How to align checkboxes and their labels consistently cross-browsers

Asked 16 years, 1 month ago Modified 1 year ago Viewed 1.3m times

1897

This is one of the minor CSS problems that plague me constantly.

How do folks around Stack Overflow vertically align ***`checkboxes`*** and their ***`labels`*** consistently **cross-browser**?

Whenever I align them correctly in Safari (usually using `vertical-align: baseline` on the `input`), they're completely off in Firefox and IE.

Fix it in Firefox, and Safari and IE are inevitably messed up. I waste time on this every time I code a form.

Here's the standard code that I work with:

```
<form>
    <div>
        <label><input type="checkbox" /> Label text</label>
    </div>
</form>
```

▶ Run code snippet Expand snippet

I usually use Eric Meyer's reset, so form elements are relatively clean of overrides. Looking forward to any tips or tricks that you have to offer!

html css cross-browser alignment forms

Share
Improve this question
Follow

edited Apr 1, 2021 at 9:14

community wiki
11 revs, 10 users 22%
One Crayon

9 Put each checkbox and label within an <li> element. Add overflow:hidden to the <li> and float the label and checkbox left. Then they all align perfectly fine. Don't put the checkbox within the label element obviously. – volume one Jan 18, 2014 at 18:37

6 I have acieved it by using `height` and `line-height` attributes, give a look to jsfiddle.net/wepw5o57/3 – TheGr8_Nik Dec 9, 2014 at 13:41

Manipulation with `position` and `top` will solve this problem Example: jsfiddle.net/ynkjc22s – Profesor08 Apr 9, 2016 at 12:59

2019. still the same issue. still need some hacks to get it work :(– devops Apr 6, 2019 at 8:40

@dieter see my answer, I've explained why hacks are needed and what approach is not hacky: stackoverflow.com/a/56558431/3995261 – YakovL Jun 12, 2019 at 10:31

40 Answers

Sorted by: Highest score (default)

1 2 Next

▲

1039

▼

✓

Warning! This answer is *too old* and *doesn't work* on modern browsers.

I'm not the poster of this answer, but at the time of writing this, this is the most voted answer by far in both positive and negative votes (+1035 -17), and it's still marked as accepted answer (probably because the original poster of the question is the one who wrote this answer).

As already noted many times in the comments, this answer does not work on most browsers anymore (and seems to be failing to do that since 2013).

After over an hour of tweaking, testing, and trying different styles of markup, I think I may have a decent solution. The requirements for this particular project were:

1. Inputs must be on their own line.
2. Checkbox inputs need to align vertically with the label text similarly (if not identically) across all browsers.
3. If the label text wraps, it needs to be indented (so no wrapping down underneath the checkbox).

Before I get into any explanation, I'll just give you the code:

```
label {
  display: block;
  padding-left: 15px;
  text-indent: -15px;
}
input {
  width: 13px;
  height: 13px;
  padding: 0;
  margin:0;
  vertical-align: bottom;
  position: relative;
  top: -1px;
  *overflow: hidden;
}
```

```
<form>
  <div>
    <label><input type="checkbox" /> Label text</label>
  </div>
</form>
```

▶ Run code snippet   Expand snippet

Here is the working example in JSFiddle.

This code assumes that you're using a reset like Eric Meyer's that doesn't override form input margins and padding (hence putting margin and padding resets in the input CSS). Obviously in a live environment you'll probably be nesting/overriding stuff to support other input elements, but I wanted to keep things simple.

Things to note:

- The `*overflow` declaration is an inline IE hack (the star-property hack). Both IE 6 and 7 will notice it, but Safari and Firefox will properly ignore it. I think it might be valid CSS, but you're still better off with conditional comments; just used it for simplicity.
- As best I can tell, the only `vertical-align` statement that was consistent across browsers was `vertical-align: bottom`. Setting this and then relatively positioning upwards behaved almost identically in Safari, Firefox and IE with only a pixel or two of discrepancy.
- The major problem in working with alignment is that IE sticks a bunch of mysterious space around input elements. It isn't padding or margin, and it's damned persistent. Setting a width and height on the checkbox and then `overflow: hidden` for some reason cuts off the extra space and allows IE's positioning to act very similarly to Safari and Firefox.
- Depending on your text sizing, you'll no doubt need to adjust the relative positioning, width, height, and so forth to get things looking right.

I haven't tried this specific technique on any projects other than the one I was working on this morning, so definitely pipe up if you find something that works more consistently.

Warning! This answer is *too old* and *doesn't work* on modern browsers.

Share Improve this answer Follow

edited Dec 22, 2022 at 10:34

community wiki
8 revs, 8 users 60%
One Crayon

293

Sometimes vertical-align needs two inline (span, label, input, etc...) elements next to each other to work properly. The following checkboxes are properly vertically centered in IE, Safari, FF, and Chrome, even if the text size is very small or large.

They all float next to each other on the same line, but the nowrap means that the whole label text always stays next to the checkbox.

The downside is the extra meaningless SPAN tags.

```
.checkboxes label {
  display: inline-block;
  padding-right: 10px;
  white-space: nowrap;
}
.checkboxes input {
  vertical-align: middle;
}
.checkboxes label span {
  vertical-align: middle;
}
```

```
<form>
  <div class="checkboxes">
    <label><input type="checkbox"> <span>Label text x</span></label>
    <label><input type="checkbox"> <span>Label text y</span></label>
    <label><input type="checkbox"> <span>Label text z</span></label>
  </div>
</form>
```

▶ Run code snippet Expand snippet

Now, if you had a very long label text that *needed* to wrap without wrapping under the checkbox, you'd use padding and negative text indent on the label elements:

```
.checkboxes label {
  display: block;
  padding-right: 10px;
  padding-left: 22px;
  text-indent: -22px;
}
.checkboxes input {
  vertical-align: middle;
}
.checkboxes label span {
  vertical-align: middle;
}
```

```
<form>
  <div class="checkboxes">
    <label><input type="checkbox"> <span>Label text x so long that it will probably wrap so let's see how it goes with the proposed CSS (expected: two lines are aligned nicely)</span></label>
    <label><input type="checkbox"> <span>Label text y</span></label>
    <label><input type="checkbox"> <span>Label text z</span></label>
  </div>
</form>
```

▶ Run code snippet   Expand snippet

Share Improve this answer Follow

edited May 10, 2021 at 11:52

community wiki
6 revs, 6 users 56%
Nathan Bowers

27 Thanks! The "vertical-align: middle" on both the input and the span worked great for me. – William Gross Aug 12, 2010 at 14:42

208

Working off of One Crayon's solution, I have something that works for me and is simpler:

```
.font2 {font-family:Arial; font-size:32px} /* Sample font */

input[type=checkbox], input[type=radio] {
  vertical-align: middle;
  position: relative;
  bottom: 1px;
}

input[type=radio] {
  bottom: 2px;
}
```

```
<label><input type="checkbox" /> Label text</label>
<p class="font2">
  <label><input type="checkbox"/> Label text</label>
</p>
```

▶ Run code snippet   Expand snippet

Renders pixel-for-pixel the same in Safari (whose baseline I trust) and both Firefox and IE7 check out as good. It also works for various label font sizes, big and small. Now, for fixing IE's baseline on selects and inputs...

Update: (Third-Party Edit)

The proper `bottom` position depends on font-family and font-size! I found using **`bottom: .08em;`** for checkbox & radio elements is a good general value. I tested it in Chrome/Firefox/IE11 in windows with Arial & Calibri fonts using several small/mid/large font-sizes.

```
.font2, .font2 input {font-family:Arial; font-size:32px} /* Sample font */

input[type=checkbox], input[type=radio] {
  vertical-align: middle;
  position: relative;
  bottom: .08em; /* this is a better value for different fonts! */
}
```

```
<label><input type="checkbox" /> Label text</label>

<p class="font2">
  <label><input type="checkbox"/> Label text</label>
</p>
```

▶ Run code snippet   Expand snippet

Share Improve this answer Follow

edited Jan 30, 2019 at 14:33

community wiki
6 revs, 5 users 47%
S.Serpooshan

11 Just a note for others: this won't work in IE6 because it doesn't support the [type=checkbox] CSS targeting. – One Crayon Apr 8, 2009 at 0:18

165

One easy thing that seems to work well is to apply a adjust the vertical position of the checkbox with vertical-align. It will still be vary across browsers, but the solution is uncomplicated.

```
input {
    vertical-align: -2px;
}
```

Reference

Share Improve this answer Follow

edited Jul 30, 2015 at 10:09

community wiki
3 revs, 3 users 47%
Frank Schwieterman

darn simple. the other middle aligns did not work for me – JesseBoyd May 26, 2023 at 19:11

101

try `vertical-align: middle`

also your code seems like it should be:

```
<form>
    <div>
        <input id="blah" type="checkbox"><label for="blah">Label text</label>
    </div>
</form>
```

▶ Run code snippet   Expand snippet

Share Improve this answer Follow

edited Nov 12, 2017 at 18:09

community wiki
4 revs, 4 users 71%
digitalsanctum

15 Not exactly true: The Label-for will allow users to click the label in order to check the checkbox, in addition to simply clicking the checkbox itself. It's quite handy for tying the two elements together. – EndangeredMassa Nov 20, 2008 at 18:30

33 If an input is nested inside a label, then clicking the label with activate/give focus to the input; the for attribute is solely for the case when the input is not nested. – One Crayon Nov 20, 2008 at 18:34

42

Try my solution, I tried it in IE 6, FF2 and Chrome and it renders pixel by pixel in all the three browsers.

```
* {
  padding: 0px;
  margin: 0px;
}
#wb {
  width: 15px;
  height: 15px;
  float: left;
}
#somelabel {
  float: left;
  padding-left: 3px;
}
```

```
<div>
  <input id="wb" type="checkbox" />
  <label for="wb" id="somelabel">Web Browser</label>
</div>
```

▶ Run code snippet   Expand snippet

Share Improve this answer Follow

edited Aug 6, 2016 at 2:33

community wiki
5 revs, 4 users 56%
Waleed Eissa

34

The only perfectly working solution for me is:

```
input[type=checkbox], input[type=radio] {
    vertical-align: -2px;
    margin: 0;
    padding: 0;
}
```

Tested today in Chrome, Firefox, Opera, IE 7 and 8. Example: [Fiddle]

Share Improve this answer Follow

edited Dec 1, 2023 at 18:10

community wiki
4 revs, 4 users 51%
Buzogany Laszlo

28

I have not completely tested my solution, but it seems to work great.

My HTML is simply:

```
<label class="checkbox"><input type="checkbox" value="0000">0000 - 0100</label>
```

I then set all checkboxes to `24px` for both height and width. To make the text aligned I make the label's `line-height` also `24px` and assign `vertical-align: top;` like so:

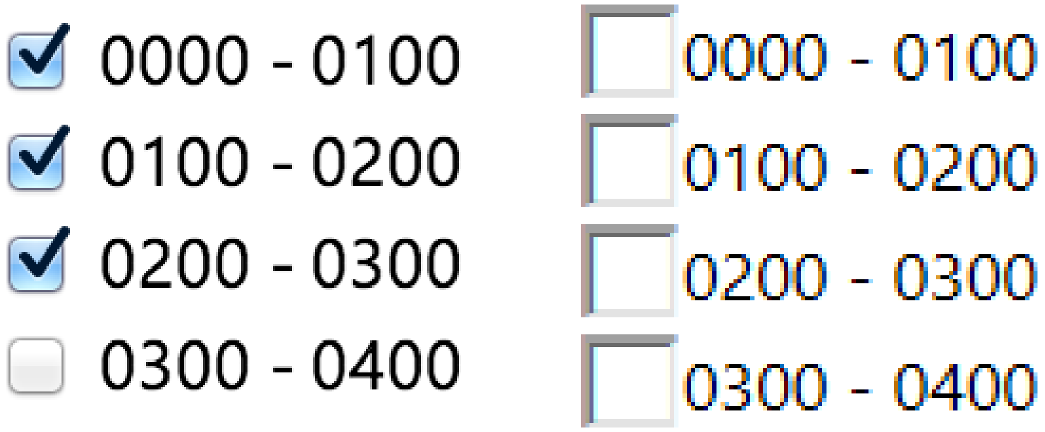
EDIT: After IE testing I added `vertical-align: bottom;` to the input and changed the label's CSS. You may find you need a conditional IE css case to sort out padding - but the text and box are inline.

```
input[type="checkbox"] {
    width: 24px;
    height: 24px;
    vertical-align: bottom;
}
label.checkbox {
    vertical-align: top;
    line-height: 24px;
    margin: 2px 0;
    display: block;
    height: 24px;
}
```

```
<label class="checkbox"><input type="checkbox" value="0000">0000 - 0100</label>
<label class="checkbox"><input type="checkbox" value="0100">0100 - 0200</label>
<label class="checkbox"><input type="checkbox" value="0200">0200 - 0300</label>
<label class="checkbox"><input type="checkbox" value="0300">0300 - 0400</label>
```

▶ Run code snippet   Expand snippet

If anyone finds that this doesn't work, please kindly let me know. Here is it in action (in Chrome and IE - apologies as screenshots were taken on retina and using parallels for IE):



☑ 0000 - 0100
☑ 0100 - 0200
☑ 0200 - 0300
□ 0300 - 0400

□ 0000 - 0100
□ 0100 - 0200
□ 0200 - 0300
□ 0300 - 0400

Share Improve this answer Follow

edited Sep 26, 2019 at 16:03

community wiki
4 revs, 3 users 78%
Patrick

Hard px values = :(– Kalnode Jan 27, 2023 at 19:50

23

I usually use line height in order to adjust the vertical position of my static text:

```
label {
  line-height: 18px;
}
input {
  width: 13px;
```

```
    height: 18px;
    font-size: 12px;
    line-height: 12px;
}
```

```
<form>
  <div>
    <label><input type="checkbox" /> Label text</label>
  </div>
</form>
```

▶ Run code snippet   Expand snippet

Hope that helps.

Share Improve this answer Follow

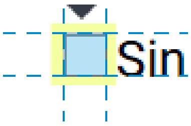
edited Aug 6, 2016 at 2:38

community wiki
2 revs, 2 users 69%
Plan B

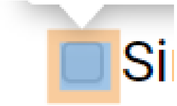
21

Let's finally take a look at the source of the problem

The checkboxes are rendered using images (one may set custom ones via CSS). Here is an (unchecked) checkbox in FireFox, highlighted with DOM inspector:



And here's the same unstyled checkbox in Chrome:



You can see the margin (orange); padding is not present (would be shown green). So what's this pseudo-margin on the right and on the bottom of the checkbox? **These are parts of the image used for the checkbox**. That's why using just `vertical-align: middle` doesn't really suffice and that's the source of the cross-browser problems.

So what can we do about this?

One obvious option is – replace the images! Fortunately, one can do this via CSS and replace those with external images, base64 (in-CSS) images, in-CSS svg or just pseudo-elements. It's a robust (cross-browser!) approach, and here's an example of such adjustment stolen from this question:

```
.checkbox-custom {
  opacity: 0;
  position: absolute;
}
.checkbox-custom,
.checkbox-custom-label {
  display: inline-block;
  vertical-align: middle;
  margin: 5px;
  cursor: pointer;
}
.checkbox-custom + .checkbox-custom-label:before {
  content: '';
  display: inline-block;
  background: #fff;
  border-radius: 5px;
  border: 2px solid #ddd;
  vertical-align: middle;
  width: 10px;
  height: 10px;
  padding: 2px;
  margin-right: 10px;
  text-align: center;
}
.checkbox-custom:checked + .checkbox-custom-label:before {
  width: 1px;
  height: 5px;
  border: solid blue;
  border-width: 0 3px 3px 0;
  transform: rotate(45deg);
  -webkit-transform: rotate(45deg);
  -ms-transform: rotate(45deg);
  border-radius: 0px;
  margin: 0px 15px 5px 5px;
}
```

```
<div>
  <input id="checkbox-1" class="checkbox-custom" name="checkbox-1" type="checkbox">
  <label for="checkbox-1" class="checkbox-custom-label">First Choice</label>
</div>
<div>
  <input id="checkbox-2" class="checkbox-custom" name="checkbox-2" type="checkbox">
  <label for="checkbox-2" class="checkbox-custom-label">Second Choice</label>
</div>
```

▶ Run code snippet   Expand snippet

You may want to read some more in-depth articles about such styling like some listed here; it's out of scope of this answer.

Ok, still what about no-custom-images-or-pseudo-elements solution?

TL;DR: looks like this won't work, use custom checkbox instead

First, let's notice that if in other browsers those pseudo-margins inside checkbox icon were arbitrary, there were no consistent solution. To build one, we have to explore the anatomy of such images in existing browsers.

So what browsers do have the pseudo-margins in checkboxes? I've checked out Chrome 75, Vivaldi 2.5 (Chromium-based), FireFox 54 (don't ask why such outdated), IE 11, Edge 42, Safari ?? (borrowed one for a minute, forgot to check out the version). Only Chrome and Vivaldi has such pseudo-margins (I suspect all Chromium-based browsers as well, like Opera).

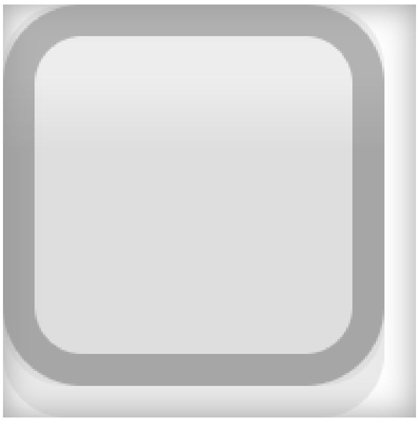
What's the size of those pseudo-margins? To figure this out one can use a zoomed checkbox:

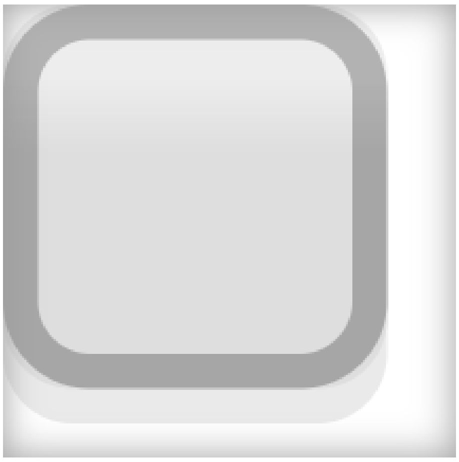
```
input {
  zoom: 10;
  box-shadow: 0 0 1px inset #999;
}
```

```
<input type=checkbox>
```

▶ Run code snippet   Expand snippet

my result is ~7% of width/height and hence 0.9-1.0px in absolute units. The accuracy may be questioned, though: try different values of `zoom` for the checkbox. In my tests in both Chrome and Vivaldi the relative size of the pseudo-margin is very different at `zoom` values 10, 20 and at values 11-19 (??):





`scale` seems to be more consistent:

```
input {
  transform: scale(10) translate(50%, 50%);
  box-shadow: 0 0 1px inset #999;
}
```

```
<input type=checkbox>
```

▶ Run code snippet   Expand snippet

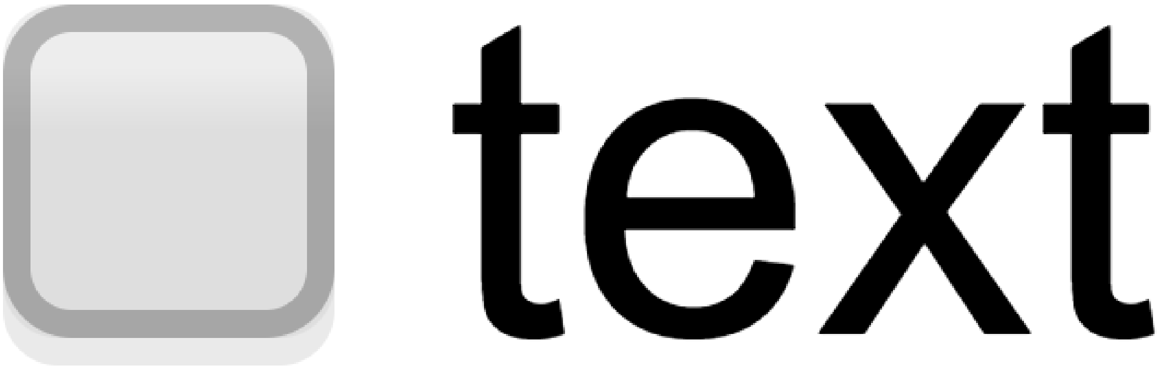
so probably ~14% and 2px are the correct values.

Now that we know (?) the size of the pseudo-margin, let's note this is not enough. Are the sizes of the checkbox icons the same for all browsers? Alas! Here's what DOM inspector shows for unstyled checkboxes:

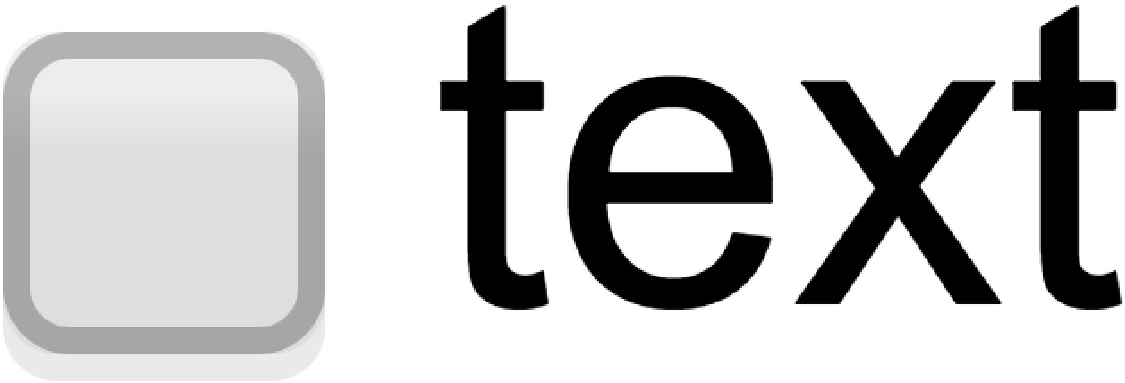
- FireFox: **13.3px**
- Chromium-based: **12.8px** for the whole thing, hence 12.8 (100% - 14%) = **11px** for what is visually perceived as checkbox
- IE 11, Edge: **13px**
- Safari: n/a (these should be compared on the same screen, I believe)

Now before we discuss any solutions or tricks, let's ask: what is a *correct* alignment? What are we trying to achieve? To certain point it's a matter of taste, but basically I can think of the following "nice" alignments' aspects:

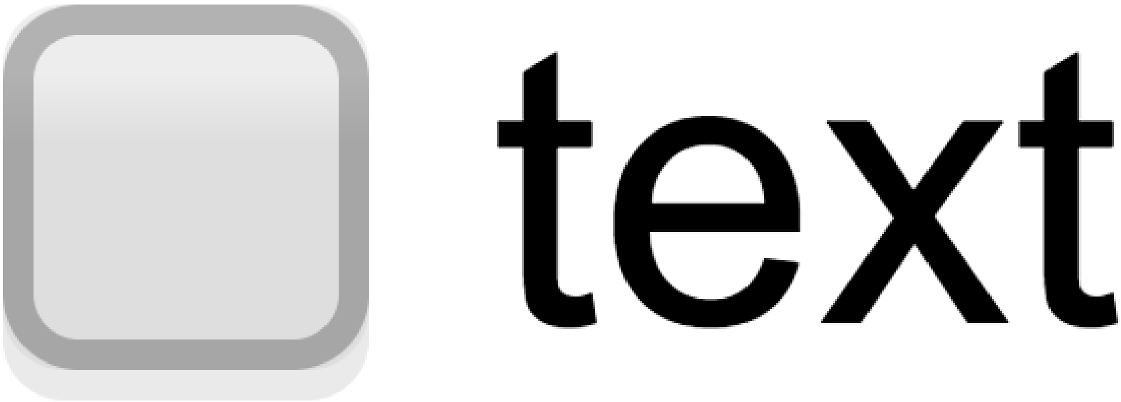
text and checkbox on the same baseline (I deliberately don't adjust checkbox size here):



or have same middle line in terms of lowercase letters:

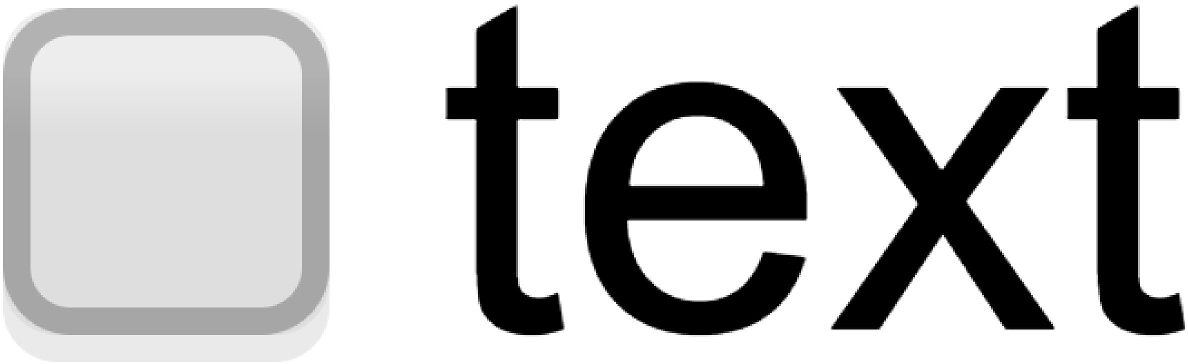


or same middle line in terms of capital letters (it's easier to see the difference for different font size):



and also we have to decide whether the size of the checkbox should be equal to the height of a lowercase letter, a capital letter or something else (bigger, smaller or between lowercase and capital).

For this discussion let's call an alignment nice if the checkbox is on the same baseline as the text and has the size of a capital letter (a highly arguable choice):



Now what tools do we have to:

1. adjust checkbox size
2. recognize Chromium with its pseudo-margined checkbox and set specific styles

3. adjust checkbox/label vertical alignment

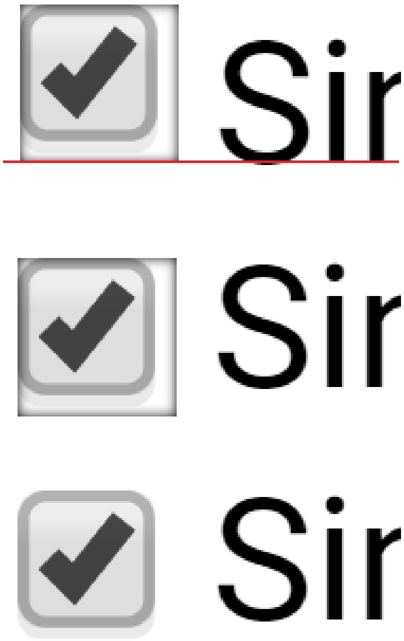
?

1. Regarding the checkbox size adjustment: there are `width`, `height`, `size`, `zoom`, `scale` (have I missed something?). `zoom` and `scale` don't allow to set absolute size, so they may help only with adjusting to text size, not set cross-browser size (unless we can write browser-specific rules). `size` doesn't work with Chrome (did it work with old IE? anyway, it's not that interesting). `width` and `height` work in Chrome and other browsers, so we can set a common size, but again, in Chrome it sets the size of the whole image, not the checkbox itself. Note: it is minimum(width, height) which defines a checkbox's size (if width ≠ height, the area outside checkbox square is added to "pseudo-margin").

   An unfortunate thing is, the pseudo-margins in Chrome checkbox are not set to zero for any width and heights, as far as I can see.

2. I'm afraid there's no reliable CSS-only method these days.

3. Let's consider vertical alignment. `vertical-align` can't give consistent results when set to `middle` or `baseline` because of the Chrome's pseudo-margin, the only real option to get the same "coordinate system" for all the browsers is to align label and input to the `top`:

   (on the picture: vertical-align: top, bottom and bottom without box-shadow)

So what result do we get from this?

```
input[type="checkbox"] {
    height: 0.95em;
    width: 0.95em;
}
```

```css
label, input {
    vertical-align: top;
}
```

```html
<label><input type="checkbox">label</label>
```

▶ Run code snippet   Expand snippet

The snippet above works with Chrome (Chromium-based browsers), but other browsers require a smaller size of the checkbox. It seems to be impossible to adjust both the size and vertical alignment in a way that works around Chromium's checkbox image quirk. **My final suggestion is: use custom checkboxes instead – and you'll avoid frustration :)**

Share Improve this answer Follow

edited Jun 16, 2019 at 12:03

community wiki
3 revs
YakovL

16

Now that flexbox is supported in all modern browsers, something like this seems like an easier approach to me.

```
<style>
label {
  display: flex;
  align-items: center;
}
input[type=radio], input[type=checkbox]{
  flex: none;
}
</style>
<form>
  <div>
    <label><input type="checkbox" /> Label text</label>
  </div>
</form>
```

Here's the complete prefixed version demo:

▸ Show code snippet

Share Improve this answer Follow

answered Jul 24, 2016 at 23:53

community wiki
Bryan Willis

5 Why the need for flex: none? – Adrian Lynch Jun 17, 2021 at 14:17

16

```
<form>
    <div>
        <label style="display: inline-block">
            <input style="vertical-align: middle" type="checkbox" />
            <span style="vertical-align: middle">Label text</span>
         </label>
    </div>
</form>
```

▶ Run code snippet   Expand snippet

The trick is to use vertical-align only in table cells or inline-block if using label tag.

Share Improve this answer Follow   edited Jun 28, 2019 at 10:46   community wiki
2 revs, 2 users 64%
Carlo Pires

14

I think this is the easiest way

```
input {
    position: relative;
    top: 1px;
}
```

```
<form>
    <div>
        <label><input type="checkbox" /> Label text</label>
    </div>
<form>
```

▶ Run code snippet   Expand snippet

Share Improve this answer Follow   edited Jun 28, 2019 at 10:44   community wiki
4 revs, 4 users 44%
gidzior

1 the question is: does it work well in all browsers? ;-) – JonSnow Apr 29, 2015 at 13:58

I don't like relative positioning because it makes element rendered above everything else on its level (it can get on top of something if you have complex layout).

11

I've discovered that `vertical-align: sub` makes checkboxes look good enough aligned in Chrome, Firefox and Opera. Can't check Safari since I don't have MacOS and IE10 is slightly off, but I've found it to be good enough solution for me.

Another solution might be to try and make specific CSS for every browser and fine-tune it with some vertical-align in %/pixels/EMs: http://css-tricks.com/snippets/css/browser-specific-hacks/

Share Improve this answer Follow

answered Nov 6, 2014 at 13:59

community wiki
waterplea

Seems to work as long as the font size is not too large. – robsch Sep 27, 2018 at 10:58

11

This works well for me:

```
fieldset {
  text-align:left;
  border:none
}
fieldset ol, fieldset ul {
  padding:0;
  list-style:none
}
fieldset li {
  padding-bottom:1.5em;
  float:none;
  clear:left
}
label {
  float:left;
  width:7em;
  margin-right:1em
}
fieldset.checkboxes li {
  clear:both;
  padding:.75em
}
fieldset.checkboxes label {
  margin:0 0 0 1em;
  width:20em
}
fieldset.checkboxes input {
  float:left
}
```

```
<form>
  <fieldset class="checkboxes">
    <ul>
      <li>
        <input type="checkbox" name="happy" value="yep" id="happy" />
        <label for="happy">Happy?</label>
        </li>
        <li>
          <input type="checkbox" name="hungry" value="yep" id="hungry" />
          <label for="hungry">Hungry?</label>
        </li>
      </ul>
    </fieldset>
</form>
```

▶ Run code snippet   Expand snippet

Share Improve this answer Follow

edited Aug 6, 2016 at 2:40

community wiki
3 revs, 3 users 40%
dylanfm

10 bless you for using ems instead of pxs. – IDisposable Jan 31, 2009 at 6:45

10

The chosen answer with 400+ upvotes did not work for me in Chrome 28 OSX, probably because it wasn't tested in OSX or that it did work in whatever was around in 2008 when this question was answered.

The times have changed, and new CSS3 solutions are now feasible. My solution uses pseudoelements to create a *custom checkbox*. So the stipulations (pros or cons, however you look at it) are as follows:

- Only works in modern browsers (FF3.6+, IE9+, Chrome, Safari)
- Relies on a custom designed checkbox, that will be rendered exactly the same in every browser/OS. Here I've just chosen some simple colors, but you could always add linear gradients and such to give it more of a bang.
- Is geared to a certain font/font size, which if changed, you'd simply change the positioning and size of the checkbox to make it appear vertically aligned. If tweaked correctly, the end result should still be near to exactly the same in all browser / operating systems.
- No vertical-alignment properties, no floats
- Must use the provided markup in my example, it will not work if structured like the question, however, the layout will essentially look the same. If you want to move things around, you'll have to also move the associated CSS

```
div.checkbox {
    position: relative;
    font-family: Arial;
    font-size: 13px;
}
label {
    position: relative;
    padding-left: 16px;
}
label::before {
    content :"";
    display: inline-block;
    width: 10px;
    height: 10px;
    background-color: white;
    border: solid 1px #9C9C9C;
    position: absolute;
    top: 1px;
    left: 0px;
}
label::after {
    content:"";
    width: 8px;
    height: 8px;
    background-color: #666666;
    position: absolute;
    left: 2px;
    top: 3px;
    display: none;
}
input[type=checkbox] {
    visibility: hidden;
    position: absolute;
}
input[type=checkbox]:checked + label::after {
    display: block;
}
input[type=checkbox]:active + label::before {
    background-color: #DDDDDD;
}
```

```
<form>
    <div class="checkbox">
        <input id="check_me" type=checkbox />
        <label for="check_me">Label for checkbox</label>
    </div>
</form>
```

▶ Run code snippet   Expand snippet

This solution *hides the checkbox*, and adds and styles pseudoelements to the label to create the visible checkbox. Because the label is tied to the hidden checkbox, the input field will still get updated and the value will be submitted with the form.

And if you're interested, here's my take on radio buttons: http://jsfiddle.net/DtKrV/2/

Hope someone finds this useful!

Share Improve this answer Follow   edited Sep 26, 2019 at 14:28   community wiki

4 revs, 4 users 79%
MusikAnimal

literally the only method that appears properly on the current firefox at the time of this comment 16 mar '23. crazy that this is still an issue. – Jony Thrive Mar 16, 2023 at 23:17

I answered it originally and it then it got changed into a community wiki. On one hand, I don't know if I'm still getting reputation points from it, but if it's staying up-to-date then I say all answers should be community wikis! – MusikAnimal Apr 12, 2023 at 5:59

7

If you use **ASP.NET Web Forms** you don't need to worry about DIVs and other elements or fixed sizes. We can align the `<asp:CheckBoxList>` text by setting `float:left` to the CheckboxList input type in CSS.

Please check the following example code:

```
.CheckboxList
{
    font-size: 14px;
    color: #333333;
}
.CheckboxList input
{
    float: left;
    clear: both;
}
```

.ASPX code:

```
<asp:CheckBoxList runat="server" ID="c1" RepeatColumns="2"
CssClass="CheckboxList">
</asp:CheckBoxList>
```

Share Improve this answer Follow

edited Jul 29, 2013 at 21:30

community wiki
Rama Subba Reddy M

6

I've never had a problem with doing it like this:

```
<form>
  <div>
    <input type="checkbox" id="cb" /> <label for="cb">Label text</label>
  </div>
</form>
```

Share Improve this answer Follow

answered Nov 20, 2008 at 19:52

community wiki
John Topley

5 As I've said to previous posters who recommended that: I don't like it because it requires unnecessary markup. Also, I tried that markup but it was difficult to prevent the label from wrapping beneath the input (while still having label/input group each on their own lines). – One Crayon Nov 20, 2008 at 19:56

8 So, instead of "unecessary" markup (used by probably almost everyone), you'd rather have unecessary CSS? – Robert C. Barth Jan 30, 2009 at 6:49

11 The problem with wrapping stands. But in general, yes I'd rather have extraneous CSS than markup since the CSS is cached, but the markup may have to be loaded anew for every new page. – One Crayon Jan 31, 2009 at 3:49

6 Compare: 1 instance of extra CSS -vs- many instances of extra markup. – Greg Aug 17, 2011 at 21:27

5

With an input type checkbox wrapped inside the label and floated to the left like so:

```
<label for="id" class="checkbox">
    <input type="checkbox" id="id">
    <span>The Label</span>
</label>
```

this worked for me:

```
label.checkbox {
    display: block;
}
.checkbox input {
    float: left;
    height: 18px;
    vertical-align: middle;
}
.checkbox span {
    float: left;
    line-height: 18px;
    margin: 0 0 0 20px;
}
```

Make sure the height of the is identical to the line-height of the (blocklevel) .

Share Improve this answer Follow

edited Dec 29, 2013 at 10:07

community wiki
3 revs, 3 users 55%
Matijs

4

Hardcode the checkbox's height and width, remove its padding, and make its height plus vertical margins equal to the label's line-height. If the label text is inline, float the checkbox. Firefox, Chrome, and IE7+ all render the following example identically: http://www.kornea.com/css-checkbox-align

Share Improve this answer Follow

edited Aug 9, 2014 at 12:37

community wiki
3 revs
user2407309

Nice! But you should consider copying the HTML and CSS code in here. Answers that depends so much on an external link are discouraged. The link is still valuable to see the results, but I think showing the code in here will help you get more up votes. – Mariano Desanze Aug 8, 2014 at 23:49

nope, they don't (compare in FireFox and in Chrome, in Chrome the checkboxes are above the baseline), I've explained why – YakovL Jun 28, 2019 at 10:49

Why, you claim the thing that's unsupported even with screenshots from both browsers. It is your job to provide proofs for a claim when you make one, not others' to disprove it. I did compare those and would add the screenshots, but comments don't support adding pictures; you can take a look at pictures in my answer though. This however may differ with browser versions, so providing screenshots with annotations containing browser versions would be helpful. – YakovL Oct 27, 2019 at 10:20

Visit the page I linked to in different browsers. – Val Kornea Oct 27, 2019 at 10:46

So I did that, did you? :) Don't you see the difference? imagebin.ca/v/50stCZZGl7Ug imagebin.ca/v/50stNBiv29Ks While in FireFox they are perfectly aligned, in Chrome the checkbox is about the baseline (by the way, you can use mentions via @ if you are interested in quicker replies) – YakovL Nov 5, 2019 at 15:13

4

The following gives pixel-perfect consistency across browsers, even IE9:

The approach is quite sensible, to the point of being obvious:

1. Create an input and a label.
2. Display them as block, so you can float them as you like.
3. Set the height and the line-height to the same value to ensure they center and align vertically.
4. For ***em*** measurements, to calculate the height of the elements, the browser must know the height of the font for those elements, and it must not itself be set in em measurements.

This results in a globally applicable general rule:

```
input, label {display:block;float:left;height:1em;line-height:1em;}
```

With font size adaptable per form, fieldset or element.

```
#myform input, #myform label {font-size:20px;}
```

Tested in latest Chrome, Safari, and Firefox on Mac, Windows, Iphone, and Android. And IE9.

This method is likely applicable to all input types that are not higher than one line of text. Apply a type rule to suit.

Share Improve this answer Follow

edited Sep 16, 2015 at 14:08

community wiki
2 revs
Henrik Erlandsson

hm, this does look well when the label starts with a lowercase letter, but if it starts with a capital one, it's much worse. Could you add a snippet here? – YakovL Jun 28, 2019 at 10:25

4

```
input[type=checkbox]
{
    vertical-align: middle;
}
```

```
<input id="testCheckbox" name="testCheckbox" type="checkbox">
<label for="testCheckbox">I should align</label>
```

▶ Run code snippet   Expand snippet

Share Improve this answer Follow

answered Jun 13, 2017 at 21:43

community wiki
Prime

3

I usually leave a checkbox unlabeled and then make its "label" a separate element. It's a pain, but there's so much cross-browser difference between how checkboxes and their labels are displayed (as you've noticed) that this is the only way I can come close to controlling how everything looks.

I also end up doing this in winforms development, for the same reason. I think the fundamental problem with the checkbox control is that it is really two different controls: the box and the label. By using a checkbox, you're leaving it up to the implementers of the control to decide how those two elements are displayed next to each other (and they always get it wrong, where wrong = not what you want).

I really hope someone has a better answer to your question.

Share Improve this answer Follow

answered Nov 20, 2008 at 18:24

community wiki
MusiGenesis

3

If you're using Twitter Bootstrap, you can just use the `checkbox` class on the `<label>`:

```
<label class="checkbox">
    <input type="checkbox"> Remember me
</label>
```

Share Improve this answer Follow

answered Apr 6, 2013 at 18:31

community wiki
Sunil D.

3

CSS:

```
.threeCol .listItem {
    width:13.9em;
    padding:.2em;
    margin:.2em;
    float:left;
    border-bottom:solid #f3f3f3 1px;
}
.threeCol input {
    float:left;
    width:auto;
    margin:.2em .2em .2em 0;
    border:none;
    background:none;
}
.threeCol label {
    float:left;
    margin:.1em 0 .1em 0;
}
```

HTML:

```
<div class="threeCol">
    <div class="listItem">
        <input type="checkbox" name="name" id="id" value="checkbox1" />
        <label for="name">This is your checkBox</label>
    </div>
</div>
```

The above code will place your list items in threecols and just change widths to suit.

Share Improve this answer Follow

edited Dec 29, 2013 at 10:08

community wiki
3 revs, 3 users 70%
Philip Bevan

3

Use simply `vertical-align: sub`, as pokrishka already suggested.

Fiddle

HTML Code:

```
<div class="checkboxes">
    <label for="check1">Test</label>
    <input id="check1" type="checkbox"></input>
</div>
```

CSS Code:

```
.checkboxes input {
    vertical-align: sub;
}
```

Share Improve this answer Follow

edited May 23, 2017 at 12:26

community wiki
2 revs
Marco Sulla

3

So I know this has been answered many times, but I feel I have a way more elegant solution than those that have been provided already. And not only 1 elegant solution, but 2 separate solutions to tickle your fancy. With that said, everything you need to know and see are contained in 2 JS Fiddle's, with comments.

Solution #1 relies on the native "Checkbox" of the given browser, though with a twist... Its contained in a div which is easier to position cross-browser, with an overflow: hidden to chop the excess of a 1px stretched checkbox (this is so you cant see the ugly borders of FF)

Simple HTML: **(follow the link to review the css with comments, code block is to satisfy stackoverflow)** http://jsfiddle.net/KQghJ/

```
<label><div class="checkbox"><input type="checkbox" /></div> Label text</label>
```

Solution #2 uses the "Checkbox Toggle Hack" to toggle the CSS state of a DIV, which has been properly positioned across browser, and setup with a simple sprite for the checkbox unchecked and checked states. All that is needed is to adjust the background-position with said Checkbox Toggle Hack. This, in my opinion, is the more elegant solution as you have more control over your checkboxes & radios, and can guarantee they look the same across browser.

Simple HTML: **(follow the link to review the CSS with comments, code block is to satisfy StackOverflow)** http://jsfiddle.net/Sx5M2/

```
<label><input type="checkbox" /><div class="checkbox"></div>Label text</label>
```

If anyone disagree's with these methods, please leave me a comment, I would love to hear some feedback on why others have not come across these solutions, or if they have, why I see no answers here regarding them? If anyone sees one of these methods fail, it would be nice to see that too, but these have been tested in the latest browsers and rely on HTML / CSS methods that are quite old, and universal as far as I have seen.

Share Improve this answer

Follow

edited Jan 25, 2018 at 8:40

community wiki
2 revs, 2 users 90%
NinjaKC

2

This is not the best way of going about solving the issue

```
vertical-align: middle
```

Adding `style="position:relative;top:2px;"` to the input box would move it down `2px`. So depending on your font size, you can move it along.

Share Improve this answer

Follow

edited May 29, 2013 at 23:04

community wiki
rene

2

`position: relative;` has some issues in IE with z-index and animations like jQuery's slideUp/slideDown.

CSS:

```
input[type=checkbox], input[type=radio] {
    vertical-align: baseline;
    position: relative;
    top: 3px;
    margin: 0 3px 0 0;
    padding: 0px;
}
input.ie7[type=checkbox], input.ie7[type=radio] {
    vertical-align: middle;
    position: static;
    margin-bottom: -2px;
    height: 13px;
    width: 13px;
}
```

jQuery:

```
$(document).ready(function () {
    if ($.browser.msie && $.browser.version <= 7) {
        $('input[type=checkbox]').addClass('ie7');
        $('input[type=radio]').addClass('ie7');
    }
});
```

The styling probably needs tweaks depending on the font-size used in `<label>`

PS:
I use ie7js to make the css work in IE6.

Share Improve this answer Follow

edited Dec 29, 2013 at 10:14

community wiki
2 revs, 2 users 69%
Bob Fanger

2

Simplest solution is

```
input {
  vertical-align: text-top;
}
```

▶ Run code snippet   Expand snippet

Share Improve this answer Follow

answered Nov 17, 2021 at 12:30

community wiki
ALeX inSide

---

1 2 Next

Highly active question. Earn 10 reputation (not counting the association bonus) in order to answer this question. The reputation requirement helps protect this question from spam and non-answer activity.