 15px;
  }
  .box2 {
    background-color: rgb(95, 234, 77);
    margin-right: 15px;
  }
  .box3 {
    background-color: rgb(154, 43, 12);
  }
```

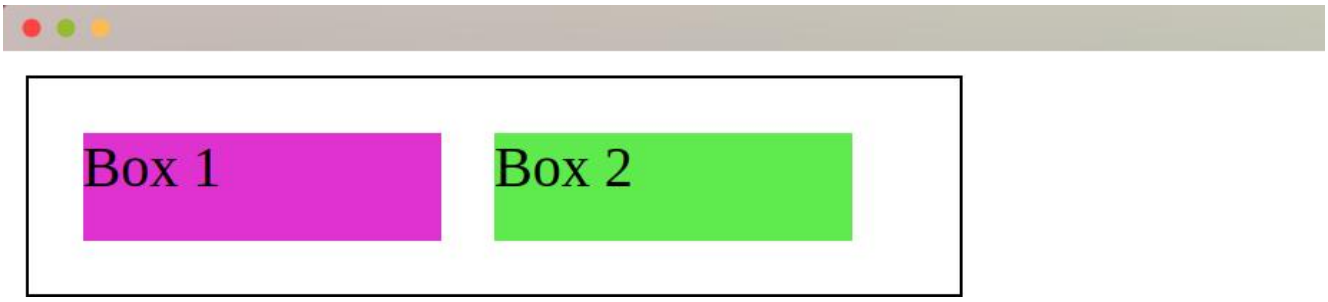
Когда конечная страница будет отображена, вы получите что-то вроде этого:



Если вы установите видимость для ящиков как ***visibility: visible***, это не будет иметь никакого эффекта, поскольку по умолчанию все ящики видимы. Однако мы можем продемонстрировать, как работает свойство visible, используя свойство display: none для одного из ящиков. В качестве примера мы можем выбрать box3. Измените CSS-код класса .box3 следующим образом:

```
.box3 {

    background-color: rgb(154, 43, 12);

    display: none

  }
```

Когда вы повторно отобразите страницу, вы заметите, что у нас есть только два ящика, один и два.



Если вы внимательны, то заметите, что наш элемент .container уменьшился в размерах. Когда вы используете свойство display: none, вставка 3 не отображается, и ее место в браузере становится свободным для других вставных элементов.

Hidden

Когда мы применяем к элементу CSS свойство visibility: hidden, он будет скрыт от конечного пользователя. Однако он по-прежнему будет занимать место. Например, мы можем скрыть Box2 с помощью этого свойства.

Для этого измените CSS-код Box2 следующим образом:

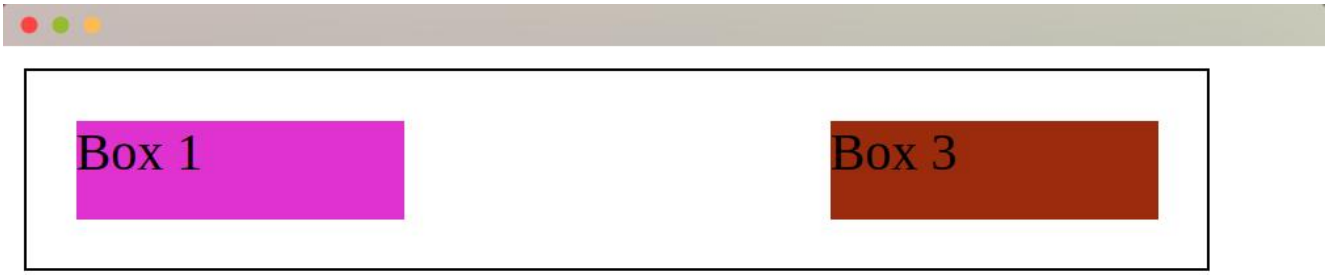
```
.box2 {
```

```
    background-color: rgb(95, 234, 77);
    margin-right: 15px;
    visibility: hidden
  }
```

Единственное изменение, которое мы внесли, это добавление этой строки:

`visibility: hidden`

Конечная страница будет выглядеть так, как показано на рисунке:



Мы видим пространство между вставкой 1 и вставкой 3, поскольку элемент вставки 2 только скрыт. Мы можем видеть, что Box 2 все еще находится в DOM, если проверим нашу отрендеренную страницу.

Collapse

Collapse – это значение видимости, которое используется для вложенных элементов. HTML-таблицы – прекрасный пример того, где можно применить атрибут ***visibility:collapse***.

Мы можем добавить следующий код для создания таблицы в нашем

HTML-файле:

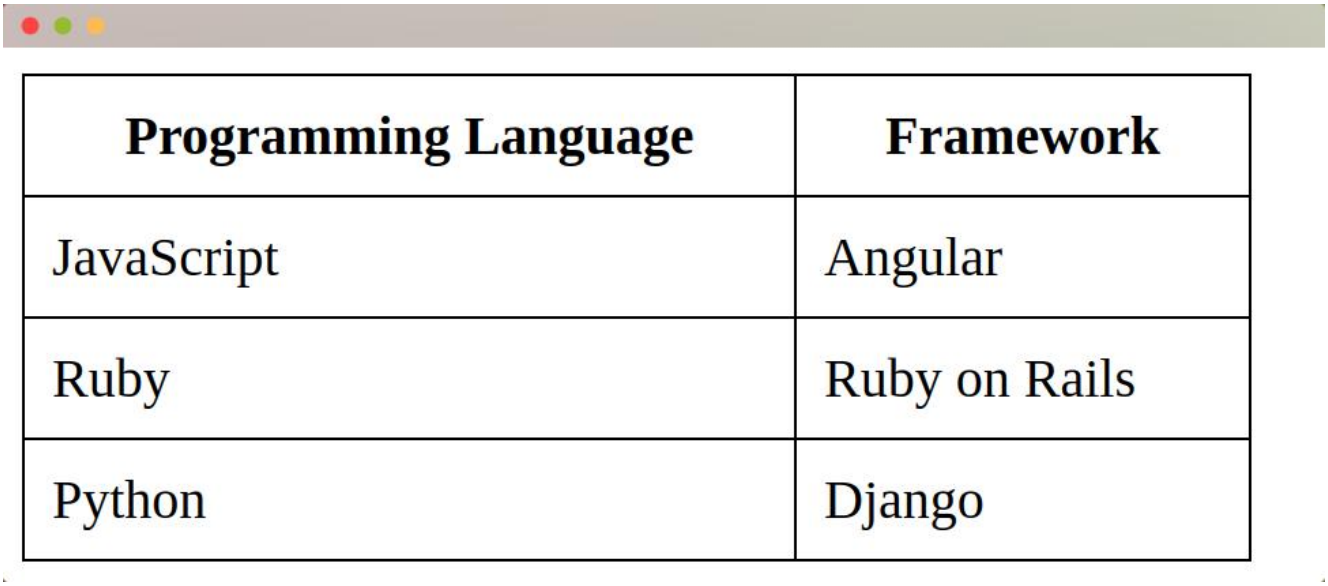
```
<table>
        <tr>
          <th>Programming Language</th>
          <th>Framework</th>
        </tr>
        <tr>
          <td>JavaScript </td>
          <td>Angular </td>
        </tr>
        <tr class="hidden-row">
          <td>Ruby </td>
          <td>Ruby on Rails</td>
        </tr>
        <tr>
          <td>Python </td>
          <td>Django </td>
        </tr>
   </table>
```

Теперь мы можем стилизовать нашу таблицу с границей в 1px для всех ее ячеек. Добавьте это в ваш файл CSS:

```
table {
```

```
    border-collapse: collapse;

    width: 50%;

  }

  th, td {

    border: 1px solid black;

    padding: 8px;

  }
```

Когда мы отобразим таблицу, мы получим следующее:



Programming Language	Framework
JavaScript	Angular
Ruby	Ruby on Rails
Python	Django

Теперь мы скроем второй ряд, чтобы продемонстрировать, как работает атрибут visibility: collapse. Добавьте это в свой код CSS:

```
.hidden-row {

    visibility: collapse;

  }
```

Когда страница будет отображена, строка с **Ruby** и **Ruby on Rails** будет скрыта.

Programming Language	Framework
JavaScript	Angular
Python	Django

Initial

Свойство visibility: initial возвращает HTML-элемент к его начальному значению. Например:

- Мы начали с отображения всех строк нашей таблицы.
- Мы свернули значение строки 2, тем самым скрыв ее.
- Теперь мы можем вернуться к исходному значению (отобразить его).

Чтобы продемонстрировать это, мы добавим кнопку, которая переключается между отображением и сворачиванием значений.

Добавьте эту кликабельную кнопку в HTML-код:

```
<button onclick="toggleVisibility()">Click Me!!</button>
```

Затем мы можем добавить функцию JavaScript, которая ищет класс .hidden-row и переключает на него класс .visible-row при нажатии на кнопку.

```
<script>

    function toggleVisibility() {

      const hiddenRow = document.querySelector('.hidden-row');

      hiddenRow.classList.toggle('visible-row');

    }
```

```
  </script>
```

Add this code to your CSS file;

```
.visible-row {

    visibility: initial;

  }
```

Значение видимости будет переключаться туда и обратно по мере нажатия на отображаемую кнопку.

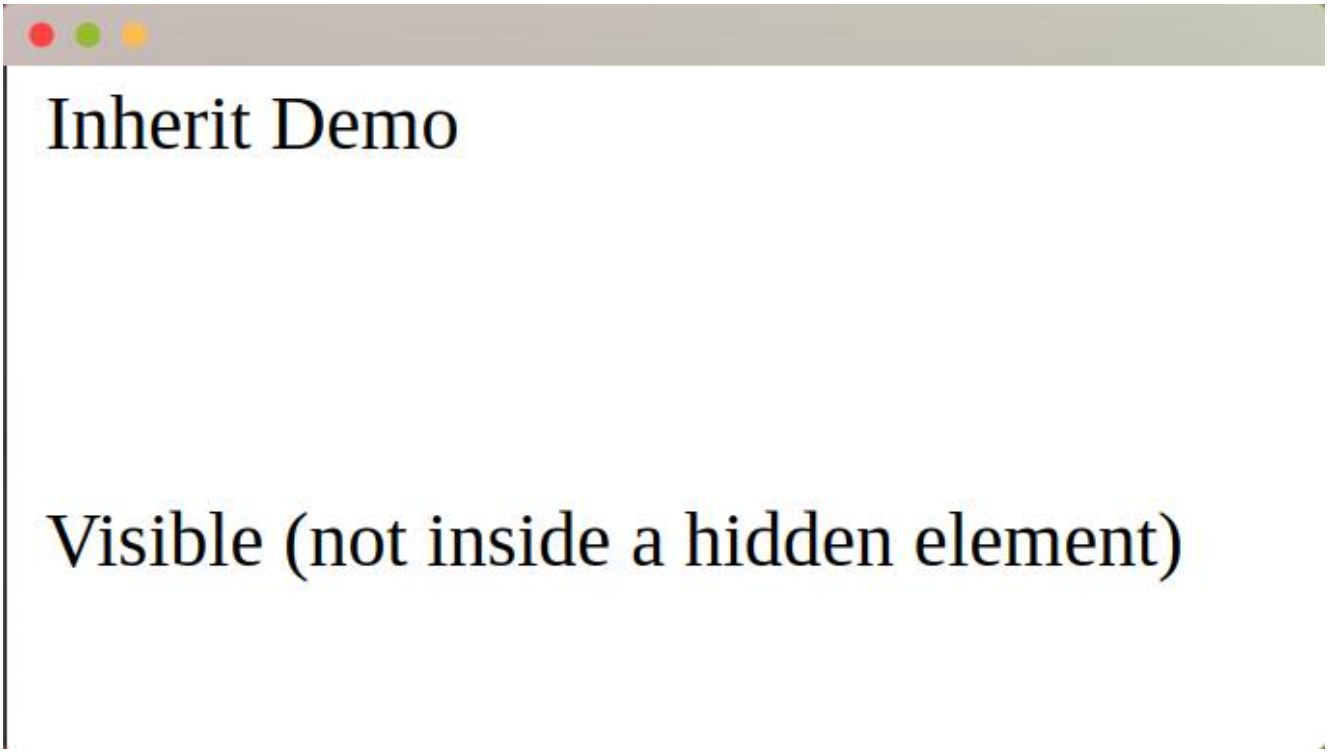
Inherit

Свойство ***visibility : inherit*** позволяет дочернему элементу наследовать свойство display от родительского.

Для демонстрации работы этого свойства мы можем воспользоваться следующим простым кодом:

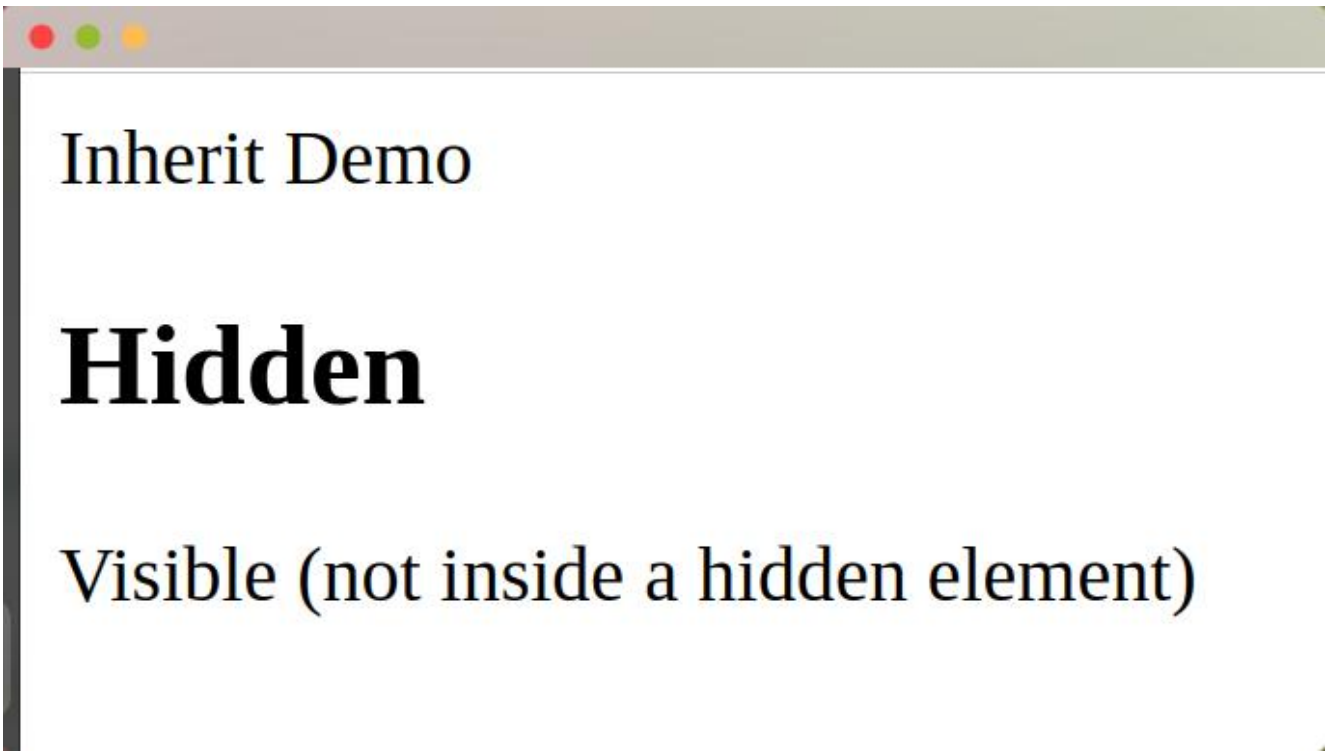
```
<h1>Inherit Demo</h1>

   <div style="visibility: hidden">

     <h2 style="visibility: inherit"> Hidden</h2>

   </div>

   <p>Visible (not inside a hidden element) </p>

 </p>
```

При рендеринге страницы будет отображаться только содержимое внутри тегов h1 и paragraph. Однако между тегами <h1> и <p> останется пространство, представляющее скрытые элементы. У нас есть один div, видимость которого мы установили как скрытую. Внутри div находится тег <h2>. Мы установили видимость <h2> как inherit, что означает, что он наследуется от своего родителя, div.



Однако если мы установим свойство div как visible, то <h2> (его дочерний элемент) также унаследует это свойство.

```
<h>Inherit Demo</h>

    <div style="visibility: visible">

      <h2 style="visibility: inherit"> Hidden</h2>

    </div>

    <p>Visible (not inside a hidden element) </p>
```



CSS visibility: hidden в сравнении с display: none

Основное различие между display:none и visibility:hidden заключается в том, что первый полностью удаляет элемент из макета, а второй скрывает элемент, но при этом занимает место.

Мы можем использовать этот код, чтобы продемонстрировать разницу между ними:

```
<!DOCTYPE html>

<html>

<head>

    <style>

        .box {

          display: inline-block;

          width: 100px;
```

```
      height: 100px;
      background-color: lightgray;
      margin-right: 20px;
    }
    .box1 {
      background-color: lightblue;
    }
    .box2 {
      background-color: black;
    }
    .container {
        padding: 10px;
        border :2px solid black;
        padding-left: 20px;
        width: 250px;
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="box box1"></div>
```

```
        <div class="box box2"></div>

      </div>

</body>

</html>
```

Если мы отобразим нашу страницу, то получим два поля рядом друг с другом:

Демонстрация display: none

Добавьте это в класс .box1:

```
display: none;
```

Когда вы отобразите страницу, вы заметите, что .box1 больше не отображается. Второй бокс (черный) также перемещается влево и занимает место, ранее занимаемое светло-голубым боксом.

Демонстрация visibility: hidden

Вместо ***display: none*** добавьте этот класс .box1:

visibility: hidden

Это свойство оставляет место для box1, но не отображает его. С другой стороны, box2 остается на своем месте.

display:none	visibility:hidden

Полностью удаляет элемент с веб-страницы	Скрывает элемент HTML, но при этом занимает место на веб-странице
Вы можете придать стиль элементу, для которого установлено значение none	Вы можете расположить и придать стиль элементу, видимость которого скрыта
Недоступно для чтения с экрана	Вы можете получить доступ к элементу, видимость которого установлена как скрытая, с помощью устройств чтения с экрана

Улучшение доступности с помощью видимости CSS

Вы можете использовать видимость CSS, чтобы скрыть элементы, которые не важны для всех пользователей. Например, вы можете использовать эту функцию для скрытия входа в систему, который будет доступен только тем пользователям, которые не вошли в систему. Вы также можете иметь боковую панель или нижний колонтитул, содержащий условия предоставления услуг или информацию об авторских правах.

Мы можем взять этот код, чтобы проиллюстрировать, как можно улучшить видимость:

```
<!DOCTYPE html>

<html>

  <head>

    <title>Homepage</title>

    <style>

      .login-form {
```

```
        display: none;
      }
      .login-text:hover + .login-form {
        display: block;
      }
      .login-form label {
        display: block;
        margin-bottom: 5px;
      }
      .login-form input {
        width: 30%;
        margin-bottom: 10px;
      }
    </style>
  </head>
  <body>
    <p class="login-text">Login</p>
    <form class="login-form">
      <label for="username">Username:</label>
        <input type="text" id="username" name="username"
required />
```

```
      <label for="password">Password:</label>

       <input type="password" id="password" name="password"
required />

      <button type="submit">Submit</button>

    </form>

  </body>

</html>
```

Форма входа в систему становится видимой только при наведении курсора на первый элемент.

Заключение

Мы уверены, что теперь вы можете комфортно использовать свойство CSS visibility в своем коде для создания плавных переходов и улучшения доступности ваших веб-страниц. Однако вы должны знать, где использовать каждое значение видимости, чтобы не получить в итоге нескладные страницы. Вы также можете скрыть важные данные от конечных пользователей, если неправильно используете свойство CSS visibility.