Dokumentation

Dropdownmenü

Die Navigationsleiste aller HTML-Seiten ist im Stil eines Dropdownmenüs erstellt. Im Folgenden wird beschrieben, wie wir diese Navigationsleiste erstellt und gestylte haben.

* Mit „:root“ wer die Variablen für die Hintergrundfarben und die Höhe des Dropdownmenüs festgelegt.
* Mit „\*“ werden Styles festgelegt, die für alle HTML-Elemente gelten.
* Mit „nav“ werden die Höhe und die Farbe der Navigationsleiste festgelegt. Diese wird vertikal zentriert.
* Mit „nav .logo“ wird das Logo der Seite gestylte und dessen Position festgelegt.
* Mit „nav ul“ wird die eigentliche Navigation festgelegt. Sie wird ans Ende der Zeile gesetzt und die einzelnen Menüpunkte werden in einer Reihe angeordnet. Die Stichpunkte der Menüpunkte werden entfernt.
* Mit „nav li“ wird die Größe der Kästen der einzelnen Menüpunkte festgelegt.
* Mit „nav li:hover“ ändern die Kästen ihre Farbe, wenn der Mauszeiger innerhalb dieser ist, zu der Farbe, die im „root“ als „—accent-color“ deklariert wurde.
* Mit „nav ul a“ wird dafür gesorgt, dass der Link schon am unteren Rand des Kästchens anklickbar ist und nicht erst wenn der Mauszeiger direkt über dem Link ist. Der Link nimmt also die gesamte Größe des „li“ ein. Zudem werden die Unterstreichungen der Links mit „text-decoration:none“ entfernt.
* Mit „.dropdown“ wird das eigentliche dropdown erstellt. Es ist wichtig, dass hier und im „li“ die Position „position:relative“ ist, da das Dropdown sich relativ zu seinem „li“ positionieren soll. Das Dropdown wird unterhalb der Navigationsleiste positioniert, hierfür gibt es im „root“ die Variable „—navbar-height“
* Mit „dropdown li“ werden die Kästen der Menüpunkte im Dropdown gestylte. Sie sind so breit wie die Menüpunkte der Navigationsleiste.
* Mit „dropdown li a“ wird der Text der Menüpunkte linksbündig. Mit „padding“ wird der Abstand zum linken Bildschirmrand bestimmt. Aufgrund des „padding“, muss die Breite angepasst werden, da die Punkte sonst schon außerhalb der Box angeklickt werden können.
* Mit „nav li:hover .droopdown“ wird das Dropdownmenü erst dann sichtbar, wenn man mit dem Mauszeiger über den Menüpunkt fährt, bei welchem das Dropdownmenü sichtbar werden soll.
* Mit „nav input[type=“checkbox“]“ wird die checkbox vor dem Menüpunkt mit dem Dropdownfunktion entfernt.
* Mit „.expandable\_li“ wird der Text des Menüpunkts mit Dropdownfunktion so formatiert, wie der Text der anderen Kästen.
* Mit „@media“ wird die Seite „responsive“ gemacht.

quiz.css

In der css-Datei „quiz.css“ werden die Checkboxen des Quiz erstellt und gestylte. Wir sind folgendermaßen vorgegangen…

* Mit „h1“ und „span“ wird die Überschrift der „quiz.ejs“ Seite so formatiert und gestylte, dass sie im Stil des Webseitenlogos ist.
* Mit „.inputcontainer“ werden die „container“ der einzelnen Fragen zu einer Klasse zusammengefasst. Jede Frage hat daher einen „.inputcontainer“.
* Mit „.inputcontainer input“ wird die vom Browser automatisch erzeugte checkbox entfernt.
* Mit „.checkmark“ wird wird eine custom checkbox erstellt.
* Mit „.inputcontainer:hover input~.checkmark“ werden die checkboxen grau, wenn man mit dem Mauszeiger darüber fährt.
* Mit „input[type=checkbox]:checked+.checkmark“ wird die checkbox beim Klicken auf diese Grau hinterlegt.
* Mit „.inputcontainer input:checked~.checkmark:after“ bleibt die Box nach dem Anklicken grau hinterlegt.
* Mit „.inputcontainer .checkmark:after“ wird die checkmark so gestylte, wie sie am Ende nach dem Anklicken der Box zu sehen ist.
* Mit „#change-color-button“ wird der button, der zum Überprüfen der markierten Kästchen geklickt wird mit einem Rahmen versehen und dessen Größe und Farbe wird festgelegt.
* Mit „#change-color-button:hover“ nimmt der Button die Akzentfarbe an und die Textfarbe wird weiß, wenn der Mauszeiger über dem Button ist.
* Mit „.container“ wird die weiße „box“ erstellt, in der sich das Quiz befindet.

Quiz (quiz.js)

Im JavaScript „quiz.js“ wird zunächst die Variable „changeColorButton“ deklariert. Mit „querySelector“ werden anhand der angegebenen „id’s“ Elemente im HTML-File ausgewählt. Diese werden dann für die Erstellung des Quiz in Variablen gespeichert.   
Somit benötigt jede Antwortmöglichkeit eine eigene „id“ und eine eigene Variable im JavaScript mit der zugehörigen „id“.

Allgemeines Codebeispiel:

var=Variablenname

const var = document.querySelector(‚#id‘);

if (var.checked) {

    var.nextElementSibling.style.backgroundColor = 'red'/‘green‘;

  } else {

    var.nextElementSibling.style.backgroundColor = '';

  }

Der obenstehende Code unterscheidet sich in der „id“ und im Variablennamen. Jede Antwortmöglichkeit hat seine eigene „id“ und seine eigene Variable.  
Ist die Antwort falsch, so wird die angeklickte Box rot, ist sie richtig, wird sie grün. Hierfür wird die if/else Funktion verwendet. Die Farben ändern sich erst, wenn der Button gedrückt wird. Es können mehrere Felder ausgewählt werden, da es mehr als eine richtige Antwort geben kann.