

Index of App Inventor 2 Projects

Gallery Project Name	Gallery link	Doc link	Description
add_hours	ai2.appinventor.mit.edu/?galleryId=6397302665510912	https://community.appinventor.mit.edu/t/need-help-datetime-hours/19055/8?u=abg	This poster wanted to add hours to a date and time.
angry_balls_v3	ai2.appinventor.mit.edu/?galleryId=5597402344980480	https://docs.google.com/document/d/1vMOewOPYmN4gzbdIXJgLeCQ1W5MzuT7ogXQqV96R0P8/edit?usp=sharing	Billiards collision angle simulation test bed.
App_TRansbluetooth_copy12	https://gallery.appinventor.mit.edu/?galleryId=9533ad18-0b79-4cf9-a1ee-5645c30f069f	https://community.appinventor.mit.edu/t/how-can-i-make-the-receiving-data-word-from-bluetooth-module-display-in-the-text-box-and-also-translate-it-in-speech-without-clicking-text-to-speech-button/107002/18?u=abg	Sample app that receives LF delimited text from BlueTooth, translates it to Nepali, then announces it out loud. A TinyDB Namespace is added to save old translations, to avoid using up the translator budget.

autofill_example	ai2.appinventor.mit.edu/?galleryId=5077800976646144	https://docs.google.com/document/d/1SakUBoYhbGbH4UtYARNgYoRAAO_dtpLyVp4Jx3Bri2k/edit?usp=sharing	An example of how to do automatic word completion in a text box
AsciiCHRASCprocedures	ai2.appinventor.mit.edu/?galleryId=5086373932040192	https://groups.google.com/d/msg/mitappinventortest/zl5JB4ZNfXc/0mYq_VKtAAJ	Utility procedures for byte list conversion to text based on Italo ASC and CHR procedures
blackjack	(too big)	https://community.appinventor.mit.edu/t/need-help-making-my-blackjack-game/145453/10?u=abg	Single player Blackjack sample, with shuffled deck and multiple Ace valuation options.
blank_delimited_data_stream	https://gallery.appinventor.mit.edu/?galleryId=074b3220-9fee-46d6-9fdd-e0e6bbe1a491	https://community.appinventor.mit.edu/t/how-do-i-filter-receiving-bluetooth-data/45627/3?u=abg	Sample code to distribute data values from a data stream with blank separators and alternating tags and values into Labels, using a dictionary to lookup the Labels by tag.
blind_quiet_clock	ai2.appinventor.mit.edu/?galleryId=5853955980263424	https://community.appinventor.mit.edu/t/how-do-you-make-an-action-happen-when-you-slide-your-finger-onto-a-sprite/57480/6?u=abg	This is a simulation of a silent watch for blind operation. Try to tell the time by slowly dragging your finger across the watch face. It is rigged to vibrate when you cross a hand or travel outside the watch face.

blocks_repair_kit	https://gallery.appinventor.mit.edu/?galleryid=198d5dee-85fe-4b77-ac23-96d09e3ae934	https://community.appinventor.mit.edu/the-blocks-area-did-not-load-properly-changes-to-the-blocks-for-screen-5475942398296064-menu-will-not-be-saved/86361/18?u=abg	A tool to search extracted .bky and .scm screen files for orphan blocks (missing parent components in the .scm file.) Needs 7-zip to extract .scm and .bky files from the .aia file.
BlueTooth_delimiter_sample	https://gallery.appinventor.mit.edu/?galleryid=65c87a31-92f7-41c2-a8cb-58ad1a524496		This is a starter kit with blocks for receiving delimited strings from simple (pre-BLE) BlueTooth devices, with multiple readings per line.
bouncy_beats_label	ai2.appinventor.mit.edu/?galleryid=5759494649937920	https://groups.google.com/d/msg/mitappinventortest/2SoKmgGVYvI/P0uppE0-FgAJ	This app demonstrates how to flash a beat count based on elapsed time for accuracy.
bouncy_bouncy	ai2.appinventor.mit.edu/?galleryid=6023434952900608	https://groups.google.com/d/msg/mitappinventortest/2SoKmgGVYvI/wJV-xO07FgAJ	This app demonstrates exact timing for a beating Ball. By calculating the time phase of a sin() function against absolute time from start, accuracy is assured, though it might skip beats.

breathe	ai2.appinventor.mit.edu/?galleryId=6299893844017152	https://groups.google.com/d/msg/mitappinventortest/2SoKmgGVYvI/ya8DRoPNFgAJ	This is a breathing exercise app, with programmable duration for inhale/exhale and hold times, in seconds.
browse_JSON	ai2.appinventor.mit.edu/?galleryId=6442549083111424	https://docs.google.com/document/d/1p6-HWASYVueO_9yqWgaYNAXmpm5bilw-s4wfg6WB9Wc/edit?usp=sharing	This is a general purpose utility to browse the tree structure of a JSON string in a ListView. It is meant to help explore the structure of big JSON texts.
browse_JSON_V2	ai2.appinventor.mit.edu/?galleryId=6577697141293056	https://docs.google.com/document/d/1p6-HWASYVueO_9yqWgaYNAXmpm5bilw-s4wfg6WB9Wc/edit?usp=sharing	This is an enhanced JSON browser, with the path added.
browse_XML_V5	ai2.appinventor.mit.edu/?galleryId=5635235807363072	https://community.appinventor.mit.edu/t/an-xml-browser-with-path-display/9899	This is a sample app to use to help navigate deep XML or JSON text. A sample xml.txt file is pre-loaded in the Media folder, or you can use its Web component to get more data to parse.
BT_bytelist	https://gallery.appinventor.mit.edu/?galleryId=8eb32cf7-e611-4c7b-9e4c-782b43380165	https://community.appinventor.mit.edu/t/how-can-i-receive-bluetooth-module-hex-text/39024/5?u=abg	Example code for receiving Unsigned Bytes from BlueTooth, with hex text conversion and accumulation.

bubble_sort_buttons_V2	https://gallery.appinventor.mit.edu/?galleryid=f15c5e61-cfd2-4c1f-bb5c-a26652cbc386	https://community.appinventor.mit.edu/t/drag-and-drop-dynamic-created-buttons-and-update-the-list/81173/18?u=abg	Word swap puzzle with an arcade claw machine simulation using a html Label and superscript codes.
bullet_cloud	ai2.appinventor.mit.edu/?galleryId=5349781031747584	https://community.appinventor.mit.edu/t/changing-sprite-in-blocks/8244/6?u=abg	This is an example of aging multiple balls using a slow clock and a list of colors by age.
button_calendar	ai2.appinventor.mit.edu/?galleryId=5847156225343488	https://docs.google.com/document/d/1Ctf-tl-oRaoziGqb-V-h9ATAtVhCPvHY65EoVW-FpZA/edit?usp=sharing	This was in response to a challenge to make a calendar made up of buttons. It's a little slow in the emulator, and can be optimized further with date-specific refreshes.
button_calendar_generic	ai2.appinventor.mit.edu/?galleryId=5059089367957504	https://docs.google.com/document/d/1Ctf-tl-oRaoziGqb-V-h9ATAtVhCPvHY65EoVW-FpZA/edit?usp=sharing	This is the generic version of the button_calendar app
button_list	ai2.appinventor.mit.edu/?galleryId=6037870371143680	https://docs.google.com/document/d/1RbPhl6cnHz3pRfqpEOVtRoMdkQIEA3Qm5BtuYq4bbGY/edit?usp=sharing	This is a sample snippet showing how to load a small list of buttons from a scrollable larger list of colors and texts.
byebyeAI	ai2.appinventor.mit.edu/?galleryId=6104963528523776	https://groups.google.com/d/msg/mitappinventortest/QBw5qNHEsxc/EmdKoQLMDwAJ	Sample countdown timer to show off Instants and Duration between Instants

canvas_slider	ai2.appinventor.mit.edu/?galleryId=4695823085207552	https://groups.google.com/d/msg/mitappinventortest/eA_T9NWTa6A/SZTS48_FAAAJ	Sample of using a thin Canvas as a Slider. No Sprite needed.
canvas_slider_vertical	https://gallery.appinventor.mit.edu/?galleryId=34e03e26-3eb4-4074-a1e7-e59fbbf61eda	https://community.appinventor.mit.edu/t/vertical-slider-appers-as-a-non-visible-component/133721/25?u=abq	This is an example of how to use a thin Canvas as a vertical Slider.
canvas_wrap	ai2.appinventor.mit.edu/?galleryId=6567192240455680	https://groups.google.com/d/msg/mitappinventortest/OSBB8tNNEUg/ozoaX2d5BwAJ	Sample code for letting a Ball wrap around the screen when it hits an edge.
capital_state_country_puzzler	http://ai2.appinventor.mit.edu/?galleryId=5791382083731456	https://docs.google.com/document/d/1lD_fcEUBCumN41DWbouORK7fj8E6aoRgY19-kgjaEPM/edit?usp=sharing	This is based on the Will Shortz NPR puzzler of 11/12/2017. Find a US State name that can be found in the concatenation of a US state capital name and a country name.
chat_labels	ai2.appinventor.mit.edu/?galleryId=5638829530808320	https://groups.google.com/d/msg/mitappinventortest/WEat3QJXUX4/-r3mNVSIAwAJ	This is an example of using pairs of left and right-aligned html labels to display a chat, using a Ball on a Canvas to drag the list offset. Chat courtesy of Abbott and Costello.

checkbox_list	https://gallery.appinventor.mit.edu/?galleryid=a550091b-b200-40af-825e-d63fc8af3c2e	https://community.appinventor.mit.edu/t/command-as-a-value-in-a-variable/46291/4?u=abghttps://community.appinventor.mit.edu/t/command-as-a-value-in-a-variable/46291/4?u=abg	Sample project to show how to use generic component blocks with list blocks and procedures, using a list of checkboxes.
chess_board	https://gallery.appinventor.mit.edu/?galleryid=04c7af52-bf49-4b00-9e95-43d75f29cb93	https://community.appinventor.mit.edu/t/online-chess-game-for-multiplayers-in-app-inventor/18005/6?u=abg	This is a starter kit for chess, using List Pickers and Unicode chess pieces.
chess_board_buttons	https://gallery.appinventor.mit.edu/?galleryid=fde06dd4-9265-468b-a7f5-8787619bcb0d	https://community.appinventor.mit.edu/t/default-clouddb-multiple-screen-intermittent-errors/25768?u=abg	This is a bare bones multiplayer chess game, using the default MIT CloudDB server to pass game history back and forth between players using the same CloudDB tag to contain a list of board positions. An 8 by 8 grid of buttons is used as the game board, with Unicode chess pieces as the button text. A chat feature passes messages through the game history list.

clapper	ai2.appinventor.mit.edu/?galleryId=6069097627123712	https://docs.google.com/document/d/1v--m632Y121CeIYOH02FfD5pBdpCbO-pkVemsZdx8ac/edit?usp=sharing	This is a small test of the speech recognizer, turning the screen background on or off based on hearing the user say "Clap On" or "Clap Off". No hand clapping is involved, just speech.
clock_always_fires_test	https://gallery.appinventor.mit.edu/?galleryId=75bd1416-89be-4199-bd53-ce59ca8f3976	https://community.appinventor.mit.edu/t/what-means-timeralwaysfires/9661/5?u=abg	A test bed for the Clock attributes • clock always fires • clock enabled
closest_sprite	https://gallery.appinventor.mit.edu/?galleryId=ad8b8baf-04ee-4dc5-8e7a-6ee3d2a79b94	https://community.appinventor.mit.edu/t/sprite-proximity-detection/100487/4?u=abg	Find the closest sprite to a given sprite.
Clo			

CloudDB_top_5_scores	https://gallery.appinventor.mit.edu/?galleryid=0ed85d2e-ae2e-4268-adda-e99b02e574d5	https://community.appinventor.mit.edu/top-5-scores-to-clouddb-single-tag/98972/3?u=abg	Simplest possible example of storing top 5 scores and player names in CloudDB under a single tag. Uses the new nb191 sort blocks for sorting.
Color_code_scroller	ai2.appinventor.mit.edu/?galleryid=4965345777025024	https://community.appinventor.mit.edu/move-a-scroll-bar/2645/6?u=abg	This is an example of a Canvas used to drive a horizontal scroll of several buttons across a downloaded color table.
color_match	https://gallery.appinventor.mit.edu/?galleryid=99273b74-30e1-4511-9aae-38abec1f1d0e	https://community.appinventor.mit.edu/two-buttons-selected/93759/5?u=abg	Match two buttons with the same color. Uses procedures and generic blocks.
Colors	ai2.appinventor.mit.edu/?galleryid=5787277265600512	https://groups.google.com/d/msg/mitappinventortest/PnRY9dY9Qjo/alt4oQD6EAAJ	A bare bones lookup of color by name

combination_lister	ai2.appinventor.mit.edu/?galleryId=5128353902362624	https://docs.google.com/document/d/1wgyItSb1xBkjUy4prDCIU3BmwFVgQiJuyCIMpKnr7UA/edit?usp=sharing	This sample app was developed in response to a post on the MIT App Inventor board: https://groups.google.com/d/msg/mitappinventortest/1_S1OtROT9w/Yz-VhyC8CQAJ It shows how to generate a list of all combinations of n out of m items, where $n \leq m$.
CRLF	ai2.appinventor.mit.edu/?galleryId=5926305773387776	https://docs.google.com/document/d/1rlcA0cOFbitKRB_4kpV4_KZuKfMDcEn_s6HOXfx5T_Q/edit?usp=sharing	This little project shows how to use one of the Web component blocks to do hex conversion. This trick works only one byte at a time.
csv_vertical_display	https://gallery.appinventor.mit.edu/?galleryId=6b927ecd-ec3b-443e-ad8a-9542e187c5e8	https://community.appinventor.mit.edu/t/which-is-the-block-that-defines-any-item-in-the-list-to-be-added-to-another-list/53632/40?u=abg	General purpose utility to show a really wide CSV table in a ListView
cylinder	ai2.appinventor.mit.edu/?galleryId=5525819635138560	https://groups.google.com/d/msg/mitappinventortest/BNuHDmniGi0/pDquEkxoeGaj	This app demonstrates the volume and area of a cylinder, with slider control.

daily_appointments	ai2.appinventor.mit.edu/?galleryId=6361588663975936	https://docs.google.com/document/d/12JOU4ju1uEB5RFRcS8ofx4raSkEC9Lps_oQypgBfj-M/edit?usp=sharing	How to store a day's appointments into a dictionary in time order.
daily_schedule	https://gallery.appinventor.mit.edu/?galleryId=7b9de6fb-caac-44ad-b463-8682fc5a09ca	https://community.appinventor.mit.edu/t/daily-schedule-in-tinydb-starter-kit/128237?u=abq	Another daily schedule, list based.
Database_Primer_JSON	ai2.appinventor.mit.edu/?galleryId=5372495967027200	https://docs.google.com/document/d/1B9WSSsaT5Pvr-N2pkalJy--XVJcOHXUH5N9H8PmgN1w/edit?usp=sharing	This is a first attempt at coding a general purpose data editor, where the field specifications are brought in through a JSON schema, and the data itself is kept in a JSON format.
date_differences	ai2.appinventor.mit.edu/?galleryId=6069685827928064	-	This app tries to tackle the tricky issue of counting years, months, and days between two dates. This is controversial.
Day_Number	-	https://groups.google.com/d/msg/mitappinventortest/xBOILPFHKA4/IIWvGXNnAQAj	Snippet showing how to save a start date in TinyDB, and show today's offset from that in days.

day_of_year	ai2.appinventor.mit.edu/?galleryId=6436320627523584	https://groups.google.com/d/msg/mitappinventortest/w33gmbSnzNc/gnnIzIHEAQAJ	How to calculate day of year based on the Gregorian Calendar, without using the Clock component or Java internals.
Deep_Copy_Dict	https://gallery.appinventor.mit.edu/?galleryId=630d28fe-08ad-48a0-92c5-d9f989fb9726	https://community.appinventor.mit.edu/t/copy-dictionary-doesnt-work-or-what-am-i-missing/46845/10?u=abq	A sample procedure to construct a deep copy of a dictionary structure. The AI2 copy dictionary block only does a shallow copy.
delayed_output	ai2.appinventor.mit.edu/?galleryId=4521982895456256	https://groups.google.com/d/msg/mitappinventortest/Rvi_tBMT8bQ/zWFd-MMVBQAJ	This is a sample solution to the problem of wanting delayed output to go to different output Labels, depending on the most recently pressed Button.
diagonal_bounce_test	ai2.appinventor.mit.edu/?galleryId=6367475708198912	https://docs.google.com/document/d/1g71sl_rR7PqHGB83tAYZfNi7HvzIkWZ-xsvhFUxpplo/edit?usp=sharing	This app demonstrates how to bounce a ball off a spinning line, using math. To start the app, drag the screen to position the spinning red line, then poke the screen to attract the bouncing ball to it.

Documenter	ai2.appinventor.mit.edu/?galleryId=6000702845878272	https://docs.google.com/document/d/1S_aCm1ZyKq3riZwv7rQibr5tZatggLLkiSzneCI9YI4/edit?usp=sharing	This sample shows you how to present a tree of documentation using a single List Picker. It needs to be upgraded to file data, not block data.
DocumenterV4	ai2.appinventor.mit.edu/?galleryId=5429926277021696	https://community.appinventor.mit.edu/t/listpicker-elements-not-showed-on-first-pick/20029/10?u=abg	This is a List Picker tree of text loaded from a CSV table file, arbitrarily deep, with bread crumbs at item 1.
drag_and_swap_JSON	ai2.appinventor.mit.edu/?galleryId=6258216806383616	https://docs.google.com/document/d/1AuhZjdMP55CI9uOrrPG6x_FpZHIPInTaUfEhiwzDUh8/edit?usp=sharing	This sample is a detailed explanation of how to avoid sprite cannibalism and how to encode and load game data in JSON files.
draw_star	https://gallery.appinventor.mit.edu/?galleryid=6739c807-5bf9-4b49-a765-8e4b16e8bb97		How to draw multi-point stars on an AI2 Canvas.
drink_recipes	ai2.appinventor.mit.edu/?galleryId=5120380011020288	https://groups.google.com/d/msg/mitappinventortest/656oDTQeomU/7-WNXaxBGwAJ	Load a file with a table of recipes, and search it by ingredients using list filter procedures

DumpTinyDB	ai2.appinventor.mit.edu/?galleryId=6232347342536704	https://groups.google.com/d/msg/mitappinventortest/vSTqVZAMnDE/0B4bvRhCcgAJ	General blocks example showing how to dump the complete contents of TinyDB. Also see TinyDB_export_import
eight_queens	ai2.appinventor.mit.edu/?galleryId=5838678182723584	-	This is the 8 Queens puzzle. Arrange 8 chess Queens on the chess board so that no two of them threaten each other.
eight_queens_generic	ai2.appinventor.mit.edu/?galleryId=5858910198693888	https://github.com/AGetzler/eight_queens_generic	This is the generic event form of the eight queens problem, using a single event for 64 possible button clicks
Excel_trigonometry	ai2.appinventor.mit.edu/?galleryId=5251866711162880	https://community.appinventor.mit.edu/t/how-to-make-a-link-between-an-excel-file-and-appinventor/3324/1?u=abg	This is a trigonometry library for people trying to convert an Excel trigonometry formula to an AI2 trigonometry formula. Everywhere the Excel formula calls a trig function, replace that call in AI2 with a call to the equivalent function in this library

exercise	ai2.appinventor.mit.edu/?galleryId=5542207801655296	https://community.appinventor.mit.edu/trying-to-get-a-pause-in-procedure-execution/24765/5?u=abg	Here's a scheduler for the only hardware available to me, myself. It's a silly exercise routine, where I have to perform irregularly spaced actions, signalling by button press completion of the action. The frequency of the action requests is hard wired into the Clock1.Timer code.
expandable_view	ai2.appinventor.mit.edu/?galleryId=6132352252903424	https://docs.google.com/document/d/1wBDcINy2AdMPfAlvUFxXfmDGitiD6-krhJk3GyZAHVs/edit?usp=sharing	This sample app turns a label and four buttons into an expandable/collapsible List View.
Ferret football	ai2.appinventor.mit.edu/?galleryId=6067163933310976	https://groups.google.com/d/msg/mitappinventortest/dB-YS07FbCc/3kh_7z1BgAJ	This is a rewrite of a student project, Ferret Football, avoiding collision event problems.
find_in_tree	ai2.appinventor.mit.edu/?galleryId=6235836271689728	https://groups.google.com/d/msg/mitappinventortest/FNms9TpBZxc/BGPa0hSQAAAJ	This JSON-like lookup utility will search by tag in a tree.

firebase_data_changed_timestamped	https://gallery.appinventor.mit.edu/?galleryid=7a5f568c-4f1b-45e0-aeb9-e98421562d50	https://community.appinventor.mit.edu/slowdown-in-retrieving-tags-from-firebase-in-real-time/84392/20?u=abg	Demo of the Firebase Data Changed event catching inserts of timestamped data.
FireBase_dump	https://gallery.appinventor.mit.edu/?galleryid=c924029e-8f2c-42fa-bc28-40ba6af5370	https://community.appinventor.mit.edu/login-error-using-firebase/78915/29?u=abg	This is a general purpose FireBase dumper/explorer. To use it, copy a FireBase component from another project by highlighting it in the Designer and hitting Ctrl-C. Come back here to Screen1 in the Designer and hit Ctrl-V to paste in that FireBase component.
first_run	ai2.appinventor.mit.edu/?galleryid=5722367573360640	https://groups.google.com/d/msg/mitappinventortest/L-lbKTr1AVU/avR3_T7iBwAJ	Sample code to detect first run of an app.
FormFiller	ai2.appinventor.mit.edu/?galleryid=6698050900852736	https://docs.google.com/document/d/1zfz1EiJdiDsf2j2amlc7IfJzcJt9dzKGmneuxpONpso/edit?usp=sharing	This is an example of using a List Picker popup to show which text boxes in a form have been omitted, then sending focus to the selected text box. Text boxes are identified by their hints.

Fruit_Stand	ai2.appinventor.mit.edu/?galleryId=5298226067537920	https://docs.google.com/document/d/1WxV3Ny9yM9YobYJSJ8CTd70ds5W9Illa1HhDEHF8kDI/edit?usp=sharing	This is a sample fruit stand app showing how to keep a price list and use it with a purchases list to calculate a grand total.
fusion_bulk_insert	ai2.appinventor.mit.edu/?galleryId=5379104097173504	https://docs.google.com/document/d/12g_66X0xoXrbfCHFOo838umQC8vE7lwhpxGQLsZQI9A/edit?usp=sharing	Example of building a bulk INSERT to Fusion Tables (6 INSERTs separated by semicolons) (obsolete)
fusion_delete	ai2.appinventor.mit.edu/?galleryId=6284888085102592	https://docs.google.com/document/d/11GNsnGmpp89cjQ7h1CUmZXZPqTefyvNyPmvkBqthIJC/edit?usp=sharing	This is a code sample showing one way to collect a list of Fusion Table ROWIDs and to submit DELETES of those rows from that list. (obsolete)
future_date_picker	ai2.appinventor.mit.edu/?galleryId=4902235747778560	https://community.appinventor.mit.edu/t/add-date-limit-to-date-picker/36409/2?u=abg	Build your own Date Picker that won't allow present or past dates.
genealogy_ABG	ai2.appinventor.mit.edu/?galleryId=6318137005441024	-	First shot at a genealogy app. Test data is off the top of my head.

Go Scroll Yourself		https://community.appinventor.mit.edu/t/go-scroll-yourself-blocks-only-no-extensions-no-javascript/60499	This tutorial is intended for people who need to display a heavily customized scrollable window into a large table. It can also be used as a CardView.
Got_Your_Nose	ai2.appinventor.mit.edu/?galleryId=5874223173206016	-	Can you associate the famous nose with its owner? Simple file input to list, matching text with images.
group_sum	https://gallery.appinventor.mit.edu/?galleryId=242e79ab-da18-4c17-840a-a3e19df4b980	https://community.appinventor.mit.edu/t/search-for-same-ingredient-and-add-them/38315/4?u=abg	Sample procedures that can simulate SQL SUM GROUP BY operations on a table (list of lists).
hex_input	ai2.appinventor.mit.edu/?galleryId=5195586314043392	https://community.appinventor.mit.edu/t/data-validation/17359/13?u=abg	This code sample shows how to make your own hex keyboard, and allow its use over multiple inputs. To switch focus of the keyboard, I had to use Button.Text as my keyboard target, and the button click event to open and close the keyboard.

histogram	https://gallery.appinventor.mit.edu/?galleryid=277d522b-23b0-41fa-a5e9-e94b49e199c9	https://community.appinventor.mit.edu/storing-result-in-tinydb-and-using-its-data-for-bar-chart/109044/8?u=abg	A sample bar chart showing the distribution of a sequence of random integers in the domain [0....25] using the AI2 bar chart facility, fed from a TinyDB tag/value.
I_jump_JY61_MPU6050_JSON_V6	ai2.appinventor.mit.edu/?galleryId=5768301433847808	https://groups.google.com/d/msg/mitappinventortest/4AvhJkZ4fA/OhOJIBIBwAJ	Accelerometer byte stream parsing for JY61 MPU6060 with JSON output using hex conversion and 8 bit to 16 bit conversion.
IEEE754	ai2.appinventor.mit.edu/?galleryId=6180600181686272	https://community.appinventor.mit.edu/crc-calculate-crc-heck-code/76620/17?u=abg	This is an example of bit manipulation, in particular the decoding of an IEEE754 floating point number 32 bit string into its decimal fraction equivalent. This was done for some one who couldn't find the printf verb in his MCU compiler.

JSON_encoder	ai2.appinventor.mit.edu/?galleryId=5323667108790272	https://docs.google.com/document/d/19J6bcPBj1kux_FwjU0l0mvXyh6XeBVQ58nxH9lOh3V0/edit?usp=sharing	Recursive JSON encoder produces indented JSON text from lists with object structures. Sample data included. Fragile, and beware reordering of attributes by Web component JSON Decode block.
JSON_finder	ai2.appinventor.mit.edu/?galleryId=4514816886702080	https://groups.google.com/d/msg/mitappinventortest/Kb_rp_iKm8QU/5NvhdPoECgAJ	This is a sample project to search a JSON multi-level tree for a branch object that contains a given tag/value pair. The sample data is from a currency conversion web site.
JSON_finder_V2	ai2.appinventor.mit.edu/?galleryId=5525586186469376	https://docs.google.com/document/d/1Tat3hC0FpZKIWo0eMZqxxsl0-9umstjYpP5KbzftZM/edit?usp=sharing	This sample has procedures that can be used to filter and search JSON structures to arbitrary depth, regardless of structure.

JSON_shredder	http://ai2.appinventor.mit.edu/?galleryId=5566307843702784	https://docs.google.com/document/d/1McqirRZ4brhFGqBQtDz6RAYziB7kWyINTj3CLT3wOG4/edit?usp=sharing	This blocks sample is meant for people who have trouble navigating and filtering JSON data. It shreds and flattens a JSON text into a table, with each row containing the path to a single attribute and value. It is capable of working with JSON objects and arrays, and should give unique rows.
Kakuro_Helper_V01	ai2.appinventor.mit.edu/?galleryId=5770654380064768	https://docs.google.com/document/d/1veFS11e0XNjA6whDdpWgcapRBhw9OSDqZDW8Tvr3gYg/edit?usp=sharing	First attempt at the game Kakuro, with buttons and labels, and an Undo/Redo stack.
Levenshtein_distance	https://gallery.appinventor.mit.edu/?galleryId=0a1b48e7-d656-4c9e-9d3d-4c5cf566a48e	https://community.appinventor.mit.edu/t/similarity-texts/27068/5?u=abg	This calculates the edit distance between two words, also known as the Levenshtein distance. It is the minimal number of single letter edits needed to transform one given word into the second given word.

list_label	ai2.appinventor.mit.edu/?galleryId=5390547027230720	https://docs.google.com/document/d/1ktIK4xUPMRC8IVEB_LkgnBYfgZQsUCRoXbUZVvAQZ0g/edit?usp=sharing	Attempt to use a canvas and a html label as a List View, with customizable element display.
log_last_five		https://community.appinventor.mit.edu/t/clarify-issue-w-looping-bt-serial/34561/2?u=abg	Log the last 5 items to a label
logger	ai2.appinventor.mit.edu/?galleryId=4936043469078528	https://docs.google.com/document/d/1Ob0YrSgN_MzPXshxbFaAdb49J4e6UPQTqXDIUBsqtvM/edit?usp=sharing	This snippet has a reusable procedure to add messages to a log, in the GUI. A multiline inactive text box allows scrolling.
logger_to_list	ai2.appinventor.mit.edu/?galleryId=6053548547244032	https://groups.google.com/d/msg/mitappinventortest/II_RIEIGddU/J0q_E-2rAwAJ	This a small sample logger using a List Picker's Elements to hold the log.
logger_to_TinyDB_list	ai2.appinventor.mit.edu/?galleryId=4702767489089536	https://community.appinventor.mit.edu/t/list-viewer-only-displaying-1-element/17680/3?u=abg	This is a sample logger procedure, saving the log to a Tinydb tag as a list.
Login_TinyDB_Multi User	ai2.appinventor.mit.edu/?galleryId=5628197448318976	https://groups.google.com/d/msg/mitappinventortest/NY7s2-PU7I8/f9r7KMABQAJ	a Multi User TinyDB Login prototype, meant for sharing a single phone among multiple users, offline.

login_TinyDB_V1	-	https://groups.google.com/d/msg/mitappinventortest/1fY5u1C48L0/Dtl89ZmaBwAJ	Starter template for TinyDB login, single user.
marquee	ai2.appinventor.mit.edu/?galleryId=6057785113051136		This is example of how to show a fixed length moving marquee of text, replenished from a long document.
max_score	ai2.appinventor.mit.edu/?galleryId=5919145619881984	https://community.appinventor.mit.edu/t/how-to-save-max-scores-in-tinydb/29454	How to maintain a high scores list in TinyDB, using sorted insertion.
max_score_cloudDB_taglist	https://gallery.appinventor.mit.edu/?galleryId=82ff5fe7-e5ef-430f-bff2-2aa4928856fd	https://community.appinventor.mit.edu/t/how-to-keep-top-scores-in-clouddb-via-taglist/45205?u=abg	This is a sample app to show how to gather a top scores list from CloudDB.
max_score_dated	https://gallery.appinventor.mit.edu/?galleryId=e64ea2a7-7829-4ecc-8d5d-f979dfbd059c	https://community.appinventor.mit.edu/t/saving-highscore/44156/14?u=abg	How to maintain a high scores list in TinyDB, using sorted insertion. Name and date are included along with score.
max_score_TinyWebDB	ai2.appinventor.mit.edu/?galleryId=5568973378289664	https://docs.google.com/document/d/1ccG7Nv_uRfRzGADa_z17Jez9NjV0yBKqFTsuOo4iAixY/edit?usp=sharing	This is a sample app to show off a good but not perfect way to keep a list of high scores under a TinyWebDB tag, using two TinyWebDB controls.

merge_sort	ai2.appinventor.mit.edu/?galleryId=5192781774192640	https://community.appinventor.mit.edu/t/merge-sort-block-procedures-library/2437	A pure blocks-only merge sort, using the recursive Divide and Conquer technique. Speed is $n \cdot \log(n)$.
merge_sort_customizable	https://gallery.appinventor.mit.edu/?galleryId=9027334d-aba8-4764-8221-5fec11fbdfd7	https://community.appinventor.mit.edu/t/sorting-tables-ascending-and-descending-by-column-contents-using-merge-sort/56785?u=abg	A blocks library that can be used to apply the merge sort against lists and tables, with customizable sort orders (ascending or descending) by column number. This is just a collection of procedures, with no user interface.
mileage_tracker	ai2.appinventor.mit.edu/?galleryId=6146358902390784	https://groups.google.com/d/msg/mitappinventortest/t_62BPx9F7U/4rck_Rc8BQAJ	This is a multiple vehicle mileage tracker. Its initial load file and vehicle ID format is distinctive only to this app.
MODBUS_CRC16	ai2.appinventor.mit.edu/?galleryId=5325856548978688		How to calculate a MODBUS CRC16 on a list of unsigned bytes
Modern_Major_General	ai2.appinventor.mit.edu/?galleryId=5479156460814336	https://docs.google.com/document/d/1zQt7H_bsXX0Y0zmwQ1wIEESyH1YyWyYbE1HR2KYW0CI/edit?usp=sharing	This is an example of trying to sing a Gilbert and Sullivan patter song using the text to speech component. It's awful.

Morse	ai2.appinventor.mit.edu/?galleryId=6212271021883392	https://docs.google.com/document/d/1nmovfgkql83YK1jRVdys3UAiikz9qtQZgdblsa7yCY/edit?usp=sharing	Morse code demo
Morse_Decode	ai2.appinventor.mit.edu/?galleryId=5064093575282688	https://groups.google.com/d/msg/mitappinventortest/G20UeeX8M-A/hidVyiWVCAAJ	Morse Code Decoder. You enter the dots and dashes, and it translates them back into letters.
move_to		https://community.appinventor.mit.edu/t/move-to-a-point-x-y-smoothly/5194/8?u=abg	Continuous movement to a point on the canvas
multi_chart	https://gallery.appinventor.mit.edu/?galleryId=efd3485c-32e3-4358-8ea0-4bf00dd5a124	https://community.appinventor.mit.edu/t/how-do-i-make-a-chart-from-a-csv-file/81158/12?u=abg	Here is a sample showing off the new DataFile component in relation to the ChartData and Chart components. Aside from 'knowing' that the date and time entries are in columns 1 and 2, this app works entirely off the csv file's headings.
multi_check	ai2.appinventor.mit.edu/?galleryId=4762670566473728	https://docs.google.com/document/d/1V74gl-vHGpUGmLcIMcxj73lhDOZuCZFXVMC0M2T2hNI/edit?usp=sharing	How to simulate multi-select in a list of (true/false) pairs, with simple text flags in a ListView

multi_counter	ai2.appinventor.mit.edu/?galleryId=6093694786994176	https://community.appinventor.mit.edu/tinydb-namespace-update-via-parallel-listviews-scrollable-by-canvas/15904	This is an example of using parallel ListViews to scroll and update a set of counters in a TinyDB NameSpace.
multi_filter	ai2.appinventor.mit.edu/?galleryId=5120947675332608	https://groups.google.com/d/msg/mitappinventortest/7jsH889DYt8/R66q86OXCQAJ	General purpose app to filter any csv by column name and value
multi_remove	ai2.appinventor.mit.edu/?galleryId=6688317277601792	https://community.appinventor.mit.edu/remove-item-from-a-list-of-lists/4920/6?u=abg	Example of how to remove multiple items from a list that match by value.
multi_timer	ai2.appinventor.mit.edu/?galleryId=6577155168010240	https://docs.google.com/document/d/1Gcu-PqYDMHoJVP24ASyzZSxNFrt4HITTtev9dVEgys/edit?usp=sharing	Service multiple events with a List and a single Clock
multicalc	https://gallery.appinventor.mit.edu/?galleryid=da59e9d7-e492-4098-9752-20069f006fff	https://community.appinventor.mit.edu/help-with-a-calculator-app/71228/31?u=abg	A sample app for doing multiple calculation types reusing a few text boxes and Labels, using Generic blocks and Dictionaries.
multiselect	ai2.appinventor.mit.edu/?galleryId=6081151525715968	https://docs.google.com/document/d/1DVfN6JoXaSINY6I6_dLFT1CO3i9TikMS9enyJvbDxq8/edit?usp=sharing	Sample demo app showing how to select multiple items from a list, in two List Pickers

nagger	ai2.appinventor.mit.edu/?galleryId=4744810520510464	https://groups.google.com/d/msg/mitappinventortest/Z4ZbyNfJ9A/GzzniBRVAwAJ	Round robin question/response scheduler, with no Timer events. I don't have BLE or Arduino hardware, so the device being sampled is the user himself, as fast as he can answer yes or no to the questions, until he hits Cancel.
NOAA_Wind	ai2.appinventor.mit.edu/?galleryId=4730786936913920	https://docs.google.com/document/d/1c2f3kJq7UUBcPw2jjQhf-4sLpTZaWR9z9VyOkZ1Vz5Y/edit?usp=sharing	Small app that gets today's wind forecast from the US NOAA weather bureau
notfound_confound	ai2.appinventor.mit.edu/?galleryId=6553745952997376	https://community.appinventor.mit.edu/t/how-to-deal-with-notfound-from-lookups/4783	This is the source code for a short essay on how to deal with notfound on lookup.
Notifier_demo	https://gallery.appinventor.mit.edu/?galleryId=01062e12-2b60-46dd-a9d1-27f58b41b2a1	https://community.appinventor.mit.edu/t/fixing-exporting-and-clearing-my-tinydb-data/65179/38?u=abg	a sample app which demonstrates the two styles of Notifier use
nthMonday	ai2.appinventor.mit.edu/?galleryId=5854062664876032	https://groups.google.com/d/msg/mitappinventortest/wbqUO7pJPxA/W87azFIDEAAJ	For calendar fans, blocks to find the nth Monday of any year, in response to a Sten Jensby challenge.

number_to_text		https://community.appinventor.mit.edu/t/very-large-number-to-text-conversion-experimental-log/18893	Take a number and convert it into English words, including thousands, millions, etc.
numbrix	ai2.appinventor.mit.edu/?galleryId=5479664697081856	https://docs.google.com/document/d/13MwUMqYE08w4gFY0DdsPb7XQ-yQIZPAuRK5rP-sZk/edit?usp=sharing	This app is intended to supply an Undo capability for working the Sunday Numbrix puzzle. The object is to complete a chain of consecutive numbers on the board, proceeding just horizontally and vertically (not diagonally.) This app sample is useful to show how to use a stack for Undo functionality.
opentdb	https://gallery.appinventor.mit.edu/?galleryId=61fbb9da-9f41-48f6-ac74-95facd4ec7f0	https://community.appinventor.mit.edu/t/string-manipulation/47895/35?u=abg	This is a sample project showing how to parse JSON responses from the web site opentdb.com, a free quiz source, and how to present the questions and multiple choice answers in a quiz format.

optional_Textboxes	ai2.appinventor.mit.edu/?galleryId=6294704014622720	https://groups.google.com/d/msg/mitappinventortest/JJ4bMwWg-HU/PtuGOUcfDAAJ	This is a general solution to the problem of how to show only non-blank text boxes and their associated TableArrangements.
ordered_search_and_lookup	ai2.appinventor.mit.edu/?galleryId=6595721213247488	https://docs.google.com/document/d/1TRGypmOaRvK37oazCuaGBdXzvt4dzzGXaHYvp22ueCs/edit?usp=sharing	This is how to provide Divide and Conquer search on a very large lookup table.
Ouroboros	ai2.appinventor.mit.edu/?galleryId=5737715493765120	https://docs.google.com/presentation/d/1XylrxzxksNy4aea8IkuslJOl0DvZgsRniSrqKhkf7pk/edit?usp=sharing	(Also known as Gorgon, or Medusa, freezing all who see it.) This little piece of code shows how you can try to make an AI2 list eat its own tail, and the resultant crash. Can you explain the error, and show you understand lists?
pad	ai2.appinventor.mit.edu/?galleryId=5250466277228544	https://groups.google.com/d/msg/mitappinventortest/f9Q3TyQ--lQ/o4i78fwlCAAJ	This is a small library of text padding functions
parallel_listview_canvas_scroll	ai2.appinventor.mit.edu/?galleryId=6276787224576000	https://docs.google.com/document/d/1oVJwZgxp91M0FPLfFA0WdIGXwwPUa0nkETQ3XJS-z5U/edit?usp=sharing	This is in response to requests on the App Inventor board for ways to scroll parallel ListViews.

PastPresentFuture	https://gallery.appinventor.mit.edu/?galleryid=6ffde740-04b3-402b-9e58-11b2527842b0	https://community.appinventor.mit.edu/linking-multiple-tinydb-together/22637/12?u=abg	This is an example of using multiple TinyDB NameSpaces to store datetimestamped events by Past, Present, or Future, and of sorting them temporaly by tag.
pattern_input	ai2.appinventor.mit.edu/?galleryId=5535792736501760	https://community.appinventor.mit.edu/how-do-i-create-a-sprite-that-moves-from-one-button-to-another/69691/14?u=abg	How to capture a pattern lock using blocks
permute	ai2.appinventor.mit.edu/?galleryId=5199168652705792	https://drive.google.com/open?id=1Vln4SWSkZ8y3bhxPBeEQ6_omhAlu2eZh	Sample result procedure to return a random permutation of a list
pet_breeds	https://gallery.appinventor.mit.edu/?galleryid=fbc6a0c8-d08c-4ce3-91ab-2b6c7511b321	https://community.appinventor.mit.edu/save-two-lists-to-tdb/77376/6?u=abg	Minimal app to maintain a list of pets and their breeds in a TinyDB NameSpace.
phone_scraper	ai2.appinventor.mit.edu/?galleryId=5821614552514560	https://groups.google.com/forum/#!msg/mitappinventortest/eZ76PRYy3Zo/Mz15b6BFAQAJ	These blocks will scrape the first phone number from a piece of text.
pick52	ai2.appinventor.mit.edu/?galleryId=6056759979540480	https://groups.google.com/forum/#!msg/mitappinventortest/NuY5WveY4ck/N-FizbQeCwAJ	This is a quick and simple EuroMillions lotto picker. 1-50 for the first 5 numbers, 1-12 for the next 2 numbers.

pizza_trader	ai2.appinventor.mit.edu/?galleryId=4775793106092032	https://docs.google.com/document/d/1jEx4wJsvGLfsMAjvY5ft2gT8LpTuS2ceLIC03frJaU4/edit?usp=sharing	This is an extension of the Pizza Party tutorial to include selecting and updating rows in a Fusion Table. (obsolete)
pointList	https://gallery.appinventor.mit.edu/?galleryId=e4f08a3d-af41-42ff-b44f-5fb8eb55d073	https://community.appinventor.mit.edu/t/solve-the-error-1004/38276/13?u=abq	A sample project to play with the Canvas Draw Point List block.
pong	ai2.appinventor.mit.edu/?galleryId=5758014207033344	https://docs.google.com/document/d/1lI6fsDne9q2z7RueUWOmVLbU5mcNPYDgw5HxoF4xzI/edit?usp=sharing	Pong for two players, without multi-touch, using the Touch Up event to move the paddles.
progress_canvas	ai2.appinventor.mit.edu/?galleryId=6245169287135232	https://drive.google.com/open?id=1KRnzfuZ4VfPebdh-atXVhzRSd7kL09pZ	This is an example of how to use a thin Canvas as a progress bar. The Slider is only to test the procedure.
race_time	ai2.appinventor.mit.edu/?galleryId=4914165876260864	https://drive.google.com/drive/folders/1ftGa6_oK2PQGF7bgfWwWaDWT9v8Wy8CS?usp=sharing	Small sketch to show how to capture Start time, End Time, and Minutes between them for a race.
radio_buttons	https://gallery.appinventor.mit.edu/?galleryId=458c9419-ecc6-40fc-891d-346c17cd2cf4	https://community.appinventor.mit.edu/t/radiobuttons-not-working-as-expected-default-selection-error/29302/27?u=abq	An example of how to use lists of button components to organize your buttons into radio groups. Also uses dictionaries.

randoms_in_range	ai2.appinventor.mit.edu/?galleryId=4538805925969920	https://groups.google.com/d/msg/mitappinventortest/pOWEZtYjx7s/mh0UrEJ5AQAj	How to generate a random list of different numbers from 1 to a limit.
reaction_time_ABG	https://gallery.appinventor.mit.edu/?galleryId=083b9256-70f2-44c2-86f3-bf37bb2f0e60	https://community.appinventor.mit.edu/t/correct-time-accuracy-for-the-reaction-time-test-app/44461/3?u=abg	Show running ms in a reaction time tester, without lag accumulation.
row_your_boat	ai2.appinventor.mit.edu/?galleryId=5021101293961216	https://drive.google.com/open?id=1R4gDVbR4H4BhDFkNiiCTFPEhnQKifSde	This is an example of how to mix sounds using multiple player components, using the Row Row Row Your Boat song.
Samaritan	ai2.appinventor.mit.edu/?galleryId=4843586337374208	https://drive.google.com/open?id=1GVm7ZyLUC9dJM-re5yLL5FRiN19MgLri	This is how to make your app display text one word at a time in the center of a canvas, like the evil program Samaritan in the CBS series Person Of Interest
schema_enums_samples	ai2.appinventor.mit.edu/?galleryId=6380137059057664		Sample schema hand conversion output
Scrollable_label	ai2.appinventor.mit.edu/?galleryId=5769381581946880	https://drive.google.com/open?id=1m3eb1IMNE9aXEMcgpD9gUYN_m14_Qmrw	Example of how to scroll a label using a thin canvas and Ball.

select_where_sample1	ai2.appinventor.mit.edu/?galleryId=6488648341520384	https://docs.google.com/document/d/12t4HLNRepCnbmcfycdF8wSHWht_D1P8hKJpTknVDChY/edit?usp=sharing	Sample app and procedures to demonstrate filtering a table.
Sheets_Latest_By_Name	ai2.appinventor.mit.edu/?galleryId=6310505935667200	https://community.appinventor.mit.edu/t/maximum-date-timestamp/16680/29?u=abg	This is an example of downloading a Google Sheet CSV table and doing a SELECT GROUP BY with MAX on it, then an OUTER JOIN to pick the rows with maximum datetimes per user.
sheets_to_YAML	(withheld)	https://community.appinventor.mit.edu/t/browse-google-sheets-in-yaml-format/74078?u=abg	Common procedures to use sheet headings to create List Picker Elements in YAML

MarathonSheets	(withheld)	https://community.appinventor.mit.edu/t/logging-marathon-runner-checkpoints-in-google-sheets/70551?u=abg	This sample is intended to show how to use Google Sheets aggregation functions like MAXIFS to summarize multirow log data into single line summaries. This sample uses the New York City Marathon as a base, though it probably can't scale to the 53,000 runners. It is based on three sheets: Checkpoints, a list of locations each runner must cross on his route Mileposts, a detail list of runner checkpoint timestamps Runners, a master list of runners, with identifying bar codes and stats Checkpoints (reference only)
----------------	------------	---	---

show_TinyDB	ai2.appinventor.mit.edu/?galleryId=5442741292630016	https://groups.google.com/d/msg/mitappinventortest/n6uJQdvaHME/ljALjPLKBgAJ	This is a bare bones List Picker to pop up the tags and values in TinyDB1, for diagnostic purposes. Drag the BeforePicking block into your Backpack, for transport across projects.
shred_rss	https://gallery.appinventor.mit.edu/?galleryId=1c0fc6af-6b7b-488b-a669-a19a024ac12e	https://community.appinventor.mit.edu/t/how-do-you-make-a-list-of-all-the-items-i-got-in-an-xml-file-from-the-web/67011/7?u=abg	This is a demo of my shred_table procedure, which can take an XML-like response content and a list of prefixes and suffixes, returning a table of text fragments bounded by those prefixes and suffixes. The first prefix and suffix you supply must precede all the other prefixes and suffixes in each text fragment.
shuffle_balls	ai2.appinventor.mit.edu/?galleryId=6228875187322880	https://docs.google.com/document/d/1IbZxNFvFdnquKsaA9fbhCI3BnYj8G8Sc6m7PV38HmSg/edit?usp=sharing	This is a shuffle effect usable for the shell game. It includes a general purpose shuffle routine.

Sierpinski	ai2.appinventor.mit.edu/?galleryId=5214268554739712	https://drive.google.com/open?id=1LxN548MWCCNZoQlrUye8EzvU8SEAHAp	This app generates a Sierpinski Triangle fractal as you watch, by repeatedly chipping off the top corners of a square to make three smaller squares, then repeating the process for each smaller square.
SieveV2	ai2.appinventor.mit.edu/?galleryId=6477875229491200	https://drive.google.com/open?id=0B5Xr7cVb-tYqS1V6V3Zic3pKQmM	This app shows how to mix real time display and calculation of prime numbers using a sieve of Eratosthenes, and a clock.
SnakePit	ai2.appinventor.mit.edu/?galleryId=5592523749457920	https://docs.google.com/document/d/1JrSnPo-7i2I_am4cHC6TgcpKhQVslZowSW4ooWDc7ww/edit?usp=sharing	This is a sample of how to use lists instead of Sprites to simulate a Snake Pit of snakes that randomly move, eat eggs, and attack each other on contact.
speaker_order	ai2.appinventor.mit.edu/?galleryId=6554397336797184	https://community.appinventor.mit.edu/t/sort-of-reorganizing-lists/5304/10?u=abg	This sample app lets you generate and edit a speaker list from a list of attendees, showing each attendee and its optional speech(es) order.

speaker_with_highlights	https://gallery.appinventor.mit.edu/?galleryid=9a5d9c8f-4a30-4dbb-94f4-5ea83199c86b	https://community.appinventor.mit.edu/t/colorazione-del-testo-alla-lettura-di-google-voice/61078/3?u=abg	Sample text to speech with concurrent text highlighting.
special-character_remover	ai2.appinventor.mit.edu/?galleryid=5909184360742912	https://drive.google.com/open?id=1mET-BD6nEX0yLV1h9E7ACnrsxvXxByrp	This is a short code snippet to remove non-alphanumeric characters from a text value. No extensions are used.
spell	ai2.appinventor.mit.edu/?galleryid=6608923263238144	https://drive.google.com/drive/folders/16qYB800m9jnm0ZZCgHoNZOHNDPILtD4?usp=sharing	This little app spells whatever you enter out loud, letter by letter. It is meant to show how to use a clock and a global index counter for a slow loop. A deluxe highlighting feature shows the current letter, using html in a label.
splatterv2	ai2.appinventor.mit.edu/?galleryid=6337164802260992	https://docs.google.com/document/d/1TsgRjJ4oolTtgCozEDEj2GbFr56mS30jutlKs1FY6qc/edit?usp=sharing	Example of shrinking randomly appearing disks, done without Balls or Sprites, using only lists and drawn circles.

sprite_pool	https://gallery.appinventor.mit.edu/?galleryid=d2585624-9fa3-408e-a2a2-a5af10eff9c4	https://community.appinventor.mit.edu/t/is-it-possible-to-store-images-on-a-canvas-without-needing-to-make-a-sprite/157844/3?u=abg	Sample Project showing how to keep a pool of available Sprites for a game.
SQLite_Tab_Bennedum	-	https://groups.google.com/d/msg/mitappinventortest/LwJVNkyvPgU/vTaelFzFgAJ	Incomplete sample use of SQLite for autocomplete using a state name table and the Tab Bennedum SQLite extension.
states	ai2.appinventor.mit.edu/?galleryid=6522241834287104	https://drive.google.com/drive/folders/0B5Xr7cVb-tYqWVBuUWIVMWFycFU?usp=sharing	This sample app shows how to use three buttons to do binary search in an ordered list of US state names. When the final state is selected, it is replaced by its postal abbreviation.
sublist_index	ai2.appinventor.mit.edu/?galleryid=5827713545404416	https://community.appinventor.mit.edu/t/checking-4-bytes-to-mark-beginning-and-end-of-a-data-packet-on-incoming-bluetooth-data/3182/20?u=abg	Attached see a general purpose technique to locate a byte subsequence in a byte stream. The test bed works off Comma Separated Value (CSV) text

switches	ai2.appinventor.mit.edu/?galleryId=4681327081816064	https://groups.google.com/d/msg/mitappinventortest/ZLhhsELUJ8M/YfCz6N_fAgAJ	For those who don't like the nb176 AI2 switch component, here is a blocks-only procedure set that allows you to use Canvases as switches. Feel free to customize your switch canvases and procedures.
table_extractor	https://gallery.appinventor.mit.edu/?galleryid=b145d8c6-228f-433d-a96e-30e50ba4dfe4	https://community.appinventor.mit.edu/t/help-in-long-and-complicated-json-how-do-i-get-the-value-which-i-want/74538/7?u=abg	Here is a generalized header-driven table extractor, that can work off arbitrary nested list and dictionary trees.
table_to_dict	https://gallery.appinventor.mit.edu/?galleryid=f358a858-fecf-4f24-8db2-e905b5660604	https://community.appinventor.mit.edu/t/list-from-list-creation-error-cannot-parse-text-argument-to-list-from-csv-table-as-a-csv-formatted-table/100502/19?u=abg	General procedure to convert a table with headers and a unique column into a dictionary. Includes sample data and a lookup sample.
table_window	https://gallery.appinventor.mit.edu/?galleryid=d80e2258-0d0c-4d31-8ca4-c5716dcd8583	https://community.appinventor.mit.edu/t/go-scroll-yourself-no-extensions-javascript-blocks-only/60499?u=abg	A tutorial project showing how to build your own scroll bar to scroll through a data table, mixing conservative web scraping of images with rapid Label filling during a drag.

text_to_google_mp3	ai2.appinventor.mit.edu/?galleryId=5935338520051712	https://docs.google.com/document/d/1ozC7h6DEE6UC77_InTOlzTI4A-qdnvhpQc1CFB8HmwU/edit?usp=sharing	This is an experiment to see if the online Google word pronunciation .mp3 library can be used as a speech facility, using the Web and Player components. (Site is down now)
thousandizer	https://gallery.appinventor.mit.edu/?galleryId=3c48abf9-bdf8-4bf1-a735-e40182031b0d	https://github.com/AGetzler/thousandizer	Blocks to format a number in thousands.
Tidy	https://gallery.appinventor.mit.edu/?galleryId=8f998df6-698d-49a8-89da-b35816333814	https://community.appinventor.mit.edu/t/how-to-correctly-square/38594/9?u=abg	Display the structure (dictionaries, lists, constants) of an unknown object in plain multi-line text form. A useful utility.
timer_abg_paused	ai2.appinventor.mit.edu/?galleryId=4879288036098048	https://community.appinventor.mit.edu/t/problem-with-timer-app/15731/21?u=abg	This sample shows how to use TinyDB to simulate a Clock that continues to run while your app is shut down. A pause capability is added.

TinyDB_Button_states	https://gallery.appinventor.mit.edu/?galleryid=59929ef0-0398-49cc-98ca-0110898eef13	https://community.appinventor.mit.edu/t/how-can-i-store-data-in-tiny-db-of-buttons-state-switch-on-and-off/109685/3?u=abg	Sample app showing how to show multiple button states in the button text, and how to load and save those in TinyDB. For a large number of buttons, a table based approach is used along with generic blocks.
TinyDB_CloudDB_export_import	https://gallery.appinventor.mit.edu/?galleryid=e0a2fc00-07a7-4354-abeb-7eebdd1bd0a8	https://community.appinventor.mit.edu/t/tinydb-namespace-backup-and-restore-via-clouddb/105244?u=abg	These blocks export chosen TinyDB NameSpaces to CloudDB and import them back, for use transporting data from device to device.
TinyDB_Contact_Dictionaries	https://gallery.appinventor.mit.edu/?galleryid=0bdbd0e2-7c9a-47c2-a7da-14b65dec9e7d	https://community.appinventor.mit.edu/t/tinydb-functions/55826/18?u=abg	Sample contact list app storing contacts in a TinyDB NameSpace, with contact attributes stored as dictionary values.
TinyDB_Backup_Restore_CloudDB	https://gallery.appinventor.mit.edu/?galleryid=129dcf65-733	https://community.appinventor.mit.edu/t/tinydb-backup-restore/97484/11?u=abg	Here is my first attempt at exporting and importing TinyDB

	b-4719-b3f8-f0cb8147d2d1		through randomly generated CloudDB tags, by NameSpace
TinyDB_export_import	ai2.appinventor.mit.edu/?galleryId=6019316599488512	https://docs.google.com/document/d/1DaSRwpLL34MTYGMd6xn7rwKh80a-lkb-WLappAD3bPg/edit?usp=sharing	Sample blocks for those wanting to export the contents of TinyDB as text, and import it back. Good for tag values of dimensions 0,1, or 2, ranging from simple values to lists to lists of lists (tables). Add your own files to export to a file. Does not yet know about Namespaces.
TinyDB_namespaces	ai2.appinventor.mit.edu/?galleryId=5542358659760128	https://groups.google.com/d/msg/mitappinventortest/YHsf1ulZJA8/-S3eQyizAwAJ	This is a demo app to test the new TinyDB namespaces feature. Do not use blanks in your namespace names
TinyDB_NameSpace_logger	https://gallery.appinventor.mit.edu/?galleryid=d35d21ab-ad35-42cc-b729-faf028391131	https://community.appinventor.mit.edu/t/creating-profile-to-store-other-reading/54202/6?u=abg	Sample app showing off the use of TinyDB NameSpaces to store different user profiles' data in different NameSpaces, and how to keep profile names in a dedicated NameSpace. The data is some kind of

			glucose reading taken several times a day, indexed by SystemTime.
TinyDB_tree_editor	ai2.appinventor.mit.edu/?galleryId=5967796550893568	https://docs.google.com/document/d/1paDnZeGS8lrVK4E5KLs-Yu-1dVroOmah0lt1Bulq7Q/edit?usp=sharing	This is a sample way to edit a tree of items in TinyDB, where the full path to a list of sub items is the TinyDB tag, like a file structure directory. / is the root.
TinyWebDB_Undo	ai2.appinventor.mit.edu/?galleryId=6707204492230656	https://groups.google.com/d/msg/mitappinventortest/2k-usFJqgcQ/kX-PPGLIEWAJ	This is a simple example of how to use TinyDB as an Undo stack for TinyWebDB.
ToDo	https://gallery.appinventor.mit.edu/?galleryId=101e953e-a66f-458e-8fa1-4e4026f01e6c	https://community.appinventor.mit.edu/show-checkboxes-in-my-text-box/50396/6?u=abg	A simple To DO list app, with memory of Done tasks too.
total_duration	https://gallery.appinventor.mit.edu/?galleryId=40b0851b-4c5a-4d94-8b0d-d9879ae432b9	https://community.appinventor.mit.edu/sum-of-time-duration/43880/10?u=abg	Sample project to show how to accumulate intervals and calculate their total Duration, based on a list of SystemTime (start,end) pairs.
translator_component_lists	https://gallery.appinventor.mit.edu/?galleryId=5e07aa83-c869-4401-8c6f-91865c559e66	https://community.appinventor.mit.edu/translating-multiple-texts/64367/5?u=abg	This is a sample app to translate lists of components (Labels, TextBoxes, Buttons) using the stock AI2 translator component.

TwentyQuestions_v2	ai2.appinventor.mit.edu/?galleryId=6062457844400128	https://docs.google.com/document/d/1Ra-jP4pMSdcXLQ1TfcJseF8zsA5nYTEhmdhJ7QSEo8M/edit?usp=sharing	A telephone script dialog example using the 20 questions algorithm. No learning part, just a text dialog file.
Ultimate_scorekeeper	ai2.appinventor.mit.edu/?galleryId=6445569998454784	https://docs.google.com/document/d/1GLQJY9q3b8KsEAcfzPEcA-j0zWgyRjTI2Pa792znsBl/edit?usp=sharing	This is a proof of concept showing how different groups of data can be kept in TinyDB, by building up multi-part tags. The application is a score keeper for the game Ultimate Frisbee.
unlocker	ai2.appinventor.mit.edu/?galleryId=5705501348986880	https://groups.google.com/d/msg/mitappinventortest/mlUmrnQnlnQ/szFCT8yIDwAJ	This is a sample code for restricting your app's initial run to a secret time dependent formula that only you know. If you use this, change the formula.
UTC	ai2.appinventor.mit.edu/?galleryId=5967293411229696	https://drive.google.com/drive/folders/14RYa5Tm8bezVqm2fPtTSZQAJVWkd-L5T?usp=sharing	This snippet shows how to calculate UTC (GMT) from Now() using DateTime Format Z.
Valentine_Match_Game_V2	ai2.appinventor.mit.edu/?galleryId=5364471291838464	https://groups.google.com/d/msg/mitappinventortest/DS2TnwPN-YE/bxmlhmeCagAJ	This is a classic match game, for Valentines Day. Original design by Galina Ra

Web_Page_Shredder_buildservers	ai2.appinventor.mit.edu/?galleryId=6644394579329024	https://community.appinventor.mit.edu/t/a-whole-lotta-building-going-on/15070/12?u=abg	This app shows a live summary of AI2 build server statuses
Web_Page_Shredder_buildservers_V2	ai2.appinventor.mit.edu/?galleryId=6537242342719488	https://community.appinventor.mit.edu/t/the-build-server-is-currently-busy-please-try-a-gain-in-a-few-minutes/4722/25?u=abg	Want to watch MIT's build servers to see how close to capacity they have come? Run this app, It also has nice web page scraper code and a bar graph routine you can adapt for your own purposes.
Web_Page_Shredder_xrates	ai2.appinventor.mit.edu/?galleryId=5853948763963392	https://docs.google.com/document/d/1Bu0SxnnvJEmUshd9jAFuc1DTdAgHFb6Sx8Bvd1Cf4vE/edit?usp=sharing	Sample code to extract table data from a web page x-rates.com
webster	https://groups.google.com/d/msg/mitappinventortest/hjKmeDqm55M/Z5HI0WEXEgAJ	ai2.appinventor.mit.edu/?galleryId=5689081269977088	This was a simple web scraping and speech input/output sample using Webster's online dictionary.
Wheres_MIT	ai2.appinventor.mit.edu/?galleryId=6367403443486720	https://docs.google.com/document/d/1S8eZBKBvtFzafH51a3bKnw_iBVHRnjHaEY1Zbjzscb0/edit?usp=sharing	Tour the universe from The Big Bang to the Massachusetts Institute of Technology, using a List Picker, a Web Viewer, and a File.

Wheres_MIT_V2	ai2.appinventor.mit.edu/?galleryId=5677038782119936	https://community.appinventor.mit.edu/t/autosave-with-no-buttons/25958/4?u=abg	Sample of an app using no global variables at all, showing how to achieve restartability.
word_guess	ai2.appinventor.mit.edu/?galleryId=6547340125798400	https://docs.google.com/document/d/12akJeFo1FK5sbVExEDAoPc2fV6fPhZn9EIWUgA0hs44/edit?usp=sharing	Also known as Hangman, without the rope
word_picker	ai2.appinventor.mit.edu/?galleryId=5394316644843520	https://groups.google.com/d/msg/mitappinventortest/0cL1e1aBPqI/YXXi3bBcCAAJ	This example shows how to select a word from a paragraph, using only a label and some navigation buttons.
word_picker_v2	ai2.appinventor.mit.edu/?galleryId=5484203062591488	https://drive.google.com/drive/folders/1s2ruWd42YEHvKgQ7N9LwzfNLwB-K1Ie?usp=sharing	This faster version of the word picker runs in constant speed, and highlights its selection in a html label.
wordull	https://gallery.appinventor.mit.edu/?galleryId=50657648-4b83-417a-8b39-33c70955565a	https://community.appinventor.mit.edu/t/wordle-in-a-day/49688?u=abg	A Wordle clone, developed in a day.
work_days	ai2.appinventor.mit.edu/?galleryId=6198327583571968	https://groups.google.com/d/msg/mitappinventortest/EvnxCUZ0zwl/Q0OjJtRW DgAJ	This app lists a requested number of work days from a given date. A local table of holidays is kept in TinyDB, user customizable.
yaml_converter		https://community.appinventor.mit.edu/t/need-to-add-another	Project to convert structures into YAML, and

		<u>r-element-from-a-js onfile-to-my-listview /79483/3?u=abg</u>	eventually vice versa.
--	--	---	---------------------------