****

**Custom JavaScript for Marketers**

**Dynamic value tracking with custom javaScript**

**When the value shows on the conversion page**

* **GTM Variable builder chrome extension method (Automatic)-**

<https://chrome.google.com/webstore/detail/gtm-variable-builder/feeboihdgpananoagfmbohoogoncndba?hl=en>

* **DOM scraping method (manual)**
* **Replacing , with .**
* **Removing $/ or anything from value**

**Replace script**

replace(/[^\d.]/g, '')

**parseFloat(variable code) - string to number method**

**When the value does not show on the conversion page**

* **The code will be placed in cHTML tag to push dynamic value**

<script>

window.localStorage.setItem('fullName', {{cJS - fullName}});

window.localStorage.setItem('email', {{cJS - Email}});

window.localStorage.setItem('value', {{cJS - value}});

</script>

* **The code will be placed in a custom JS variable to pull the dynamic value**

function(){

return window.localStorage.getItem('value');

}

**Form field tracking - dynamic value**

**1st code**

function(){

return Copy js path.value

}

**2nd code**

function(){

return document.querySelector('input[name="email"]').value

}

**3rd code**

function(){

return document.getElementsByName('shipping\_email')[0] ? document.getElementsByName('shipping\_email')[0].value: '';

}

**Dynamic Currency Tracking**

function() {

var priceElement = document.querySelector('.woocommerce-Price-currencySymbol');

var currencySign = priceElement ? priceElement.textContent.charAt(0) : '';

var currencyCode;

switch (currencySign) {

case '£':

currencyCode = 'GBP';

break;

case '€':

currencyCode = 'EUR';

break;

case '$':

currencyCode = 'USD';

break;

case '৳':

currencyCode = 'BDT';

break;

//add more curreney bellow the comment

default:

currencyCode = '';

}

return currencyCode;

}

**Code for pulling product id (not compatible with all site)**

function(){

return document.querySelectorAll("[data-product-id]")[0].dataset.productId;

}

**Page Reload Script**

function(){

return performance.navigation.type === 1;

}

**Adding CSS Class and ID with GTM**

**Add Class with GTM**

copyjspath.classList.add('mystyle')

**Add ID with GTM**

copyjspath.setAttribute('id', 'test 4')

**Transaction ID Code from URL:**

function() {

var ahad\_url = {{Page URL}};

var transactionID = ahad\_url.split("/")[5];

return transactionID;

}

**Product Quantity**

function() {

var getDataFromArray = {{dlv-Cart Items}};

var initialValue = 0;

return getDataFromArray.reduce(

function(accumulator, currentValue) {return accumulator + currentValue.quantity},

initialValue

);

}

**Total Value from array**

**When the price in string mode**

function() {

var getDataFromArray = {{dlv-Items}};

var initialValue = 0;

return getDataFromArray.reduce(

function(accumulator, currentValue) {return accumulator + parseFloat(currentValue.price)},

initialValue

);

}

**When the price in number mode**

function() {

var getDataFromArray = {{dlv-Items}};

var initialValue = 0;

return getDataFromArray.reduce(

function(accumulator, currentValue) {return accumulator + currentValue.price},

initialValue

);

}

**Array Builder code**

**Single Product-**

function(){

return [{

item\_name: 'test product',

price: 40,

item\_id: 65,

quantity: 4

}]

}

**Multiple Products**

**Code -1**

var items = [];

document.querySelectorAll(".cart-item").forEach(function(item){

var \_item = {};

\_item.item\_name = item.querySelector('.cart-item-header').innerText;

\_item.price = parseFloat(item.querySelector('.c-price').innerText.replace(',', ''));

items.push(\_item);

});

console.log(items);

**Code - 2**

var items = [];

var item = {};

var productList = document.querySelectorAll('.cart\_item');

for (var i = 0; i < productList.length; i++) {

item.item\_name = productList[i].querySelector('.product-name').innerText;

item.price = parseFloat(productList[i].querySelector('.product-price').innerText.replace(/[^\d.]/g, ''));

item.quanity = parseInt(productList[i].querySelector('.input-text').value);

items.push(item);

}

console.log(items);

**Set and Retrieve Array ( Local storage )**

Set = JSON.stringify()

Retrieve = JSON.parse()