CSS Layout - The position Property

The `position` property specifies the type of positioning method used for an element (static, relative, fixed, absolute or sticky).

The position Property:

The `position` property specifies the type of positioning method used for an element.

There are five different position values:

- `static`
- `relative`
- `fixed`
- `absolute`
- `sticky`

Elements are then positioned using the top, bottom, left, and right properties. However, these properties will not work unless the `position` property is set first. They also work differently depending on the position value.

1/position: static;

HTML elements are positioned static by default.

Static positioned elements are not affected by the top, bottom, left, and right properties.

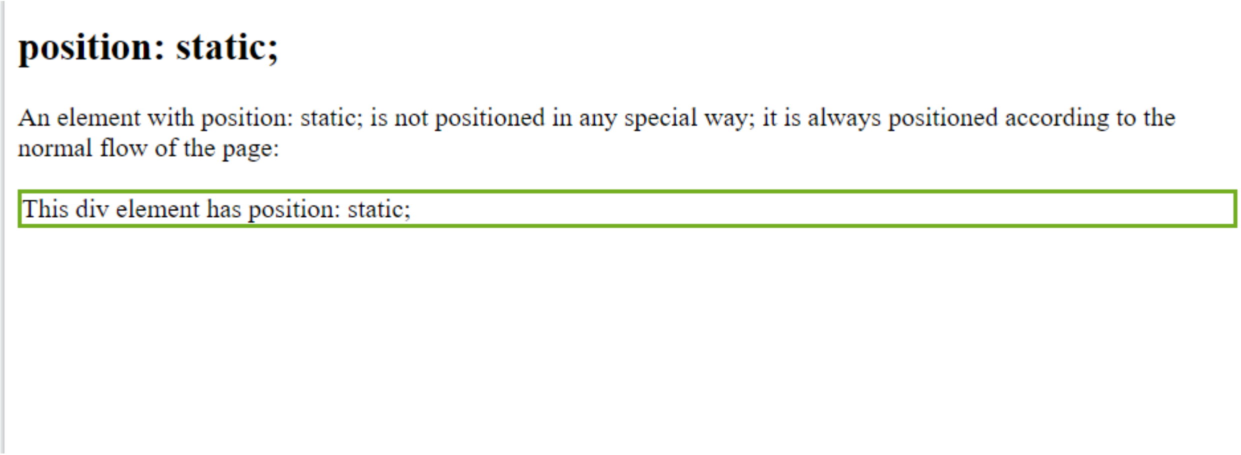
An element with `position: static;` is not positioned in any special way; it is always positioned according to the normal flow of the page:

This <div> element has position: static;

Here is the CSS that is used:

```
div.static {
  position: static;
  border: 3px solid #73AD21;
}
```

***Output:**



position: static;

An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page:

This div element has position: static;

2/position: relative;

An element with `position: relative;` is positioned relative to its normal position.

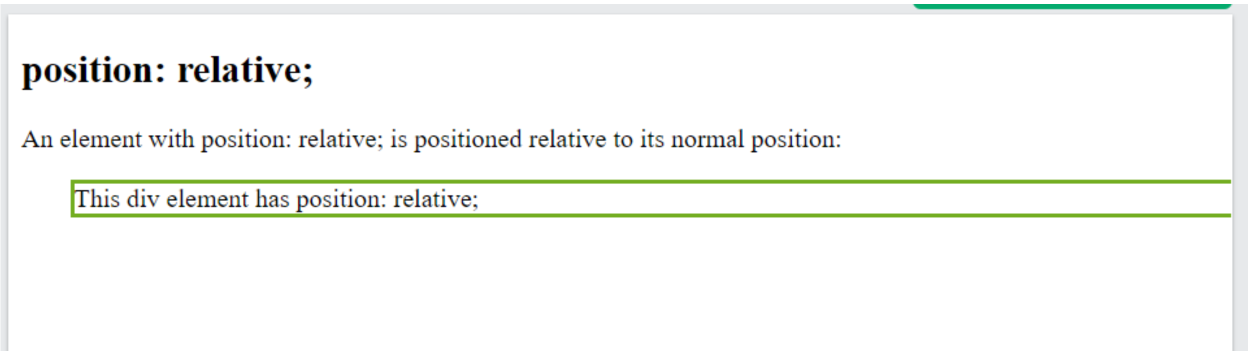
Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap left by the element.

This <div> element has position: relative;

Here is the CSS that is used:

```
div.relative {
  position: relative;
  left: 30px;
  border: 3px solid #73AD21;
}
```

***Output:**



position: relative;

An element with position: relative; is positioned relative to its normal position:

This div element has position: relative;

3/position: fixed;

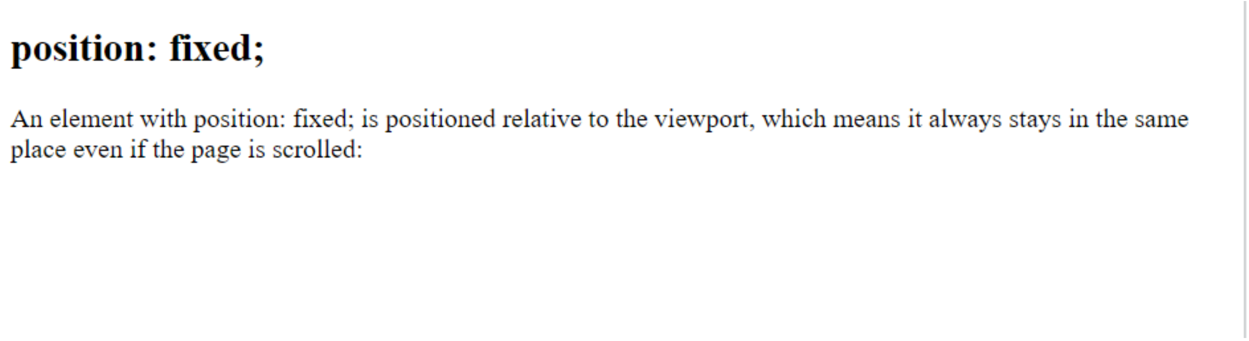
An element with `position: fixed;` is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element.

A fixed element does not leave a gap in the page where it would normally have been located.

Notice the fixed element in the lower-right corner of the page. Here is the CSS that is used:

```css
div.fixed {
  position: fixed;
  bottom: 0;
  right: 0;
  width: 300px;
  border: 3px solid #73AD21;
}
```

***Output:**



position: fixed;

An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled:

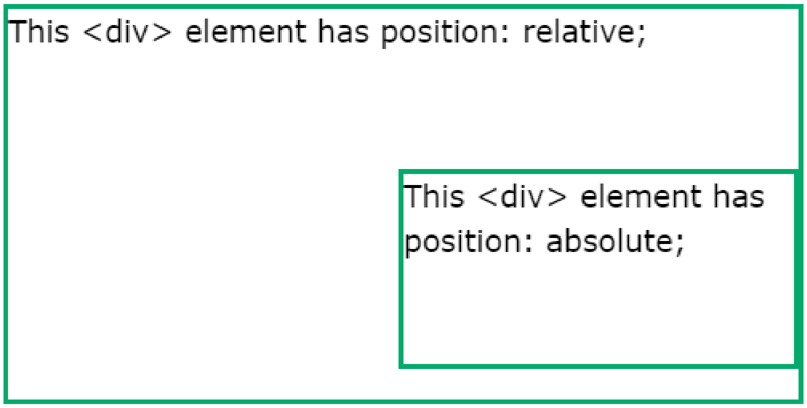
4/position: absolute;

An element with `position: absolute;` is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).

However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.

Note: Absolute positioned elements are removed from the normal flow, and can overlap elements.

Here is a simple example:



This <div> element has position: relative;

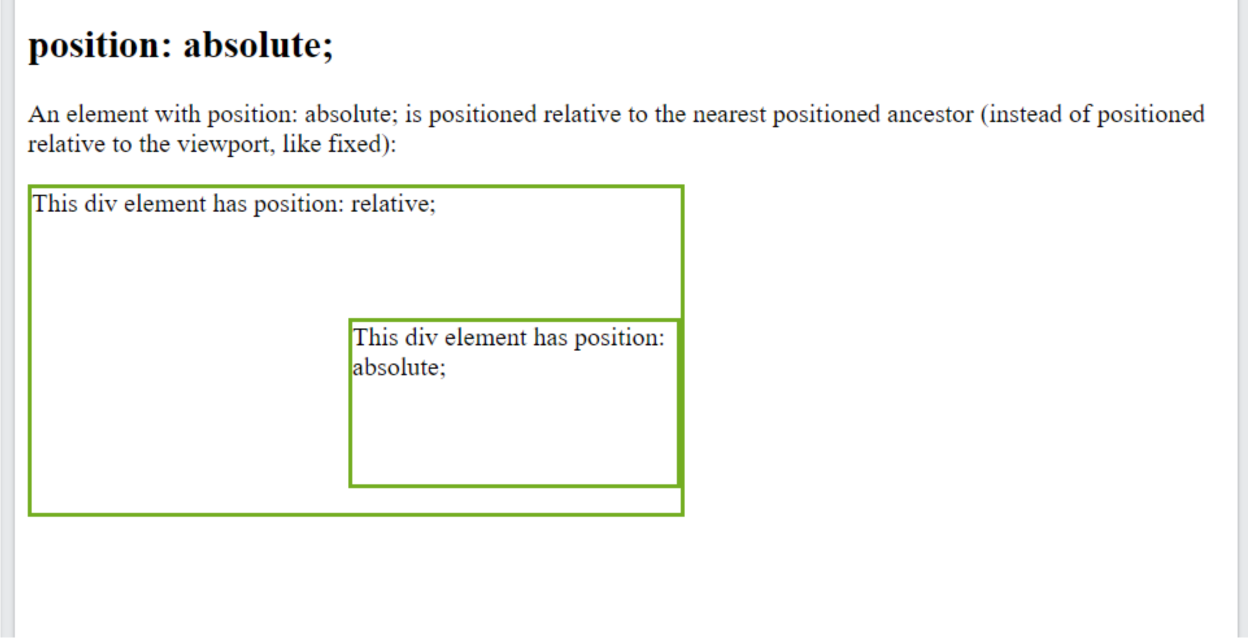
This <div> element has position: absolute;

Example

```
div.relative {
  position: relative;
  width: 400px;
  height: 200px;
  border: 3px solid #73AD21;
}

div.absolute {
  position: absolute;
  top: 80px;
  right: 0;
  width: 200px;
  height: 100px;
  border: 3px solid #73AD21;
}
```

***Output:**



position: absolute;

An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed):

This div element has position: relative;

This div element has position: absolute;

5/position: sticky;

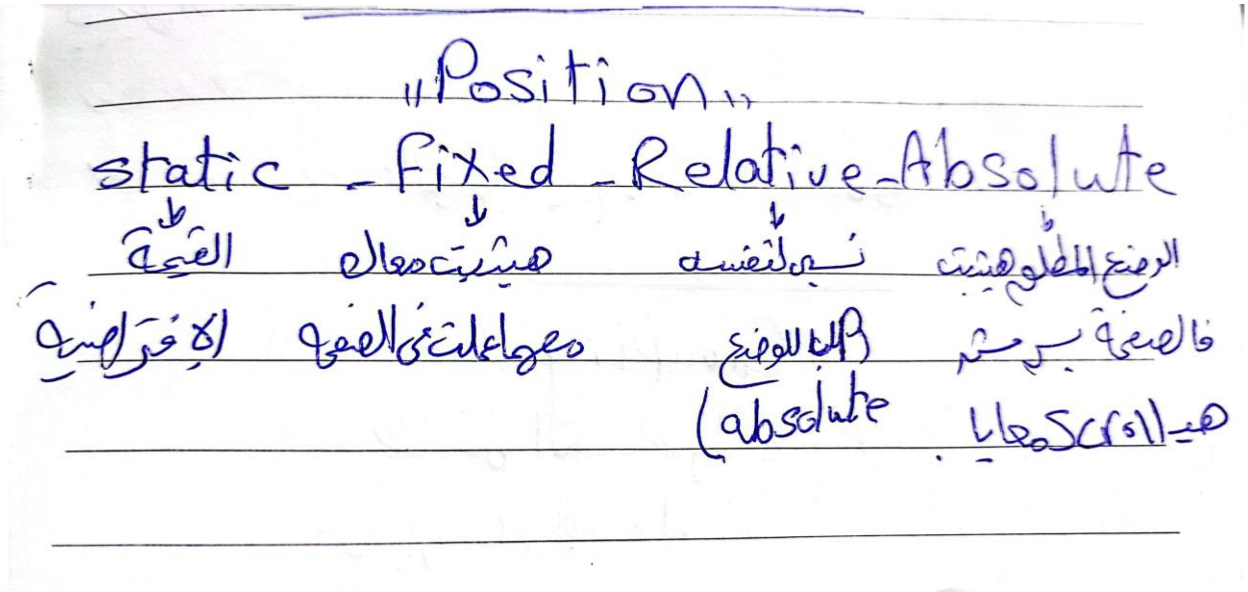
An element with `position: sticky;` is positioned based on the user's scroll position.

A sticky element toggles between `relative` and `fixed`, depending on the scroll position. It is positioned relative until a given offset position is met in the viewport - then it "sticks" in place (like position:fixed).

Note: Internet Explorer does not support sticky positioning. Safari requires a -webkit- prefix (see example below). You must also specify at least one of `top`, `right`, `bottom` or `left` for sticky positioning to work.

Arabic Explain:

1



"Position"

static - Fixed - Relative - Absolute

static: القيمة الافتراضية

Fixed: هيثبت معاك وهو ماشي على الصفحة

Relative: نسبي لنفسه (الأب للوضع absolute)

Absolute: الوضع المطلق هيثبت فالصفحة بس مش هيـ scroll معايا.

2

DATE / /   M T W T F S S ○ ○ ○ ○ ○ ○ ○

" Position "

① Static (default)　② Relative　③ Absolute　④ Fixed　⑤ Sticky

① ~~للعناصر الثابتة~~ reset اقدر أعمل بيه
لأنواع ال position التانية ودة بيخلي العنصر
في مكانه محجوز فال workflow زي ماهو.
ومش بيتحرك في أي حتة.

Position: Static;

② نسبي

العنصر بيتحرك بناءا على نفسه ومش بيأثر على
المحتوى اللي معاه

```
Position: relative;
top: ;  bottom: ;  ;
```

③ الوضع المطلق

العنصر بيطلع بره ال workflow كأنه مش موجود فالصفحة

```
position: absolute;
left
bottom
right
top
```

بيخلي العنصر يتوزع هنا في أعلى الصفحة

④ بيتحرك بناءً ع الصفحه بغض النظر هو فين
وكمان بيعمل معاه scroll
Position : fixed;

⑤ Position : sticky;
يتم وضعه بناءً اعلى موضع التمرير الخاص بالمستخدم
يعني بشتغل ويعمل Scroll لحد ما هو
يفضل ثابت مكانه عند نقطة معينة أنا احددها

←————————→

" Z_index "

. بيشتغل مع ال Position
. هو نظام الطبقات يعني أعطيه فوق ايه بناءً
على رقم معين كل ما الرقم كان أكبر كل ما ظهر
العنصر ده فوق باقي العناصر اللي واخده رقم أقل منه

Z-index : auto
قيمة افتراضية

Z_index: number
بدي له الرقم اللي انا عاوزاه