Create a css checkbox menu

IMPORTANT in the one using spans, the hamburger is just for show, a big checkbox is located right ontop of it. Do this or use a clickable label as is done in the block display approach

Block Display Approach

This is the least lines of code and great for something quick.

HTML

Code Hierarchy

1. Code Hierarchy
   - <div class="nav-wrapper">
     - children
   - <input type="checkbox" id="nav-trigger" name="menu">
   - <label for="nav-check-trigger">Put icon or whatever here</label>
   - <div class="nav-link-container">
     - children
   - ul
     - children
   - links

CSS

nav-wrapper [type=checkbox] { display=none; }::

Make the checkbox invisible

nav-link-container {display: none;}::

Make the links automatically disappear

#nav-trigger:checked ~ * ~ .nav-link-container { display: block; }

Make everything in the .nav-link-container display as block element

Styling

- nav-wrapper for general positioning and layout
- nav-link-container for styling how your list is organize, select ul and li when necessary

  input[type=checkbox]:checked ~ label {}

  Style what the actual clickable icon will do and look like when clicked here

- add a class to the label and stylize how it looks here.

> **NOTE** Be aware of what is being selected to animate.

Slide in Approach

Built off the logic of the simple approach but also including some other methods that could be used instead of the above version. * animations * css made icon

Slide Out Menu

Based on the Hamburger turned into an X method, if you use the above setup, be aware of how display hidden-block relations are set NOTE: These steps are related to the span Hamburger

HTML

ul class="menu"

Doesn’t have to be a ul, but be aware of selection hierarchy in the css

- **position: absolute** Will keep it in a fixed position above everything — especially good for ful opaque background
- **width: ###px** Set however wide you need your menu
- **padding** Minimally consequential to the function of this, but remember to add it otherwise it will look ugly and possibly render weirdly
- **padding-top** Actually important, set this to keep links from conflicting with menu button

margin: -100 0 0 -50

This is where the magic begins. -100 sets it off the page from the top, 0 right and 0 bottom for a left side nav (have -50 in right position if right mounted), and -50 to pull the whole ul box off the page to the left

Transformation in HTML

transform-origin: 0% 0%

This is the point that the transformation oves from. point of origin.

transform: translate(-100%, 0)

horizontal, then vertical. negative number will move from left to right, a positive number will move from right to left

> **IMPORTANT** Make sure that your top wrapper position settings are correct.

transition: transform 0.4s(time) ease or a cubic-bezier()

Set the time of the transition and the movement style, ease works well, you can use cubic beziers too.

Make it slide

#nav-wrapper input:checked ~ ul { transform: none; }

This makes the whole thing work. It is predicated on being able to select the ul element as sibling to the input:checked. If this relationship isn’t correct, then u need to change how u select.

NOTE

Keep in mind how ~, ~ * ~, and + can be used to select things. Similarly with child selectors.

Span Hamburger (can turn into an X)

This works by making the checkbox really big and positioning it ontop of the hamburger/x shape using z-index

Setup

HTML

- Create a nav wrapper div
- Set the input type to checkbox and give it an id
- 3 empty spans placed together
  - These need to by stylized to be visible
- make the <ul> sibling to the spans and the input
- links are of course children to the ul

NOTE Configuration of your hierarchy will be really important for extra functionality.

CSS

1. The hamburger Icon (Static)

   .nav-wrapper span

   Makes it look like something

   - **display: block**
   - **width** Set to a few px bigger than the input. **height** set to 4ish px
   - **margin-bottom: 5px** to separate the lines, different sizes are cool
   - **position: relative** so it will be overlapped by the input
   - **z-index: 1** to keep it visible but below the input

2. The Super sneaky checkbox!

   .nav-wrapper input {}

   most important for functionality

   - **width** and **height** must be as big as the **span** hamburger
   - **position: absolute**

    - **top, right, bottom, left** settings to locate above the hamburger
    - **display: block** (so it's able to be used, unlike the previous method)
    - set **cursor: pointer** — so it feels clickable
    - **z-indez: 2** or whatever number will place it ontop
    - **opacity: 0;** make it insible

Animations

Add to the following declarations

nav-wrapper span::

set your animation movements etc here * **transform-origin: 0px 0px** to set position * **transition: transform time(s) cubic-bezier(0.77,0.2, 0.05,1.0)** works well - within this transition declaration, also set background and opacity, opacity can be a little longer (like 2-3s). One of the ease's works best with the above cubic-bezier

Set transform origin

This doesn't seem as necessary, but set the origins by calling * nav-wrapper span:first-child:: { transform-origin: 0% 0%; } * nav-wrapper span:th-last-child(2) { transform-origin: 0% 100%; }

Create Crossmark

Creating the shape for when the checkmark is checked .Top line nav-wrapper input:checked ~ span:: * **transform: rotate(45deg) translate(-2px, -1px);** to rotate the spans by 45 degrees, translate adjusts the actual position to account for rotation * make sure that **opacity: 1;** — but potentially not necessary * **background-color** may need to be changed depending on what's going on with the background

Bottom Line

nav-wrapper input:checked ~ span:nth-last-child(2)

> transform: rotate(-45deg) translate(0, -1px);
>
> - this will rotate it the oposite and accont for changing placement based on rotation **this may be able to be messed with through the transform origins that were set on the spans**

nav-wrapper input:checked ~ span:nth-last-child(3)::

- opacity 0
- transform: rotate by 0 deg and make small with scale (0.2 works well)