IncentMe Web Design Notes:

\*On AD Click

1. Identify the ad clicked
2. Show details of the ad
   1. Show price expected to pay for item(s)
   2. Show details of item(s)
3. Save button for each item
4. OnClick()
   1. if logged-in & has card on file >> save offer
   2. if logged-in & doesn't have card >> ad credit card form >> save offer

\*create an object with values for each ad

*- need a function that will load store details on pageLoad();*

\*On Normal Display/View

1. Create an Ad Timer
2. Link Ad Timer to Point Value
   1. Create an Array of values for Algorithm
      1. Need Time remaining
      2. Need Ad-Point Spectrum Of Values (e.g. 0 to 100)
      3. Set Ad Remaining Spectrum Factor/Value (0 to 50)
      4. ~~Set Store/Merchant Rank (Hard Code)~~
      5. ~~Set discount value (Hard Code)~~
      6. ~~Set Number of Ad Offers (Hard Code)~~
      7. ~~Set a Base Ad Point Figure (i.e. 100 pts)~~
      8. ~~Set Mid-Point Value (i.e. 50% of Base Ad Point Figure)~~
      9. ~~Set Base Percentage (i.e. 20%)~~
      10. ~~Calculate Base Discount Rate~~
   2. Create Activity Occurrence
      1. Ad Ledger (Total of all combined ad offers)
      2. Time of calculated Updates
      3. Base Coefficient Values (y1, y2, x1, x2, etc.)
3. Link Point Value to Ad Availability

Function Details:

* function purpose
* state of function (working, complete, refactor, review)
* for which stakeholder (consumer, advertiser, both)
* category of function (ad-display, ad-algorithm )

ID-1: CompileAdList –

* Insert descriptive text about the store into ad offers.
* (working) – needs non-manual process implemented
  + (Update) – add for loop to get values for Each Stores Object
  + (Update) – MOVED function to under ID-12 Position. Other functions weren’t executing causing points not to display within ads. (i.e. y-axis and x-axis tables where undefined).
  + (Update) – Add Point updating feature to ad offer from object values
* Advertiser
* Ad-display

ID-2: calculateTime –

* Calculate time left before an ad offer/campaign expires
* (working) – need to update the adEndTime variable to get from store object
  + (Update) – added for loop to get values for each object
  + (Update) – added adCreateTime property to “stores” Object
  + (Update) – added adEndTime property to “stores” Object
* (review) – possibly break up internal methods
* Advertiser
* Ad-display

ID-3: getAdPointValue –

* Calculates the “**BASE AD POINT”** value used for converting discount percentages
* (working) – need to associate a unique adBasePercentageDiscountAverage for store type [a]
* (working) – need to associate a unique adPointMidPercentage for store type [b]
* (working) – need to update store object to have storeType property & values for [a] & [b]
* Advertiser
* Ad-algorithm

ID-4: getXAdPointSpectrum –

* Generate discount percentage value for horizontal (x-axis) row; (used for discount percent value set by advertiser)
* (working) – need x-axis variable to be set to at “.0001” or less. << lower decimal values cause browser to freeze
* Advertiser
* Ad-algorithm

ID-5: getYAdPointSpectrum –

* Generate vertical (y-axis) column for ad-ledger; (used for the current/live discount percent value of all ad offers)
* (working) – need y-axis variable to be set to at “.0001” or less. << lower decimal values cause browser to freeze
* Advertiser
* Ad-algorithm

ID-6: getYPivotPointValues –

* Generates vertical (y-axis) values from a mid/pivot point; values are used as a coefficient for the calculated yAdPointSpectrum & advertisers set discount percentage
* (working) – need different y-Axis pivot points; change by store type
* Advertiser
* Ad-algorithm

ID-7: getBaseDr –

* Calculates the adjusted base discount rate from the pivot point for y-plus and y-minus.
* (working) – variable baseMidStartValue needs to be different for each store type; its associated by store type
* Advertiser
* Ad-algorithm

ID-8: combineDr –

* Combines baseDrPlus & baseDrMinus array into a single array
* (review) – possibly review to add into one function with function ID-7
* Advertiser
* Ad-algorithm

ID-9: getFinalAdLedgerTable –

* Generates & calculates a final x & y table for all possible values??
* (review for removal) – y-axis and x-axis are not multiplied by each other; generating a table outputs incorrect values. (leaving in case its use was overlooked)
* Advertiser
* Ad-algorithm

ID-10: getTotalAdOffers –

* Calculates the discount percentage total & ad offer total in circulation; returns the discount average percent. The percentage value returned is the active y-axis column value. The index value of the y-axis position is then used to get the active discount-rate coefficient.
* (working) – need to calculation to include/be specific to store Type
  + (update) – remove arguments; store object has been passed in as object. Arguments was resulting in NaN due to each property of stores object not being an integer.
* Advertiser
* Ad-algorithm

ID-11: getOriginatingDiscountPercent –

* Calculates the ad-point conversion rate for a single percentage point; stores discount percentage is multiplied by the discount-rate coefficient
* (working) – if changes are made to yAdPointSpectrum, yIncrement NEEDS to be same. Otherwise, problems will occur with using index values of other tables.
* (working) – can this function be used within each stores Object? Once a value is received, it processes and determines the discount convert rate
  + (refactor) – removed arguments; updated return statement; upconvert point values; upconvert of BaseDr is optional and will depend on how large y-axis and x-axis values are.
* Advertiser
* Ad-algorithm

ID-12: getStoreAdPointValue –

* Uses the stores discount percentage, multiplies it by the calculated/current discount convert rate. The returned value is the ad offers active point value and displayed within the ad offer(s).
* (working) – Insertion is hard-coded. Needs to change to be more flexible and work for all stores.
  + (Review) – Remove function; refactored into function ID-1
* Advertiser
* Ad-algorithm

ID-13: getXYAdActivity –

* Sets the Y axis values for multiple ad-activity tables; Used for further manipulating ad-point system.
* (review) – review for improvements
* Advertiser
* Ad-algorithm

ID-14: getAdActivityTable –

* Generates the X values and calculates the multiplier values for the ad-activity table; uses the x1 values as multiplier and ad-activity Y-values to result in modifier values.
* (working) – need tables that are created associated to rank levels
* Advertiser
* Ad-Algorithm

ID-15: adOccuranceFactor –

* Gets the active multiplier from the generated ad-activity table(s); Returns the modifier value that’s used to further adjust point value of offer.
* (review) – review for improvements
* Advertiser
* Ad-algorithm

ID-16: setMTableAdver –

* An object for holding the measurement conditions for each rank level. This is currently manually defined.
* (working) – work on formula that’s capable of modify values using the users previous activity as the base.
* Advertiser
* Rank-algorithm

ID-17: calculateObjectValues –

* Used as a function to gather data from multiple (4) objects; calls function ID-18 passing values as parameters
* (working) – requires significant rework, but works in its ability to get values and input
* (working) – correct to pass object values into arrays; objects are causing multiple passes resulting in output errors (if statements not directing data as intended)
* Advertiser
* Rank-algorithm

ID-18: calculateStoreValues –

* Collects rank measurement actions, determines whether measurement action satisfies minimum condition, adds rank points to advertiser object
* (working) – works in its ability to get values and input; requires complete rework
* (working) – too many nested for loops; passing objects is causing multiple output bugs (if statements not directing data as intended; one click causes all rank levels of a matched measurement event to execute)
* (working) – measurement param values currently assigned manually; Need to get correct values through loop or some other method
* Advertiser
* Rank-algorithm

Function Details:

* function purpose
* state of function (working, complete, refactor, review)
* for which stakeholder (consumer, advertiser, both)
* category of function (ad-display, ad-algorithm, rank-algorithm)

\*possibly create a storeType object? Inherit the values by stores associated type.