

Diese Anleitung zeigt am Beispiel der Umrechnung von Windgeschwindigkeiten die Erstellung einer einfachen plattformübergreifenden App (HTML5) mit Hilfe des Intel XDK. Der Schwerpunkt liegt auf Berechnung mit mathematischen Formeln und der Nutzung von Javascript.

Als Voraussetzung ist bereits das Intel XDK (<https://software.intel.com/en-us/intel-xdk>) installiert.¹ Diese Anleitung bezieht sich auf die Version 3522 des Intel XDK. Aktuell werden aber in kurzen Abständen neue Versionen veröffentlicht, bei denen sich die Bedienung teilweise erheblich verändert.

Erforderliche Vorkenntnisse:

- Sicherer Umgang mit dem Dateisystem des verwendeten Betriebssystems
- ggf. die Grundlagen zum Arbeiten mit XDK (siehe Skript „Eine Reiseführer-App erstellen“)

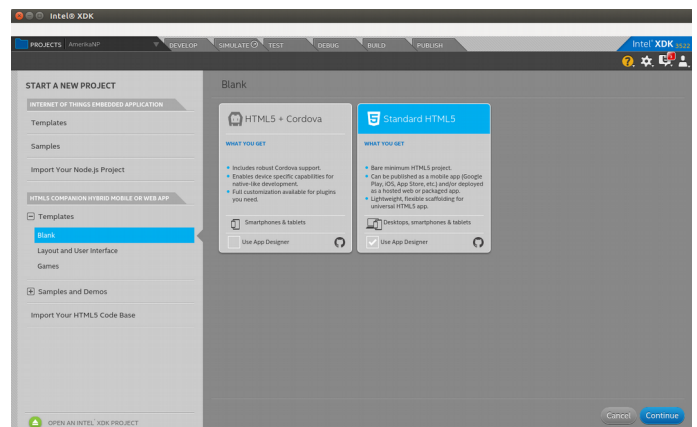
1. Eine App anlegen

Die App, die du hier erstellst, kannst du dir wie eine computergestützte Präsentation (PowerPoint oder Impress) vorstellen. Sie besteht aus mehreren Seiten, auf denen du Texte, Bilder, Karten, Videos usw. einfügst. Zwischen den Seiten kannst du z.B. mit Buttons hin- und herspringen.



Starte das Programm „Intel XDK“. Solltest du auf dem Welcome-Bildschirm nach einer Anmeldung gefragt werden, kannst du einfach auf „Later“ klicken.

Beginne dein neues Projekt mit dem „App Designer“. Klicke dazu unter „HTML5 COMPANION HYBRID MOBILE OR WEB APP“ auf Templates → Blank. Wähle dann „Standard HTML5“ und setze unbedingt den Haken bei „Use App Designer“ und klicke rechts unten auf „Continue“.

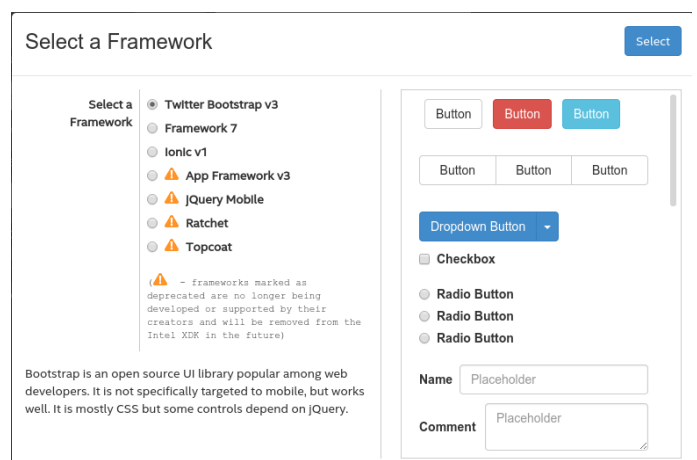


Gib deinem Projekt einen Namen und lege den Pfad fest, in dem deine App erstellt werden soll.

Im folgenden Fenster wird dir eine kurze Einführung angeboten. Diese kannst du dir anschauen oder auch überspringen.

Nun fragt dich das XDK, welches Framework du einsetzen willst. Ein Framework ist sozusagen der Unterbau deiner App. Je nach Framework sehen die Schaltflächen usw. unterschiedlich aus. In diesem Beispiel wird das Framework „Twitter Bootstrap v3“ benutzt. Damit passt sich die spätere App optimal an die Optik des Betriebssystems des Endgeräts an.

Nun erzeugt das XDK eine neue App. Sie besteht bisher nur aus einer einzelnen Seite, die du mit Inhalt füllen kannst. Zunächst werden wir eine Titelseite erstellen.



¹ Nach der Installation muss man sich einmal mit einem registrierten Benutzernamen anmelden. Anschließend kann wieder auf Logout geklickt werden. Nun lässt sich das Programm auch ohne Anmeldung nutzen, die Anmeldemaske kann ignoriert und mit „Cancel“ weggeklickt werden, sie erscheint aber immer wieder.

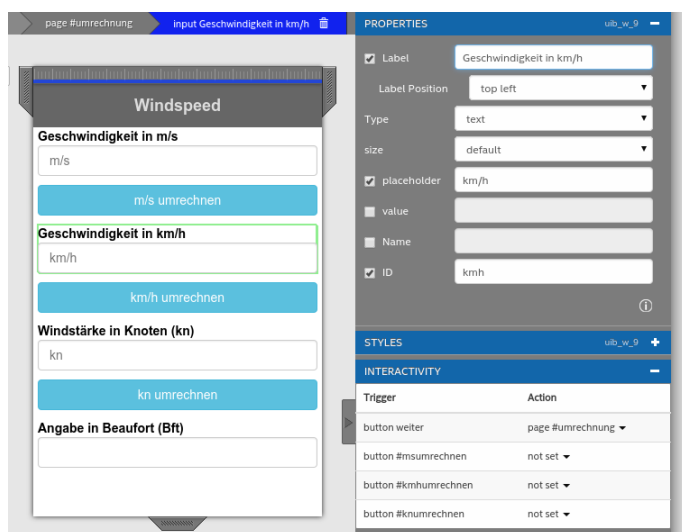
2. Inhalte einfügen

Zunächst kannst du eine **Startseite** anlegen. In unserem Beispiel wurde ein Header, ein Bild, ein Text und ein Footer per drag and drop in die App gezogen. Der Button „weiter“ liegt im Footer und verweist auf die zweite Seite.

Wenn du nicht mehr weißt, wie die Elemente angelegt bzw. konfiguriert werden, kannst du im Skript „Eine Reiseführer-App erstellen“ nachschauen.

Für die eigentliche Umrechnung wurde eine eigene Seite angelegt. Auf ihr befindet sich ein Header. Für jede Einheit musst du außerdem ein Input-Feld (mit einer eindeutigen ID) und einen Button für die Umrechnung erstellen.

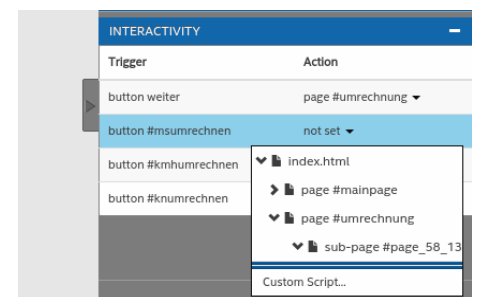
In unserem Beispiel sieht das folgendermaßen aus:



3. Eigene Skripte erstellen (Javascript)

Eine Formel kannst du dem XDK nicht per drag and drop beibringen. Deshalb musst du ein eigenes Script erzeugen. Dazu wählst du unter „INTERACTIVITY“ hinter deinem Umrechnungsbutton statt „not set“ „Custom Script...“

Mit einem Klick auf „EDIT SCRIPT“ kommst du in den Editor.



Der Cursor steht schon an der Stelle, an der du den Quellcode bearbeiten musst. Unterhalb des Namens des entsprechenden Buttons findest du den Text „/* your code goes here */“. Diesen Text löschst du und ergänzt dafür die folgenden Zeilen:

```

14
15      /* button #msumrechnen */
16      $(document).on("click", "#msumrechnen", function(evt)
17      {
18          var mswert = document.getElementById('ms').value;
19          var kmhwert = mswert * 3.6;
20          var knwert = mswert / 0.5144;
21          document.getElementById('kmh').value = kmhwert;
22          document.getElementById('kn').value = knwert;
23          return false;
24      });

```

```

12      /* button weiter */
13      $(document).on("click", ".uib_w_4", function(evt)
14      {
15          /*global activate_page */
16          activate_page("#umrechnung");
17          return false;
18      });
19
20      /* button #msumrechnen */
21      $(document).on("click", "#msumrechnen", function(evt)
22      {
23          /* your code goes here */
24          return false;
25      });
26

```

Erklärung:

In Zeile 18 wird die Variable „mswert“ definiert und ihr der Wert aus dem Feld mit der ID „ms“ zugewiesen.

In Zeile 19 wird die Variable „kmhwert“ definiert und ihr neuer Wert berechnet ($1 \text{ m/s} = 3.6 \text{ km/h}$).

In Zeile 20 wird die Variable „knwert“ definiert und ihr neuer Wert berechnet ($1 \text{ kn} = 0.5144 \text{ m/s}$).

In Zeile 21 wird dem Feld mit der ID „kmh“ der Wert der Variablen „kmhwert“ zugewiesen.

Achtung: Alle Klammern und Strichpunkte müssen genau so gesetzt werden. Solltest du einen Fehler beim Abtippen gemacht haben, wird deine App nicht funktionieren.

Für die Umrechnung ausgehend von km/h sieht der passende Quellcode dann folgendermaßen aus:

```

34      /* button #kmhumrechnen */
35      | $(document).on("click", "#kmhumrechnen", function(evt)
36      | {
37          var kmhwert = document.getElementById('kmh').value;
38          var mswert = kmhwert / 3.6;
39          var knwert = mswert / 0.5144;
40          document.getElementById('ms').value = mswert;
41          document.getElementById('kn').value = knwert;
42          return false;
43      });
44

```

Die Umrechnung von Knoten funktioniert entsprechend.

Prinzipiell ließe sich auch die Beaufortskala berechnen. Wir wählen hier aber den Weg der If-Abfrage. Damit du diese nicht für jeden Button neu schreiben musst, erstellst du im Quellcode eine „function“. Der beste Platz dafür ist vor der Befehlszeile „function register_event_handlers()“

Erklärung:

In Zeile 8 wird die Funktion definiert. Der Wert in der Klammer wird vom Funktionsaufruf (siehe unten) übergeben. Wir können also mit diesem Wert arbeiten.

In Zeile 11 wird die Variable bftwert definiert.

Zeile 12: Wenn der übergebene Wert (knwert) kleiner als 1 ist, wird in die Variable bftwert der Text „1 – leichter Zug“ geschrieben.

Zeile 13-23: Wenn die IF-Abfrage nicht erfolgreich war, werden nun weitere Abfragen gestartet.

Zeile 24: War keine IF-Abfrage erfolgreich, so wird der Befehl hinter else abgearbeitet.

In Zeile 25 wird dem Feld mit der ID „bft“ der Wert der Variablen „bftwert“ zugewiesen.

```

8      function beaufort_bestimmen(knwert)
9      | {
10     |     var bftwert = 0;
11     |     if (knwert < 1) {bftwert = '1 - leichter Zug';}
12     |     else if (knwert < 4) {bftwert = '2 - leichte Brise';}
13     |     else if (knwert < 7) {bftwert = '3 - schwache Brise';}
14     |     else if (knwert < 11) {bftwert = '4 - mäßige Brise';}
15     |     else if (knwert < 16) {bftwert = '5 - frische Brise';}
16     |     else if (knwert < 22) {bftwert = '6 - starker Wind!';}
17     |     else if (knwert < 28) {bftwert = '7 - steifer Wind!';}
18     |     else if (knwert < 34) {bftwert = '8 - stürmischer Wind!';}
19     |     else if (knwert < 48) {bftwert = '9 - Sturm!';}
20     |     else if (knwert < 56) {bftwert = '10 - schwerer Sturm!';}
21     |     else if (knwert < 64) {bftwert = '11 - orkanartiger Sturm!';}
22     |     else {bftwert = '12 - Orkan!'}
23     |     document.getElementById('bft').value = bftwert;
24     | }
25     | function register_event_handlers()
26     | /
27
28
29

```

Letzendlich musst du noch festlegen, wann die neu geschriebene Funktion aufgerufen werden soll. Dazu ergänzt du bei jedem Button die Zeile „beaufort_bestimmen(knwert);“. Damit wird die Funktion aufgerufen und die Windgeschwindigkeit in Knoten an die Funktion übergeben.

```

56      /* button #kmhumrechnen */
57      | $(document).on("click", "#kmhumrechnen", function(evt)
58      | {
59      |     var kmhwert = document.getElementById('kmh').value;
60      |     var mswert = kmhwert / 3.6;
61      |     var knwert = mswert / 0.5144;
62      |     document.getElementById('ms').value = mswert;
63      |     document.getElementById('kn').value = knwert;
64      |     beaufort_bestimmen(knwert);
65      |     return false;
66      | }
67      | );
68
69
70

```

4. Extra

Da sich beim Berechnen oft Zahlen mit vielen Nachkommastellen ergeben, ist es sinnvoll, diese zu runden. Dazu kannst du die mathematische Funktion „Math.round“ nutzen.

```

/* Rundung auf zwei Stellen: */
Math.round(kmhwert*100)/100;

```