Cheatography

Symfony & Twig Cheat Sheet

by ryguydg via cheatography.com/22137/cs/4445/

Twig Filters

filter	{{ post.published_at|filter() }}

date	{{ post.published_at|date("m/d/Y") }}
	{{ post.published_at|date("m/d/Y", "Europe/Paris") }}

date_modify	{{ post.published_at|date_modify("+1 day")|date("m/d/Y") }}
format	{{ "I like %s and %s."|format(foo, "bar") }}
replace	{{ "I like %this% and %that%."|replace({'%this%': foo, '%that%': "bar"}

number_format	{{ 200.35|number_format }}
	{{ 9800.333|number_format(2, '.', ',') }}

url_encode	{{ data|url_encode() }}
json_encode	{{ data|json_encode() }}
convert_encoding	{{ data|convert_encoding('UTF-8', 'iso-2022-jp') }}

title	{{ 'my first car'|title }}
	{# outputs 'My First Car' #}

capitalize	{{ 'my first car'|capitalize }}

raw	{{ var|raw }}
	{# var won't be escaped #}

trim	{{ ' I like Twig.'|trim('.') }}
upper	{{ 'welcome'|upper }}
lower	{{ 'WELCOME'|lower }}
striptags	{% some_html|striptags %}

join	{{ [1, 2, 3]|join('|') }}
	`{# returns 1|2|3 #}`

split	{{ "one,two,three"|split(',') }}
	`{# returns ['one', 'two', 'three'] #}`

reverse	{% for user in users|reverse %} ... {% endfor %}
absolute	{{ number|abs }}

length	{% if users|length > 10 %}
	...
	{% endif %}

sort	{% for use in users|sort %} ... {% endfor %}
default	{{ var|default('var is not defined') }}

Twig Filters (cont)

keys	{% for key in array|keys %} ... {% endfor %}

escape	{{ user.username|e }}
	`{# is equivalent to #}`
	{{ user.username|e('html') }}
	{{ user.username|e('css') }}
	{{ user.username|e('js') }}

Twig

dump	{{ dump(myBag.myVar) }}
relative url	{{ path('blog_show', {'slug': 'my-blog-post'}) }}
absolute url	{{ url('blog_show', {'slug': 'my-blog-post'}) }}
default	{{ myVar|default() }} default value if myVar is undefined or empty
forloop	{% for item in navigation %} <li><a href="{{ item.href }}">{{ item.caption }}</a></li> {% endfor %}

Services

Call a Service	$this->getContainer()->get('myCompany.my_service')

Repositories

Call a Repository	$this->manager->getRepository('\MyCompany\MyBundle\ORM\Model\MyEntity')

Service persisting

```
$validatedOrganization = new ValidatedOrganization();
$validatedOrganization->setOrganizationIdentifier($number);
$validatedOrganization->setOrganization($organizationType);
$validatedOrganization->setRegisteredName($description);
$this->entityManager->persist($validatedOrganization);
$this->entityManager->flush();
```