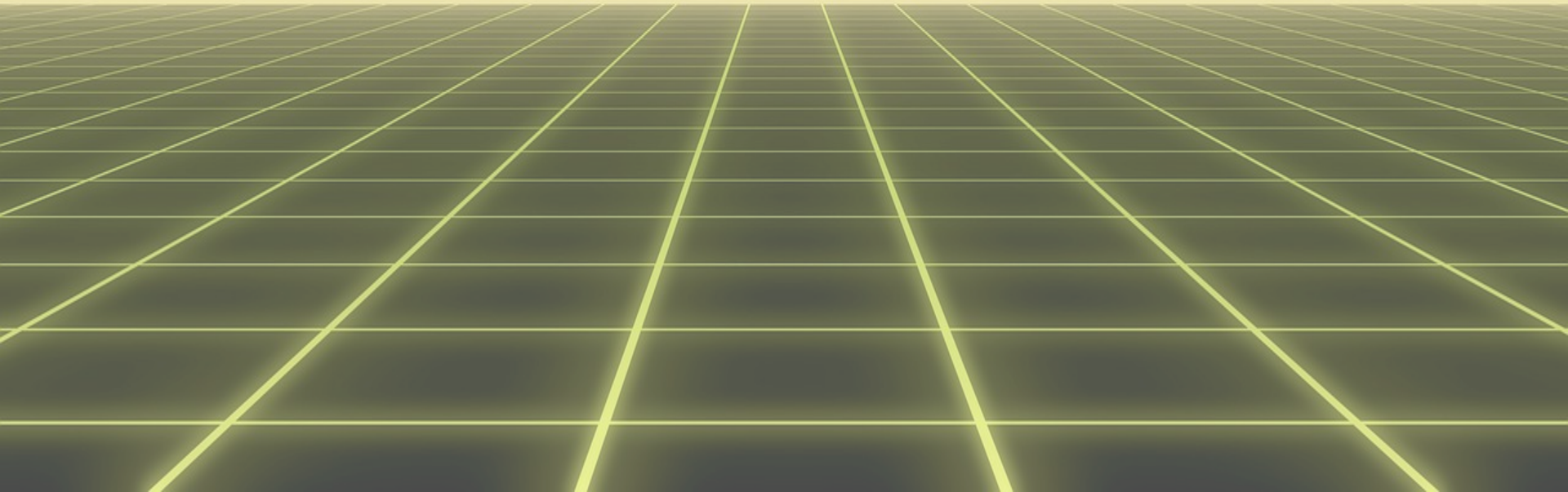


CSS 3 Positioning and Grid Layout



Overview

- **margin** properties
- **display** property
- **dimension** properties
- **position** properties
- **grid layout** (new CSS 3)

margin properties

- **margin** properties create space **around** an element
- 5 main margin properties
 - i. **margin-top**
 - ii. **margin-right**
 - iii. **margin-bottom**
 - iv. **margin-left**
 - v. **margin**

margin properties (cont.)

- **margin-top** property
 - ❖ creates space **above** of an element
 - suitable for creating space between elements

For example:

```
.p_class  
{  
  margin-top: 80px;  
}
```



Contact Us

Send us a message

Username:

Password:

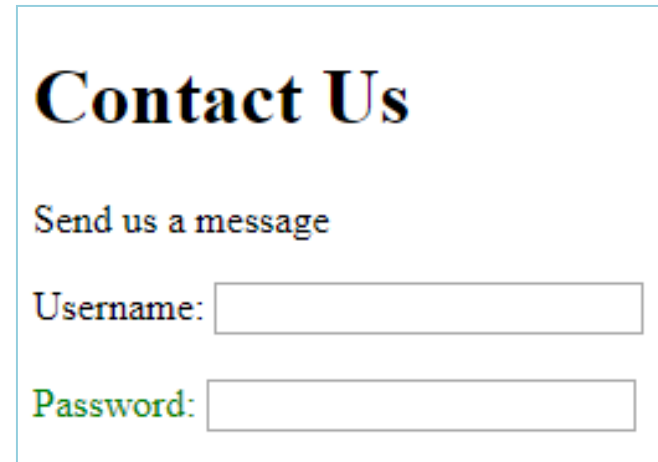
In this example: **margin-top** property will create a space of 80 pixels **to the top** of the **<p>** element

margin properties (cont.)

- **margin-right** property
 - ❖ creates space **on the right** of an element

For example: suitable for creating space between inline elements

```
.p_class  
{  
    margin-right: 80px;  
}
```



Contact Us

Send us a message

Username:

Password:

(Note: **margin-right** usually won't show a visible difference unless

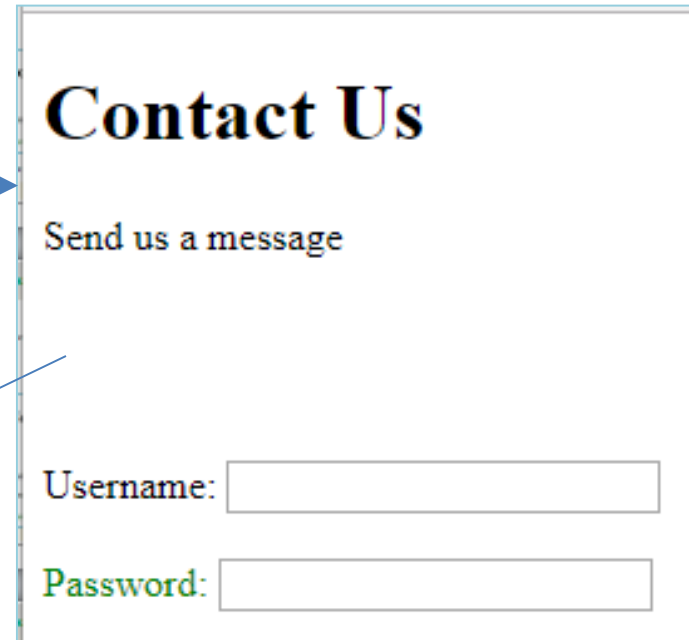
- other elements (inline elements) are on the same line
- the element is aligned to the right edge of the webpage (i.e. **text-align:right** ; property)

margin properties (cont.)

- **margin-bottom** property
 - ❖ creates space **below** an element

For example:

```
.p_class  
{  
    margin-bottom: 80px;  
}
```



In this example: **margin-bottom** property will create a space of 80 pixels **below** the <p> element

margin properties (cont.)

- **margin-left** property
 - ❖ creates space **on the left** side of an element

For example:

```
.p_class  
{  
    margin-left: 100px;  
}
```



Contact Us

Send us a message

Username:

Password:

In this example: **margin-left** property will create a space of 100 pixels **to the left** of the **<p>** element

margin properties (cont.)

- **margin** property
 - ❖ can be used to create space on the **top**, **right**, **bottom** and **left** of an element
 - ❖ margin: **top-value right-value bottom-value left-value**
margin property uses four values

For example:

```
.p_class  
{  
  margin: 20px 20px 50px 70px;  
}
```

top

right

bottom

left

Contact Us

Send us a message

Username:

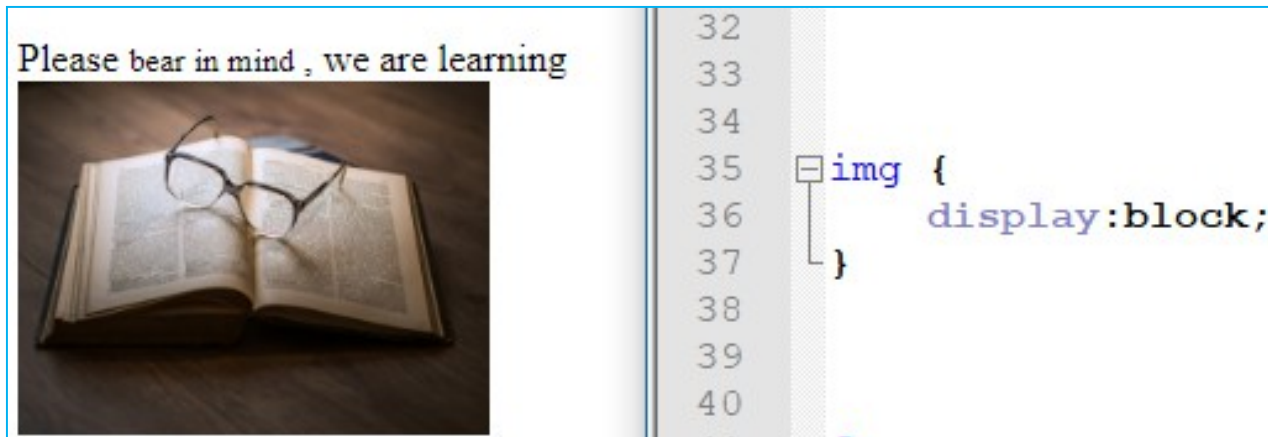
Password:

display property

- display property:

- ❖ specifies the display behavior of an element

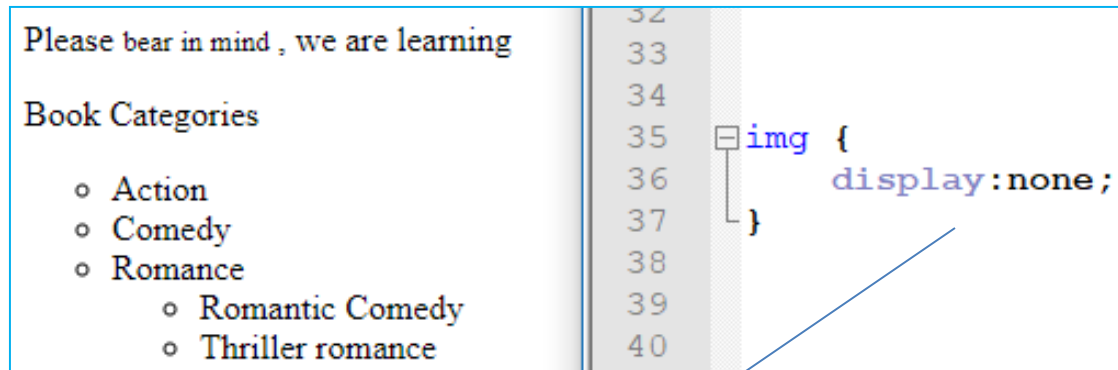
For example (an **inline** element can be **displayed like a block** element)



- By default, **img** is an inline element, (will display on the same line as the previous element),
- It can be displayed **like a block element**

display property (cont.)

- **display** property, main values:
 - ❖ **block**
 - ❖ **inline**
 - ❖ **none**
- **none** can be used to hide an element from visibility
 - (element can be later displayed using **JavaScript**)



The screenshot shows a browser's developer tools interface. On the left, the DOM tree is expanded to show a list of book categories: "Please bear in mind , we are learning", "Book Categories", "Action", "Comedy", "Romance", "Romantic Comedy", and "Thriller romance". On the right, the CSS pane shows a rule for an `img` element: `img { display:none; }`. A blue line points from the `img` selector in the CSS rule to the `img` element in the DOM tree.

For example: **img** will not be visible in the browser

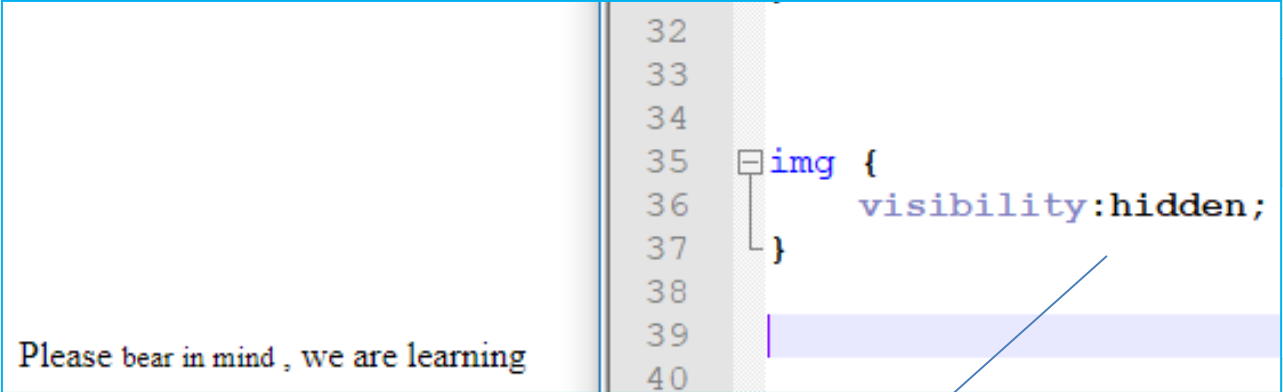
display property (cont.)

- **visibility** property

- ❖ specifies whether an element is visible or not

- ❖ **values:**

- visible
- hidden



```
32  
33  
34  
35 img {  
36     visibility: hidden;  
37 }  
38  
39  
40
```

Please bear in mind, we are learning

For example: **img** will be invisible in the browser

display property (cont.)

- **display** and **visibility** difference:

display: none; (element is removed and does not take any space on the web page)

visibility: hidden (element is **invisible** but will still take up space on the web page)

display

```
32 Please bear in mind , we are learning
33
34 Book Categories
35 ◦ Action
36 ◦ Comedy
37 ◦ Romance
38   ◦ Romantic Comedy
39   ◦ Thriller romance
40
```

```
35 img {
36   display: none;
37 }
```

visibility

```
32
33
34
35 ◦ Action
36 ◦ Comedy
37 ◦ Romance
38   ◦ Romantic Comedy
39   ◦ Thriller romance
40
```

```
35 img {
36   visibility: hidden;
37 }
```

Please bear in mind , we are learning

dimension elements

- 2 main **dimension** properties
 - ❖ width
 - ❖ height

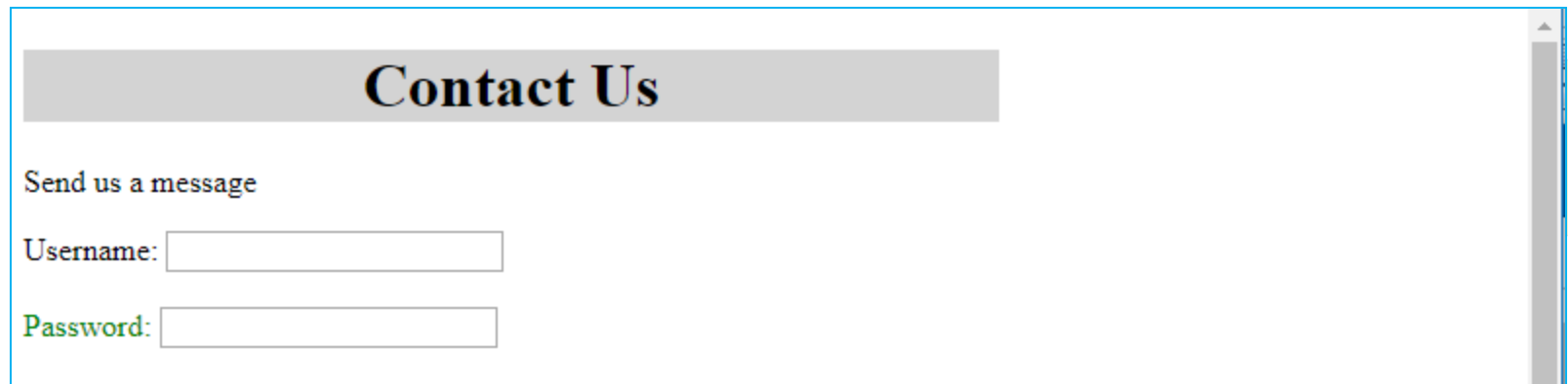
dimension elements

- **width** property
 - ❖ specifies the width of an element
- **height** property
 - ❖ specifies the height of an element

dimension elements (cont.)

- width property example

```
.h1_class  
{  
    width: 500px;  
    text-align: center;  
    background-color: lightgrey;  
}
```



In the example: **width** of the **h1** element will be 500px
text-align will center the align based on the 500px instead of the full browser width

position properties

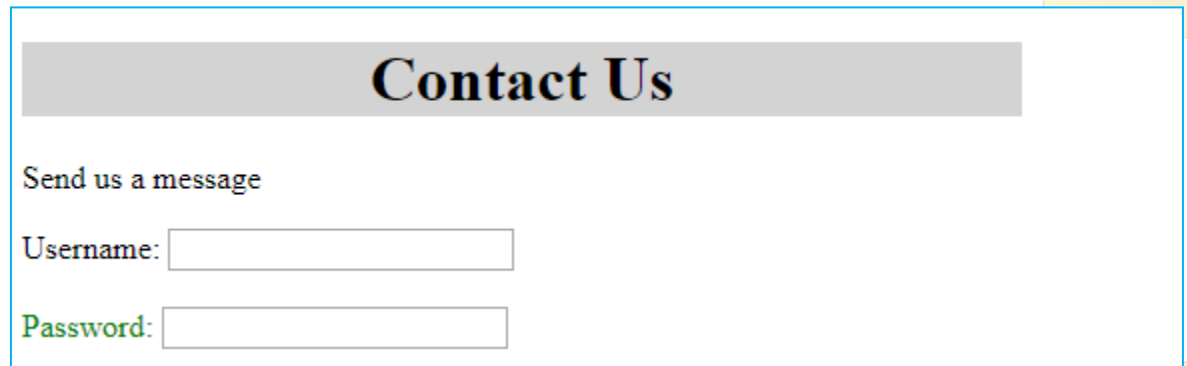
- **position** property
 - ❖ specifies how an element will be positioned on a web page
 - ❖ 4 main possible values
 - static**
 - relative**
 - absolute**
 - fixed**

position properties

- **static position**
 - ❖ static position is the default (does not require CSS code)
 - ❖ this is whereby the element follows the normal, top to bottom, order of the web page elements

```
.h1_class  
{  
    width: 500px;  
    text-align: center;  
    background-color: lightgrey;  
    position: static;  
}
```

- **position: static;** is the default
- **usually** does not need to be specified in CSS



Contact Us

Send us a message

Username:

Password:

position properties

- **relative position**

- ❖ element is positioned similar to its default (static) position
- ❖ However, the properties **left**, **right**, **top** and **bottom** can now be used

to specify the position of the element

```
.h1_class  
{  
    width: 500px;  
    text-align: center;  
    background-color: lightgrey;  
    position: relative;  
    left: 50px;  
}
```

- **left** property leaves space to the left of the element (similar to **margin-left**)

- **left**, **right**, **bottom**, **top** properties cannot be used with static positioning



Contact Us

Send us a message

Username:

Password:

position properties

- absolute position

- ❖ element is positioned relative to the nearest non-static element
- ❖ If no non-static element is found, the element will be positioned relative to the top-left corner of the browser

```
.p_class  
{  
    position: absolute;  
    top: 2px;  
}
```



Send us a message **Contact Us**

Username:

Password:

- ❖ Note: other elements will ignore the space which the element would normally take up with static positioning
 - can be used to create overlapping elements

position properties

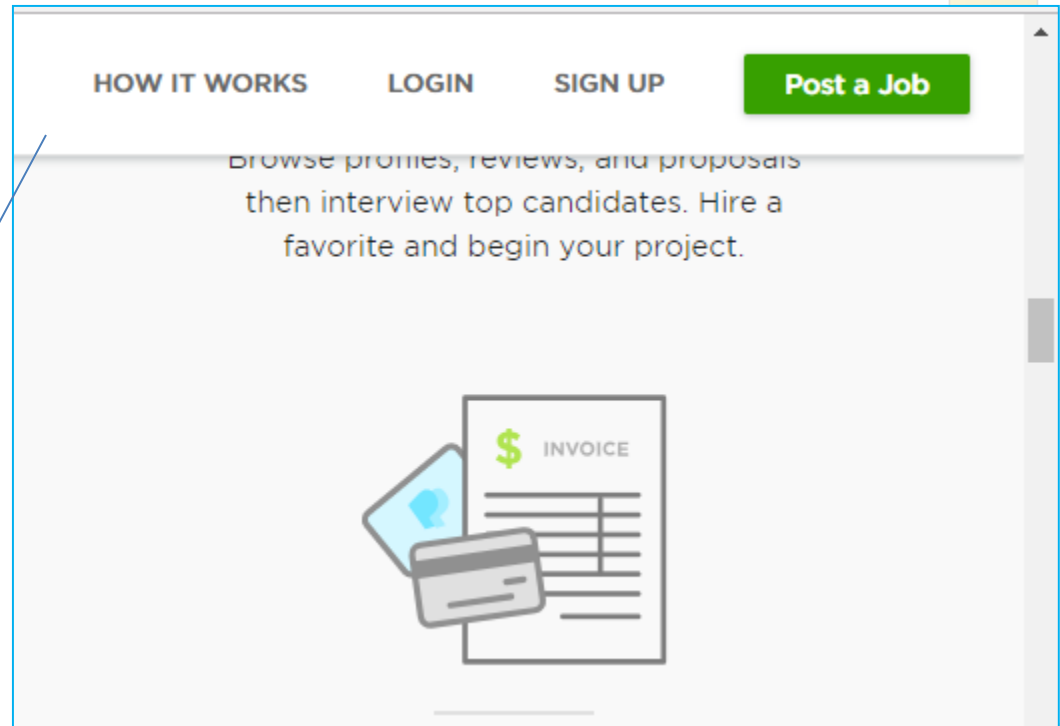
- fixed position

- ❖ element's position is relative to the browser's view

- this means that the element remains in the same position even if the user scrolls down the webpage

- can be used to display content which is always visible to the user

- (for example a **fixed** main menu bar)



grid layout

- grid-layout
 - ❖ provides **rows** and **columns** which the web designer can use to affect the layout of the webpage
 - ❖ **new in CSS 3**

grid layout

- grid-layout

- ❖ **display** property with **grid** keyword

- allows an element to **use a grid-layout**

```
section { display: grid;}
```

- ❖ **grid-column** property:

- used on the **nested** (inner) elements of the grid-layout element
- specifies the **column** to place an element within the grid-layout

grid layout

```
.two_columns{
  display:grid;
}

.list1 {
  grid-column:1;
}

.list2 {
  grid-column:2;
}
```

CSS file

```
<section class="two_columns">
  <section class="list1">
    <p> Book Categories </p>
    <ul >
      <li> Action </li>
      <li> Comedy </li>
      <li> Romance </li>
    </ul>
  </section>

  <section class="list2">
    <p> Best Sellers </p>
    <ol reversed>
      <li> The Revolution</li>
      <li> The Westerners </li>
      <li> ICT Webpages </li>
    </ol>
  </section>
</section>
```

HTML file

Book Categories

- o Action
- o Comedy
- o Romance

Best Sellers

3. The Revolution
2. The Westerners
1. ICT Webpages

Web browser: **section** element using **grid-layout**, now displays **2** columns for nested elements

grid layout (cont.)

- Note: there are various uses of grid-layout
 - ❖ most common use is putting elements into columns



grid layout (cont.)

❖ grid-template-columns property

- Used, on the grid-layout element, to adjust the width of the grid columns

```
.two_columns{  
  display:grid;  
  grid-template-columns: 25% 75%;  
}
```

