**Dynamics**

**Variables**
Variables are used to dynamically insert text into site tags (like an eVar or event). Variables are wrapped in `%...%` syntax.

<table>
<thead>
<tr>
<th>Category</th>
<th>Action</th>
<th>Label (Optional)</th>
</tr>
</thead>
<tbody>
<tr>
<td>blog post</td>
<td>$(this.title)</td>
<td>path: %URL%</td>
</tr>
</tbody>
</table>

- use selected element `%this.attrib%`
- use child element `%target.attrib%`

Dynamically insert the title and inner HTML for: `<a title="link" href="'">Hello!</p></a>`

- title `%this.title%`  
- Output `link`  
- inner HTML `%target.innerText%`  
- Output `Hello!`

Other example uses:
- href `%this.href%`
- class `%this.className%`
- alt `%this.alt%`
- src `%this.src%`
- inner text `%this.@text%`
- custom attribute `%this.getAttribute(name)%`

**Data Elements**
Data Elements are an alternative way of setting a variable. These are most useful when you use value(s) across multiple rules.

- using data elements in a rule `%Element Name%`
- set a data element `_satellite.setVar("Element Name",value);`
- get a data element `_satellite.getVar("Element Name");`

**Cookies**
Cookies store values over a certain period of time.

- They are similar to Data Elements, but instead allow you to set a cookie memory in days (integer).
- set a cookie `_satellite.setCookie("Cookie Name",value,days);`
- read a cookie `_satellite.readCookie("Cookie Name");`  

**Comp Sci**

**JavaScript**
Below are some of the most commonly used methods in JavaScript. Note that many DTM implementations don't even use JS, but just in case...

**JavaScript Regular Expressions**

- var RegEx = /pattern/mod
- function global matching
- /g
- case insensitive
- /i
- single line mode
- /s
- multi line mode

**JavaScript Selectors**

- get element by id `getElementById("id")`
- get elements by class `getElementsByClassName("class")`
- get elements by tag `getElementsByTagName("tag")`

**JavaScript Logic Tests**

- if statement
- else if statement
- switch statement
- default:
- for
- next loop

**jQuery**
Below are some common jQuery methods. If available, consider using Sizzle as a selector.

**Core Patterns**

- match any character *
- 0 or more of previous expression +
- 1 or more of previous expression ?
- 0 or 1 of previous expression
- start of string $`
- end of string |
- "OR" match on both sides
- explicit quantifier notation
- explicit character set match
- NOT one of the characters in set
- special character literal

**Logic**

**Diagnostics**
DTM has useful functions built into the platform that help QA tags. The code below can be used in functions or typed directly into your console.

- enable staging library in production
- localStorage.setItem(sdsat_stagingLibrary,true);
- enable debug mode (useful to QA rules)
- _satellite.setDebug(true);
- hide browsing activity from DTM
- localStorage.setItem('hide_activity',true);
- cross-browser compatible console.log()
- _satellite.notify('text to display',[1-5]);

**Regular Expressions**
Regular Expressions (RegEx) is a method of matching that can be used within DTM (and JavaScript and others) to select groups of text.

**Examples**

- matches a or b or c
- matches any letter from a-z (lowercase)
- matches letter from A-Z (uppercase)
- matches any number from 0-9
- matches 3 of preceding expression
- matches 3-9 of preceding expression
- matches 3 or more

by jimalytics.com

@jimalytics