New Hire Guided Tasks

+ Use <http://www.mivascript.com> for function references

+ Main info resource: <https://docs.miva.com/>

+ Utilize the allmvtdos files and look up the functions in our [LSK](https://www.miva.com/support/downloads)

1. Miva Search & Facets

**Task:**

- Set up Miva search preview & facets on your dev store.

**Example Site:**

- <https://www.artificialplantsandtrees.com/full-christmas-trees.html>

**Reference Resources:**

- <https://docs.miva.com/how-to-guides/search-preview-improved-search>

- <https://docs.miva.com/videos/videos-runtime-search-search-preview>

- <https://docs.miva.com/how-to-guides/facets-category-search-filtering>

- <https://docs.miva.com/how-to-guides/combination-facets>

2. Create a "dev" conditional based on your customer account and show a message.

3. Show a message for only logged in customers

4. Turning on URI management

**Task:**

- Review the URI management videos & docs again if needed

- Set up URI management in your store & generate Canonical URIs for all of your pages/categories/products

- Ensure that URIs are not changed/updated when changes are made to products.

- Make a change to your URI structure for categories, and then generate the new URIs while having the old URIs 301 redirect via URI management to the new Canonical URI.

- Pick a product, assign it to two categories, then generate two additional URIs for it that include the category name in the URL path. Example: domain.com/CategoryName/ProductName.html

- Review how to output the URIs in the template code.

- Create a blank page and output the following using our URI output variables

+ URL to the home page auto detecting the protocol

+ URL to the BASK page with the secure https protocol

+ URL Account Creation page with https and the parameters "wholesale=1" in the query string.

+ Output Canonical URI for a specific product. Fallback to "Normal" URI if no canonical. Fallback to long URL if neither canonical nor normal are available.

- Hint: Try dumping and testing the following mvt:do functions for this: URI\_Load\_Product\_Canonical(), URIList\_Load\_Product(), Store\_Product\_URL().

5. Cookies Writing/Reading (Multiple Languages)

**Task:**

- Output all current cookies to the page so you can view them in the page content using Miva template language.

- Hint: See global variable "g.request\_cookies"

- Create a cookie that expires in 1 day using Miva functions

- On another page, read the cookie and if found valid, output a message using all Miva functions.

- Now to the same process of creating a cookie, reading it and outputting a message using JavaScript.

**Reference Resources:**

- <https://docs.miva.com/code-samples/set-cookie>

6. Google Product Feed & Scheduled Task to generate feed

**Task:**

- Create a Google Product Feed

- Only load available products. Google shopping is for public shoppers, so we should only show them public facing products.

- See Google Shopping guidelines for required fields.

- Create custom fields for non-standard Miva fields like "Brand" or "GTIN", "Google Product Category", etc.

- Create new image type to be used in feed for the product image. This should be a newly created image type just for this feed, not the default “main” image type.

- Load your feed with variants so you can output the variant specific products in the feed instead of the individual master. For master products that do not have variants generated for them, just output the master product in the feed.

- Create the ability to omit certain products from the feed. Hint: Use custom field and <mvt:exit/>. Since our feeds have an iterator template, and doesn't actually use <mvt:foreach> tags, the exit tag will stop that iteration.

- Create a custom field for "Feed Description" that is used. Make a fallback that if the custom field is empty, we use the default product description, but we strip HTML from it. Hint: look at miva\_html\_strip() MivaScript function.

- For variant products, load the variant URL. This is the master product URL, with the param "variant\_id=XXX" where "XXX" is the variant's ID. This this cause that variant's attributes to be pre-selected when you load the product's page. One way the variant ID can be obtained is by running the mvt:do function "<mvt:do file="g.Module\_Library\_DB" name="l.success" value="ProductVariantPartList\_Load\_Part(variant\_product\_id, productvariantparts var)" />". Note: The Variant Product's ID is different than the Variant's ID. Ask for clarification on this.

**Reference Resources:**

Google Docs

- <https://support.google.com/merchants/topic/7438667?hl=en&ref_topic=7259125>

- <https://support.google.com/merchants/answer/7052112?hl=en>

- <https://support.google.com/merchants/answer/6324436>

- <https://www.google.com/basepages/producttype/taxonomy-with-ids.en-US.txt>

Miva Docs

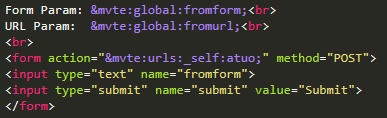
- <https://docs.miva.com/reference-guide/feeds>

- <https://docs.miva.com/how-to-guides/template-feed>

7. How do global variables get populated from Forms/Query Params

**Task:**

- Create a blank page template, and in the template code the following:



- Visit the page and enter a value in the input field and submit the form to see what happens with the "Form Param" output.

- Now instead, add in the URL of your page the query params of "fromurl=XXXX" where XXXX is a random value of your choosing.

8. Set up ajax add to cart functionality

**Task:**

- Choose either to add this feature to the “Quick View” on a category page (if currently using a theme that supports this) or create a new blank page that loads in a product with attributes, and then has an ajax add to cart feature.

- Take in to account when not all required attributes are passed

- Display some sort of success message or indication that the product was successfully added.

9. Variant change ajax call to pull in description from a "custom-calls" page

**Task:**

- Create a page called "custom-calls"

- Create logic on "custom-calls" page that loads in a variant product's description.

- Hint: Check mvt:do function ProductList\_Load\_Variant()

- Conditionlize the logic on "custom-calls" to only run when certain params are passed to the page so that "custom-calls" can be used as a central spot for other custom functionality calls as well.

- On the product page, run JavaScript variant change function (see intro helper file) and make the ajax call to your "custom-calls" page.

- Update the product page description with the newly loaded variant description any time the variant is changed.

10. Example of creating ReadyTheme product listing that is displayed with our slick slider js code

**Task:**

- Choose the home page, a category page, or product page template

- Create a ReadyTheme product listing and assign 5+ products to the list (ensure all products have images).

- Output the product listing in the page template, but tie into the slick slider to create a product carousel.

- Generally, you'll find example of the slick slider code in our scripts.js files.

- The Levels theme should have a working example of a slick slider already as part of the SFNT homepage.

**Example:**

- See the "Related Products" display: <https://www.planttherapy.com/peppermint-essential-oil>

11. Take a MM/DD/YYYY format value and check if unix time is greater or less than current time.

**Task:**

- Pretend that the value comes from user input or somewhere that you only have a MM/DD/YYYY formatted date string.

- Split the string to obtain each part of the date.

- Use our system variable to obtain the current unix time

- Check out MivaScript function mktime\_t()

**Use Case:**

- You might utilize this type of functionality by creating some logic where a merchant can specify a specific date where some custom logic/content begins to display on their site once it reaches that date.

**Reference Resources:**

- <https://docs.miva.com/videos/dev-system-variables>

12. Hide a specific image type from product page image machine

**Task:**

- Work in your main PROD page template.

- Create an additional image type aside from "main" if you don't have one currently created

- Add an image for that type to a product in your store

- Customize image machine logic so that it skips any images it finds with that image type.

- Hint: The "type\_code" holds the image type value in the image machine JavaScript function ImageMachine.prototype.ImageMachine\_Generate\_Thumbnail

- Hint: check your console! We want to avoid any errors occurring. Elements may need to be removed after the fact.

- Ask for guidance as needed on this task. It is a tricky one.

13. Meta refresh vs output header reload

**Task:**

- Create a page that does a while loop with a counter 5 times.

- Each turn of the loop, make the page refresh. You'll use refreshes like this to ensure pages do not time out when creating custom reports/feeds/etc.

- Create logic that will use a Meta Refresh if the page is being run manually through a browser, and use an Output Header to reload if the page is being run through a cron job scheduled task.

- Hint: We can determine this by passing a parameter in the URL for when being run by a cron job scheduled task.

- Use miva\_output\_header() and Meta Refresh HTML tag.

- When the loop is complete, output a message on page and dynamically show how many iterations were run (in this case it should show, 5).

**Examples:**

- Example Usage: <mvt:eval expr="miva\_output\_header( 'Location', g.my\_next\_url )" />

- Example Usage: <meta http-equiv="refresh" content="0;URL='&mvt:global:my\_next\_url;'" />

14. mvt:call

- This item is informational - mvt:call will be used in next task.

- See helper file for examples

- The mvt:call makes a server sided call to the specified URL, so it does NOT share the same customer sessions. For example, if you have an mvt:call running for a customer front-end experience, and you try to grab some basket data, the data you get back will NOT be related to that customer's session. You would need to include the Session\_ID param with the customer’s session to make the endpoint load the same session. Example:

<mvt:call action="'https://www.domain.com/mm5/merchant.mvc?Action=ADPR&Product\_Code=testproduct&Quantity=1&Session\_ID=' $ g.session\_id" METHOD="'GET'"></mvt:call>

- <https://docs.miva.com/template-language/mvtcall>

15. Take Miva structure > Convert to JSON > Send to PHP

**Task:**

- Note: This will be built with the intention of running it for only one order at time.

- Create a new Order Batch Report

- Utilize the Max-json-encode module to convert all of the order data to JSON data (see intro helper file).

- Run an mvt:call to send the JSON encoded order data to a PHP file

- If the PHP page is hit without the required POST data, kill the page.

- In PHP file, grab the sent POST data and json\_decode the data.

- Hint: If sending a variable like "l.settings:orderjson" through the mvt:call, you can grab that POST data in the PHP file like: $\_POST['settings:orderjson']

- Output the order ID, customer billing information, and some basic item data cleanly on page, each on their own lines, to be used as the mvt:call return value.

- Back in the batch report, display the result of the mvt:call's return value.

- NOTE: Generally you should concatenate the returned s.callvalue to a new variable within the mvt:call tag. The mvt:call tag works kind of like a loop, where it may return multiple chunks of data, so concatenating s.callvalue will give us the complete set of data after the closing mvt:call tag.

**Reference Resources:**

- <https://github.com/mghweb/miva-max-json-encode>

- <https://docs.miva.com/template-language/mvtcall>

- <https://docs.miva.com/code-samples/mvtcall-pass-custom-headers>

16. WordPress Integration

**Task:**

- Follow all of the steps in the following guide to set up a blog in the /blog/ directory on your dev store and follow all setups to pull in the Miva Header/Footer/Basket etc.

- Work with support have a database created for your dev store in order to connect your WordPress installation to (ask for help on this).

- <http://blog.merchant.local/?p=1490>

17. Batch report that generates a feed file of order data

**Task:**

- Create an order batch report

- Loops through all of the order data and generate a CSV file on your server

- Utilize a variety of products in your order: Product with attribute and option data with prices & Master product with no attributes

- The CSV should contain the following headers and data: Order#, Billing First Name, Billing Last Name, Items Subtotal, Shipping Cost, Tax, Other Charges, Order Total

- Create another batch report that is sales report for all products that were part of a specific category

**Reference Resources:**

- <https://docs.miva.com/videos/writing-csv-file>

- <http://www.mivascript.com/item/file_append.html>

- <http://blog.merchant.local/?p=4180> <<< Review this blog post as it discusses important information on where/how to more securely store certain feed files and data.

18. User Price Groups

**Task:**

- Set up each of the different price groups under the "Type" dropdown.

- Successfully run and test each discount/promotion on your store to confirm they are correctly working before moving onto the next price group.

- Test how Exclusions work, Qualifying restrictions, and basket restrictions.

- Test some of the different eligibility settings, in particular set up a coupon based promotion.

- Download and install the Generate Coupons module off the app store: <https://apps.miva.com/generate-coupons.html>

- Utilize the Generate Coupon functionality to create and give customers a unique coupon code whenever they first create an account

**Specific Challenge:**

- Make it so that when a specific coupon promotion is applied, the user is no longer able to receive a normal basket or product discount that would otherwise be applied.

- Create a promotion where when you buy any two of a pool of 3 qualifying products, then the customer receives $5 off their order.

- Create a Gift Certificate product (Using Miva's native gift certificate features) and create a promotion where when a coupon is applied, get $15 off your order, but none of that discount can be applied towards gift certificates.

**Reference Resources:**

- Look up various price group info on <https://docs.miva.com/>

- <https://docs.miva.com/reference-guide/account-credit-gift-certs>

19. Availability Group usage

**Task:**

- Create an Availability Group on your store called "Wholesale"

- Create a custom catalog or "purchase list" of products that organizes all products that this group will get specials on.

- Make it so this Wholesale group can only purchase with a credit card, and they receive a special wholesale flat rate shipping method

- Create an order custom field for "Wholesale ID" that shows a text field for this group during checkout that can be populated with a value and saved to the order when purchased.

20. Business Account Usage

**Task:**

- Create a business account for Wholesale customers. Now utilize a business account to apply the Wholesale Availability Group settings and a wholesale price group, rather than having to assign every customer to each group.

21. Sort your OSEL shipping method results by price lowest to highest

- Check out MivaScript function QuickSortArray()

22. Product Multi Add

**Task:**

- Create a page that has a multi add form, where users can add multiple different products to the cart with one form submission.

- <https://docs.miva.com/how-to-guides/setup-miva-multi-product-add>

23. Show/Hide "Other State/Province" Field

**Task:**

**-** If you are currently using a ReadyTheme Framework, save/export the framework and switch back to our default framework.

- Create logic on one of the customer information sections (like OCST) where the state is set to "Outside US" in dropdown it will hide/show the state text input

- You can review the following example logic: <https://gist.github.com/steveosoule/4f78bbcdd53c69c22dbc>

24. Add additional validation on customer fields

**Task:**

- Add JavaScript based validation (can use jQuery) to OCST page to validate all required attributes. Create a user friendly experience.

- Also add an "I agree to terms and conditions" checkbox that is required, and also save their selection to an order custom field.

25. Custom Continue Shopping Button

**Task:**

- Create a custom "Continue Shopping" button on the BASK basket page that will either take the customer to the last category they viewed, or if they have not viewed a category take them to the home page

- Utilize basket custom fields to store & read/write the data

- <https://docs.miva.com/template-language/custom-field-reference-documentation>

26. jQuery Zoom Setup

**Task:**

- Set up jQuery Zoom on your product page or alternate display product page

- Refer to this how-to guide: <http://blog.merchant.local/?p=4130>

- The name of each code section (at the bottom of each) tells where the code should be place. For example, the JS should be in the image machine template of the PROD page. Ask if clarification is needed.

27. Facebook Login

**Task:**

- Setup Facebook Login on your store

- Follow the instructions located in the following link: <http://blog.merchant.local/?p=1527>

- You can use your own Facebook account to set up the app under or ask for our company shared account and then remove the app afterwards.

- NOTE: Read comments at the bottom. Guide was written in 2012 and there are some updates that have been commented

28. Customer Custom Fields

**Task:**

- Update your Account Creation page and Account Edit page to allow customers to provide you their Birthday & Gender

- Save these values to the customer’s account

29. Import mapping

**Task:**

- Review all sections of Data Management docs: <https://docs.miva.com/reference-guide/data-management>

- In particular review the Manually Assign columns method. This is a great way to create custom imports for merchants and ensure the data is mapped properly.

- When importing data to a store, it is a good idea to almost always create a new import and manually map the data fields to ensure they import correctly. Test importing the data on a small batch initially. Always export a backup of the complete data before running import in case anything goes wrong.

- <https://docs.miva.com/reference-guide/import-settings#saving-import-options>

30. XML Provisioning Import/Export

**Task:**

- Create a page or feed that loops through your store's products, and outputs the XML for all attribute and options data for each product.

- On the same page, utilize mvt:do functions to pull in all attribute template data in the store and outputs XML for them.

- For regular attributes, check how the PROD page template pulls in a product’s attribute data.

- For attribute templates, use mvt:do functions: AttributeTemplateList\_Load\_All(), AttributeTemplateAttrList\_Load\_Template(), AttributeTemplateOptionList\_Load\_Attribute().

- Hint: Check XML tags ProductAttribute\_Update, ProductAttributeOption\_Update, AttributeTemplate\_Update, AttributeTemplateAttribute\_Update, AttributeTemplateAttributeOption\_Update

**Reference Resources:**

- <https://docs.miva.com/xml-provisioning>

- "master\_provide.xml" file in the "provisioning" folder in the LSK

31. Miva Array Functions

**Task:**

- Create a Miva array with the following values, in this specific order: One, Two, Three, One, Five, Four, Six

- Delete the records with the values of "One" and "Four" from the array.

- Insert the value "Ten" at the end of the array, and insert the value "Seven" as the 3rd index.

**Reference Resources:**

- Review the examples located here to understand how example functions work: <https://snippets.miva.com/id/1528934064/1>

- Possible Functions: miva\_array\_find(), miva\_array\_search(), miva\_array\_deserialize(), miva\_array\_delete(), miva\_array\_insert(),miva\_array\_insert\_var(), miva\_array\_elements()

32. Admin Order Management

**Task:**

- Create a new order with you as the customer via the admin.

- Add some items. Add some with attributes.

- Add a shipping method charge, ensure there is tax on the order as well.

- Give the order a discount

- Create a shipment of that entire order. Mark it as shipped and put in a fake tracking number. Pretend it is shipping via USPS and has a tracking link.

33. Shipping methods & rules

**Task:**

- Turn on UPS Ready Tools for real-time UPS rates and enable some methods (use the credentials found here, <https://wiki.dev.mivamerchant.com/index.php/UPS_Ready(R)_Tools>

- Create a Free flat rate shipping method.

- Create a "Freight" flat rate shipping method

- Use the shipping rules so that the only time the free method shows is if the order total is above $100 and the person is in the state of California.

- Use shipping rules so that the Freight method is the only method that id displayed if the total order weight is over 100lbs.

34. Order Encryption Keys

**Task:**

- Turn on order encryption under Store Settings.

- Enabled and set up Credit card with simple validation payment module.

- Place an order using a visa card # 4111 1111 1111 1111.

- Look in the admin under manage orders and see what order encryption does on this screen.

35. Admin User Groups

**Task:**

- Create a User Group that has limited access to the admin. The user should only be able to access manage orders.

- Assign a user to this group. Login as this user and see how the admin changes

36. Add a state sales tax of 9.75 for all California customers

37. Customer Field Changes

**Task:**

- Make the fax number hidden and the phone number optional.

- Make the billing address the primary fields. NOTE: When this is done it may (Based on page settings: See "Prevent Automatic Updates to This Template" checkbox) change the template on every page that the customer fields’ item is assigned, so preview that page once this is changed and make sure the layout still works.

38. Demo of order confirmation email and how global variables and custom order fields react

- This will be a demo that we'll walk you through. Please ask once you reach this step.