Rezepte-App

Ziel: 16. Juni 2016

* Web-App auf dem Handy (html5-Anwendung)

1. Rezepte-App mit Bild, Zutaten (personenporioniert), Beschreibung, Einkaufsliste und Stoppuhr
2. 1. Entwurf Layout

🡪 Unterschiedliche Seiten

|  |  |  |  |
| --- | --- | --- | --- |
| START  Rezept  Einkaufsliste  Stoppuhr  🡪 Links | REZEPT  Bild  Zutaten (Person)  Beschreibung | EINKAUFEN  Zwiebel  300g Hackfleisch  1kg Tomaten | Stoppuhr  Zähler ist gestellt  Am Ende ertönt Klingel |

Sliden zur nächsten Seite

1. 1. Idee zur Umsetzung

Rezept (objekt) 🡪 auf eine Seite

* Bild
* Zutat 1 \* Menge 1
* Zutat 2 \* Menge 2
* Zutat 3 \* Menge 3
* ...
  + auf Personen runterrechnen
* Zeit 1
* Zeit 2
* Freitext
* Notiz (optional), bzw. Rezepte editieren

1. 2. Seite

* Einkaufsliste
* Zutat 1 \* Menge 1
* Zutat 2 \* Menge 2
* Zutat 3 \* Menge 3
* .....

🡪 persistent speichern

1. Komponenten

* Eingabe
* Zutat in Einkaufsliste hinzufügen (Add)
* Einkaufsliste speichern (Save)
* Löschen von Einkaufsliste (Delete bzw. Clear)
* Strukturierte Daten persistent Speichern
* Editieren durch Nutzer
* Suche in den Zutaten (welches Rezept besitzt diese Zutaten?)

1. Plan

* JSON Doku lesen
* Objektstruktur (semistrukturiert) anlegen
* Rezepte einfügen
* Daten laden
* Menü erstellen mit Buttons zu Rezept, Einkaufsliste (drawMenu (...), Funktionen von Tasten)
* Einkaufsliste erstellen 🡪 darin suchen
* Zutaten auf Personen runterrechnen (🡪 in g/kg, damit berechenbar ist)
* Stoppuhr je nach Schritt auf Zeit voreingestellt (verstellbar) als Pop-Up (im Pop-up soll ein Startknopf sein. Uhr soll ablaufen und zum Schluss klingeln. <https://www.sitepoint.com/build-javascript-countdown-timer-no-dependencies/>
* Speichern (persistent)
* Mehrere Rezepte, blättern in Rezepten zu nächster Rezeptseite
* Rezepte erstellen/editieren (Nutzer)
* Verschönerung der App
* Web-App auf Handy

1. Objektstruktur

Zuerst versuchen wir die Struktur für das Rezept zu erstellen.

Die Überlegung war, ob man alles als ein gemeinsames Objekt macht mit unterschiedlichen Strings, die jeweils Arrays haben oder mehrere Objekte mit verschiedenen Strings.

Wir versuchen es mit einzelne Objekten.

„Rezept“ {

„Zutaten“: [

{„zutat1“:“Karotten“,“zutat2“:“Tomaten“,„zutat3“:“Zwiebeln“}, 🡪1. Rezept

{„zutat1“:“...“,“zutat2“:“...“,“zutat3“:“...“} 🡪 2. Rezept

],

„Menge“: [

{„menge1“:800,“menge2“:500,“menge3“:1}

],

}

Jedoch wird es im Browser in unserem html Dokument nicht so angezeigt.

Daher müssen wir wohl jedes Rezept einzeln eingeben und nicht mehrere Rezepte in ein Objekt einpassen.

<http://www.tutorialspoint.com/json/json_objects.htm>

Wie kann man am besten die Struktur von Zutat 1 + Menge 1 darstellen?

Versuche waren mit einer Tabelle

Jsonversuch8.html

Oder ohne json nur anhand von javascript.

Maikes Version. Test.html

Beide Versionen haben wir zusammengefügt.

Maiketest.html

Datenstruktur

Rezept:

* Liste 🡪 Array

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

* Textfeld

|  |
| --- |
|  |

* Zeit

|  |
| --- |
|  |

* (optional: Portionen)

|  |
| --- |
|  |

var aktRezept;

aktRezept.text

aktRezept.zeit

aktRezept.liste.zutat[0].name

aktRezept.liste.zutat[0].menge

aktRezept.liste.zutat[0].einheit

for (aktRezept.liste.length)

Außerdem:

In unsere Tabelle für die Zutaten, unsere Liste, fügen wir noch einen Button hinzu. Dieser fügt die jeweilige Zutat in die ShoppingList

addtoShoppingList

(tomate, 500, g)

Buttons von <http://www.iconsdb.com/custom-color/arrow-icons.html>

#3950E6

+, 🡪, 🡨

2. Rezept erstellt:

Jamies Tomatensuppe: jamie.html

Von Rezept zu Rezept blättern

Als Nächstes:

Menü erstellen mit Buttons zu Rezept und Shoppinglist, optionale Suche nach Zutaten in Rezepten, Bilder, Begrüßung

Shopping-list

Rezept

Home

Rezept

Shoppinglist

Startseite Seite von Rezepten mit Menü

Shopping-List erstellen als shoppinglist.js

Und Button mit Shopping-List verknüpfen. Wie man Zutat in shopping-List übernimmt.

Problem:

Wie speichert man alles persistent? In einer Library.

Deshalb am besten alles in einem js-Dokument speichern.

Das bedeutet, dass man die anderen Rezepte alle in einem Dokument haben wird. Diese müssen dann sichtbar und unsichtbar durch button-drücke gemacht werden.

Praktisch: jamie.html und salat.html in ein Dokument einfügen

= allinone.html

Problem: nur noch ein rezept wird angezeigt, da sie übereinander liegen.

Wenn Library verwendet wird 🡪 passender Code gefunden 🡪 darf man verwenden

🡪 **zitieren**

🡪 begründen

🡪 verstehen durch **kommentieren** des Codes

* Prototyping

Suche: jQuery Liste

Selbst Zutaten hinzufügen, Portionen runterrechnen.

🡪 pfeffer.html, magnus.html

Shoppinglist zum eintragen.

🡪 umändern, damit man schon vorhandenes in die shoppinglist adden kann.

Quelle für Paprika-Zucchini-Nudeln

[file:///Users/laraherbrich/Desktop/Programmieren/hilfe.html#](file:///Users/laraherbrich/Desktop/Programmieren/hilfe.html)

neu: 🡪hilfe.html

Erstellung von Buttons, zur verlinkung auf functions muss die richtige Zeichensetzung beachtet werden. Da sich alles in einem Javascript-Script befindet, muss man html-code in anführungszeichen setzen. Damit functions funktionsfähig sind, und als javascript angezeigt werden, müssen sie mit \“ ..... \“ gekennzeichnet werden.

hinzufügen von eintrag in die shoppingliste.

Durch function addToListe() wird ein Eintrag, der zuvor in function zeigeRezept() definiert wurde, in die shoppinglist übernommen. Dafür wird push-Befehl verwendet.

Diese werden in einer ungeordneten Liste in Klick-Reihenfolge aufgeführt.

um alles in der shoppinglist zu löschen benutzen wir splice <http://www.mediaevent.de/javascript/Javascript-Array-Methoden.html>

ermöglicht vollständige Leerung der Liste.

Problem: leere Liste wird erst angezeigt, wenn erneut zeigeShoppinglist() aufgerufen wird.

Schwierigkeiten auch beim löschen von einzelnen Einträgen in der Shoppinglist. Bisher hat dies nur durch .pop-funktion geklappt, doch diese löscht jeweils nur den letzten Eintrag in der Shoppingliste.

Shoppinglist wird erst durch klicken auf den Link: shoppingliste angezeigt. Dabei verschwinden die Rezepte. Welche durch klicken auf den Link:Rezept angezeigt werden, wodurch die shoppingliste wieder verschwindet.

Rezepte auf Portionen runterrechnen. Dafür neue function erstellt: function zeigePortionen(). In dieser wird die Menge durch 2 geteilt. Damit wird von 4 auf 2 Portionen runtergerechnet. Function wird am Ende im display ausgegeben. Runtergerechnete Menge kann in shoppingliste durch hinzufügen übernommen werden.

Um auf 1 Portion runterzurechnen, wurde die selbe Rechnung übernommen wie bei 2 Portionen. Man hat das ausgehende Rezept durch 4 geteilt.

Maybe Baby <https://wiki.selfhtml.org/wiki/JavaScript/Anwendung_und_Praxis/komfortable_Timer-Funktion> Timer

Hat funktioniert!!! Der Timer zählt runter, wenn man auf den Button klickt. Wenn der Timer fertig gezählt hat, erscheint ein Pop-Up durch alert() mit Fertig. Dazu soll noch ein Ton erklingen.

Problem: Der Timer bleibt bei 1 Sekunde stehen. Das ist doof, da man so die vorgegebene Zeit nicht mehr sieht, die runtergezählt wurde. Wenn man erneut auf den Timer klickt, während er läuft, setzt sich die Zeit wieder auf den Anfang zurück.

Für den Timer und die Portionen wurden verlinkte Buttons benutzt. Die Timer-Buttons wurden direkt in den Zubereitungstext integriert.

Quelle: <http://www.w3schools.com/html/html5_webstorage.asp>

<https://wiki.selfhtml.org/wiki/JavaScript/Web_Storage#localStorage>

Wir haben es geschafft die Shoppinglist zu speichern. Mit der Funktion populateStorage () werden die Inhalte der Shppinglist im Key shoppingInhalt gespeichert. Dafür ist der Befehl localStorage.setItem maßgebend. Diese function ist in die function addToListe() integriert.

Um alles zu löschen, wurde der Befehl localStorage.clear() der function allesLoeschen () hinzugefügt.

Darauf wird man auf die Hauptseite verwiesen.

Problem: wenn die Shoppinglist leer ist, kann man nicht auf sie zugreifen. Jedoch wird beim Anklicken der Shoppinglist, wenn diese leer ist, ein Fenster aufgerufen, dass sich keine Einträge in der Shoppinglist befinden.

<http://soundbible.com/1496-Japanese-Temple-Bell-Small.html>

unser Sound für das Ende des Timers.