

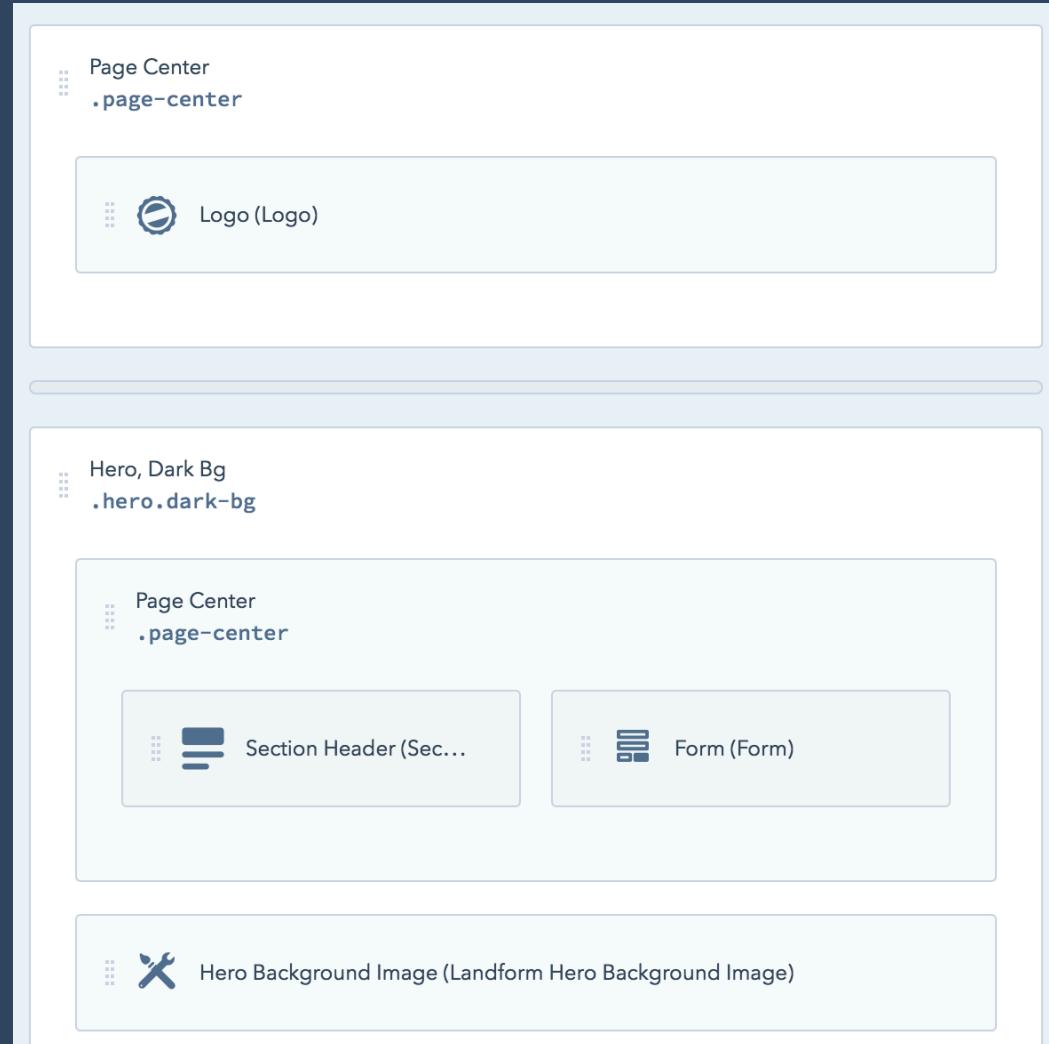


HubL

WHAT IS HUBL,
AND WHEN WOULD
YOU USE IT?

TEMPLATING LANGUAGE

A simple language that gives quick, standardized access to data without having to use a full-fledged programming language like PHP



Drag and drop templates

You won't need to write markup code because the drag and drop builder writes it for you.

Modules

HubL is an important part of building custom modules.

The screenshot shows the HubSpot CMS interface for editing a module. The main area is a code editor with three tabs: HTML + HUBL, CSS, and JS. The HTML tab contains the following HubL code:

```
11 <div class="hs-accordion" {%- if module.max_1_open %}data-sync="true"{%- endif %}> {# << Section 1 #}
12 {%- for accordion in module accordions %} {# << Loop through the accordions group #}
13 <div class="hs-accordion__item" aria-expanded="{% if accordion.open_by_default %}true{%- else %}false{%- endif %}">
14 <button>
15 {{ accordion.title }}
16 <span class="hs-accordion__item-icon"></span>
17 </button>
18 <main>
19 <div class="hs-accordion__item-content">
```

The CSS tab contains the following styles:

```
1 .hs-accordion__item {
2   margin: .5em;
3 }
4 .hs-accordion__item button {
5   display: block;
6   width: 100%;
7   font-size: 1em;
8   padding: 1em;
```

The JS tab contains the following JavaScript:

```
6 function buildAccordion(el) {
7
8   var accordion = el,
9     accordionItems = accordion.children,
10    oneAtTime = accordion.getAttribute('data-sync'),
11    hideAllItems = function(ex) {
12      /* Loop through all items and hide them */
```

On the right side, there's a sidebar titled "Fields" which lists the fields for the "Accordions (Field group)" section:

- Label (Text) - title
- Content (Rich text) - content
- Open by default (Boolean) - open_by_default

At the top right of the editor, there are buttons for "Make available for templates", "Preview", and "Publish changes".

```
157      {> end_widget_block <}
158    </div>
159    <!--end layout-widget-wrapper -->
160  </div>
161  <!--end widget-span -->
162 </div>
163 <!--end row-->
164 </div>
165 <!--end row-wrapper -->
166 {% set isCustomer = ( 144165 in contact.list_memberships or account.account_status__c == "Active" ) %}
167 <div class="row-fluid-wrapper row-depth-1 row-number-2">
168   <div class="row-fluid">
169     <div class="span12 widget-span widget-type-cell" style="" data-widget-type="cell" data-x="0" data-y="0" data-z="0">
170       <div class="row-fluid-wrapper row-depth-1 row-number-3">
171         <div class="row-fluid">
172           <div class="span12 widget-span widget-type-raw_html" style="" data-widget-type="raw_html">
173             <div class="cell-wrapper layout-widget-wrapper">
174               {% widget_block raw_html "module_1478541641428100820" overrideable=false, cell=false, component=true %}
175               {% widget_attribute "value" %}
176               {% text 'non_customer_banner_heading' label='(non-customer lifecycle) banner heading' %}
177               {% text 'non_customer_banner_subheading' label='(non-customer lifecycle) banner subheading' %}
178               {% image 'non_customer_banner_image' label="(non-customer lifecycle) banner image", ext="jpg" %}
179             <div class="mql-sales__header header-1 header--pricing">
180               <div class="header__bg-img" style="background-image: url({{ widget_data.non_customer_banner_image }})">
181                 <div class="header__content">
182                   <h1 class="header__headline header__headline--1">{{ widget_data.non_customer_banner_heading }}
183                   <p class="header__subhead header__subhead--1">{{ widget_data.non_customer_banner_subheading }}
184                 </div>
185               </div>
186             </div>
187             {% end_widget_attribute %}
188             {% end_widget_block %}
189           </div>
190         </div>
191       </div>
192     </div>
193   </div>
194 </div>
```

Coded templates

Coded templates, especially blog templates, use HubL extensively.

HubL and Jinja

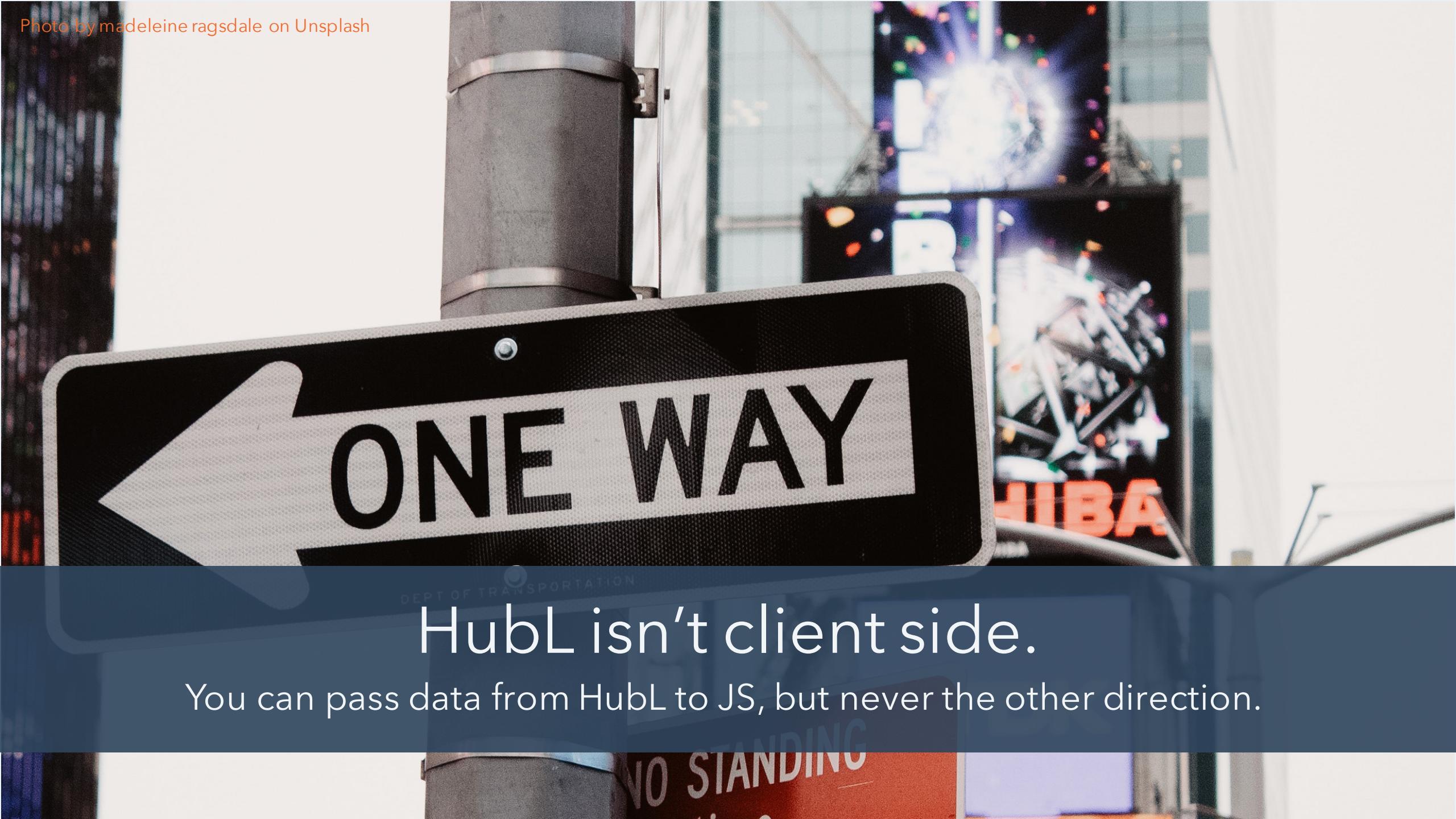
HubL is essentially a subset of Jinja,
but it doesn't support all of Jinja and
includes a lot of its own markup.



Jinja



HUBL ISN'T A PROGRAMMING LANGUAGE.



HubL isn't client side.

You can pass data from HubL to JS, but never the other direction.

HubL is a fairly deep topic.

Get familiar with the documentation,
developer tools, and community.



FUNDAMENTALS OF HUBL



THIS CONTENT ASSUMES EXPERIENCE WITH
COMMON PROGRAMMING ELEMENTS.

```
{% set my_variable = "a statement does something" %}

{{ an_expression_evaluates_to_a_value }}

{# COMMENTS AREN'T RENDERED CLIENT SIDE #}
```

Statements, expressions, and comments

```
{% set my_variable = "I'm a string!" %}
```

```
{{ my_variable }}
```

```
{# OUTPUT: I'm a string! #}
```

Setting and retrieving a variable

```
{% set my_list = ["string 1", "string 2", "string 3"] %}  
  
{{ my_list[0] }}  
  
{# OUTPUT: string 1 #}
```

Creating a list and retrieving the first value

```
{% set my_dictionary = {"name":"bobby", "age":26, "eye_color":"blue"} %}

{{ my_dictionary.eye_color }}

{# OUTPUT: blue #}
```

Creating a dictionary and retrieving a value

```
{% macro add_numbers(n1, n2) %}  
    {{ n1+n2 }}  
{% endmacro %}  
  
{{ add_numbers(4, 5) }}  
  
{# OUTPUT: 9 #}
```

Creating and calling a macro

```
{% set friends = ['Joe', 'Rachel', 'Doug', 'Winston'] %}

{% if friends is iterable %}
    <p>My friends are {{ friends|join(', ') }}</p>
{% else %}
    <p>My friend is {{ friends }}</p>
{% endif %}

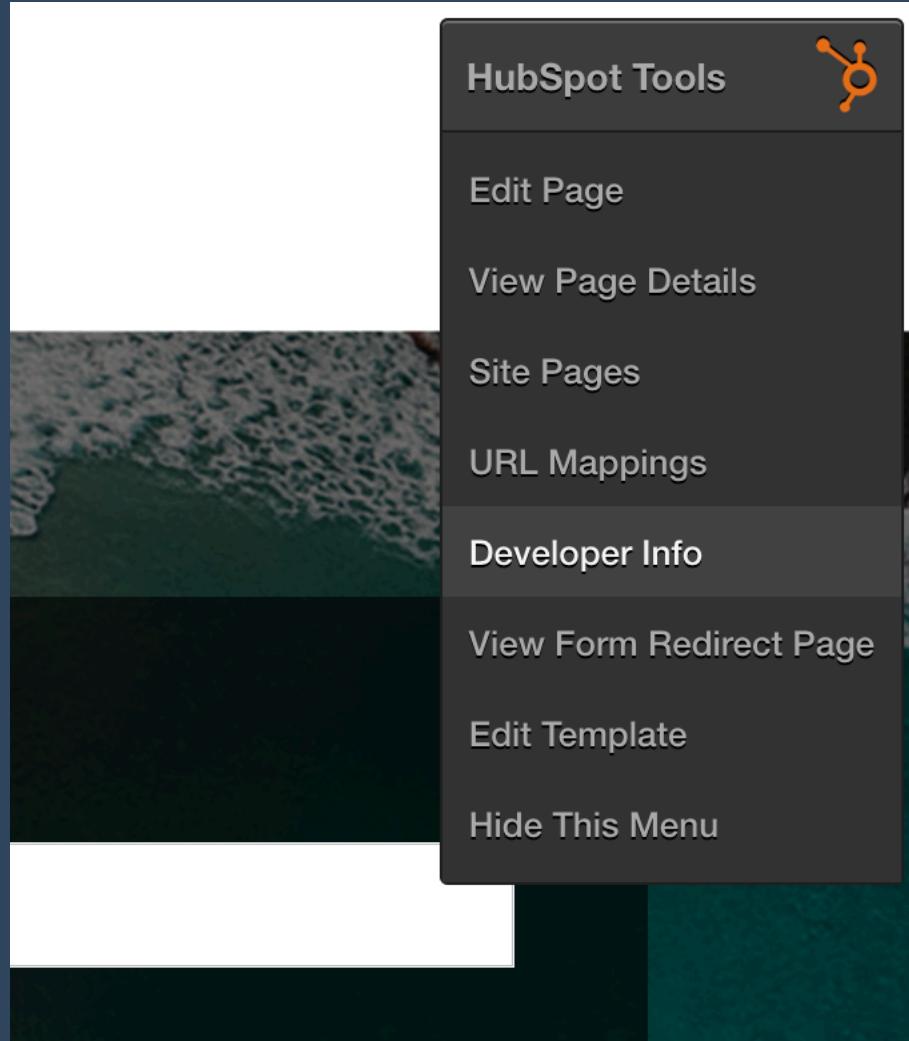
{# OUTPUT: My friends are Joe, Rachel, Doug, Winston #}
```

Conditional statements and expression tests

```
{% set numbers = ['one', 'two', 'three'] %}
{% for number in numbers|reverse %}
    <p>Loop index:{{ loop.index }} Number: {{ number }}</p>
{% endfor %}

{# OUTPUT:
   Loop index:1 Number: three
   Loop index:2 Number: two
   Loop index:3 Number: one
#}
```

Loops with properties and filters



```
{  
  "logo": "<img style=\"border:0;margin:0;padding:0;display:block;text-decoration:none;\" src=\"  
  class=\"hs-site-logo\" >",  
  "render_pass_id": "c61d1983-f16d-451c-ac41-ecbcf0201504",  
  "local_time_zone": "America/New_York",  
  "true": true,  
  "content": {  
    "portal_id": 2574624,  
    "site_id": null,  
    "id": 5843918100,  
    "slug": "landform-landing-test",  
    "content_type_id": null,  
    "content_group_id": null,  
    "campaign": null,  
    "category_id": 1,  
    "state": "DRAFT",  
    "publish_date": 0,  
    "created": 1528230566276,  
    "updated": 1536231292927,  
    "template_path": "generated_layouts/5910081253.html",  
    "name": "<span id=\"hs_cos_wrapper_name\" class=\"hs_cos_wrapper hs_cos_wrapper_meta_field type=\"meta_field\" data-hs-cos-type=\"text\">test landform landing</span>",  
    "archived": false,  
    "author_name": "Jeff Ausura",  
    "deleted_at": 0,  
    "freeze_date": 1531509020000,  
    "processing_status": "",  
    "preview_key": "pWGYuVXA",  
    "ab_test_id": null,  
    "created_by_id": null,  
    "updated_by_id": null,  
    "domain": "",  
    "resolved_domain": "jausura-2574624.hs-sites.com",  
    "legacy_id": null,  
    "performable_guid": null,  
    "is_draft": true,  
    "subcategory": "site_page",  
    "ab_status": null.  
  }  
}
```

Developer Info outputs page data in JSON

```
  "content": {  
    "portal_id": 2574624,  
    "site_id": null,  
    "id": 5843918100,  
    "slug": "landform-landing-test",  
    "content_type_id": null,  
    "content_group_id":  
    "campaign": null,
```

`{{ content.slug }}`

{# OUTPUT: landform-landing-test #}

Data stored as HubL dictionaries

MODULE MARKUP IN CODED TEMPLATES

Two flavors of syntax

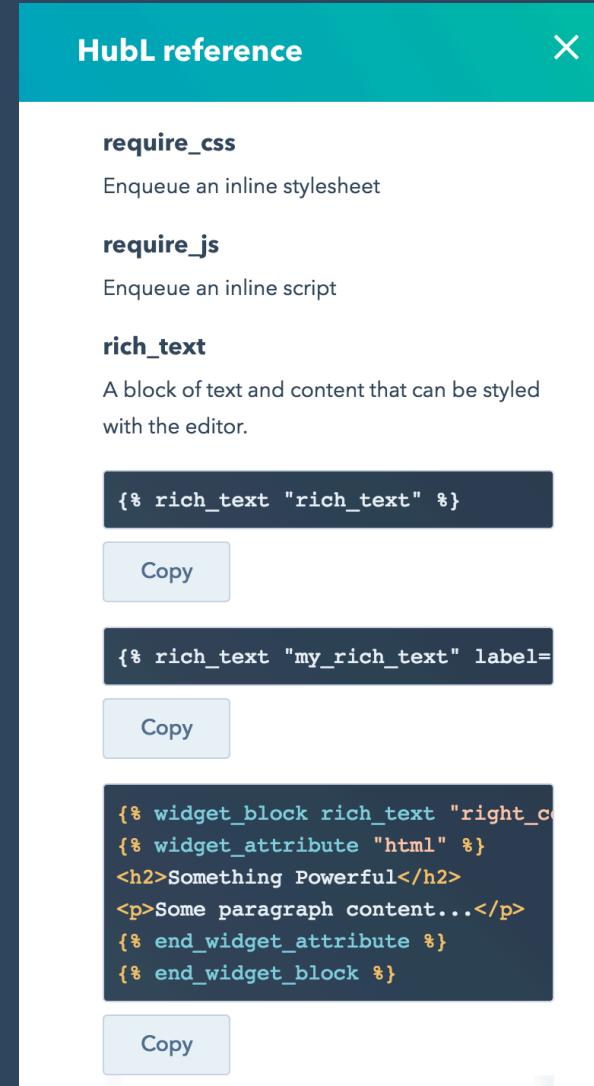
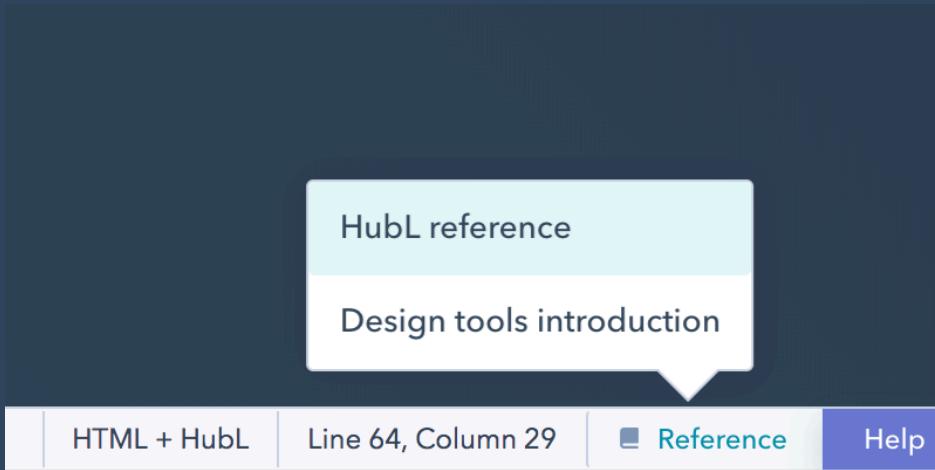
- Basic module syntax
- Block syntax

```
{% text "movie_name" label="Movie Name", value="Citizen Kane" %}
```

Basic module syntax

```
{% widget_block rich_text "movie_description" overrideable=True, label="Movie Description" %}
{% widget_attribute "html" %}
<h2>Citizen Kane</h2>
<p>Citizen Kane is a 1941 American mystery drama film by Orson Welles,
its producer, co-screenwriter, director and star.
The picture was Welles's first feature film. Nominated for Academy Awards
in nine categories, it won an Academy Award for Best Writing (Original Screenplay)
by Herman J. Mankiewicz and Welles. Considered by many critics, filmmakers,
and fans to be the greatest film of all time,
Citizen Kane was voted as such in five consecutive British Film Institute
Sight and Sound polls of critics, and it topped the
American Film Institute's 100 Years.</p>
{% end_widget_attribute %}
{% end_widget_block %}
```

Block module syntax



Reference inside design manager

Parameters available for all modules

Certain modules have certain [special parameters](#), but there are also parameters supported by all modules. Below is a list of parameters supported by all modules.

Parameters name	Type	Description	Default value
label	string	Defines the internal title of the module in the content editor. This parameter can also be used give users additional instructions.	None
overrideable	boolean	Controls whether or not the module can be edited in the content editor. This parameter is the equivalent of using the lock module feature in the	True

no_wrapper	boolean	Setting no_wrapper=True removes the wrapping markup from around the content of a module. Module output to the page is always wrapped in a with special classes. This wrapping markup makes it so when you click the module in the preview pane, the editor scrolls to that module. There may be instances where you want to remove the wrapper, such as if you want to use a text module to populate the destination of an anchor tag href attribute.	False
------------	---------	---	-------

Parameters supported by all modules

WHAT ARE HUBL
FILTERS, AND HOW ARE
THEY USED?

HubL filters handle a wide variety of tasks.

Filters are used for everything from formatting a date object, to casting data types, to filtering sequences.

```
{% set mySeq = ['thing1', 'thing2', 'thing3'] %}  
  
{{ mySeq|length }}  
  
{# outputs 3 #}
```

The length filter

```
{% set brand_color = '#ff9900' %}

a.brand:hover{
    color: rgba('{{ brand_color|convert_rgb }}', .5);
}

{# OUTPUT:
a.brand:hover{
    color: rgba(255, 153, 0, .5);
}
#}
```

The convert rgb filter

```
{% set myBooks = [{'title':'Gravitys Rainbow', 'read':true},  
{'title':'Bleeding Edge', 'read':false}, {'title':'Vineland', 'read':true},  
{'title':'Mason and Dixon', 'read':false}] %}  
  
{% set unreadBooks = myBooks|rejectattr('read') %}  
  
<h1>Books I haven't read yet</h1>  
{% for book in unreadBooks %}  
    <h2>{{ book.title }}</h2>  
{% endfor %}  
  
#{ OUTPUT:  
    <h2>Bleeding Edge</h2>  
    <h2>Mason and Dixon</h2>  
#}
```

The reject attribute filter

```
{% set myBooks = [ {'title':'Gravity's Rainbow', 'read':true},  
{'title':'Bleeding Edge', 'read':false}, {'title':'Vineland', 'read':true},  
{'title':'Mason and Dixon', 'read':false}] %}  
  
<h1>Books I haven't read yet</h1>  
{% for book in unreadBooks|rejectattr('read') %}  
  <h2>{{ book.title }}</h2>  
{% endfor %}  
  
{# OUTPUT:  
  <h2>Bleeding Edge</h2>  
  <h2>Mason and Dixon</h2>  
#}
```

Shorthand application of reject attribute

```
{{ site_settings|pprint }}
```

The pretty print filter