BIX5
헬퍼 함수 가이드

Guide for BIX5Helper.js

정품을 구입하신 고객에게는 기술상담 및 지원을 제공합니다.

(주)빅스소프트
서울시 영등포구 영신로 220, 610 호 (영등포동 8 가, KnK 디지털타워)
Tel. 02-2655-9784, bix5cs@bixsoft.net

BIXSOFT 빅스소프트

목차

1. 템플릿 엔진 소개

소개

Handlebars란 무엇인가요?

Handlebars는 간편한 **템플릿 언어**입니다.

템플릿과 입력 객체를 사용하여 HTML 또는 다른 텍스트 형식을 생성합니다. Handlebars 템플릿은 일반 텍스트와 유사하며, Handlebars 표현식이 포함되어 있습니다.

template

```
<p>{{firstname}} {{lastname}}</p>
```

Handlebars 표현식은 `{{` 와 일부 내용, 그리고 `}}` 로 구성됩니다. 템플릿이 실행되면 이러한 표현식은 입력 객체의 값으로 대 체됩니다.

언어 기능

간단한 표현식

앞서 설명한 것처럼, 다음 템플릿은 두 개의 Handlebars 표현식을 정의합니다.

template

```
<p>{{firstname}} {{lastname}}</p>
```

입력 객체에 적용하면:

input

```
{
  firstname: "Yehuda",
  lastname: "Katz",
}
```

표현식은 해당 속성으로 대체됩니다. 결과는 다음과 같습니다:

output

```
<p>Yehuda Katz</p>
```

내장 헬퍼(Built-in Helpers)

#if

`if` 헬퍼를 사용하여 조건부로 블록을 렌더링할 수 있습니다. 인수가 `false`, `undefined`, `null`, `""`, `0`, 또는 `[]` 를반 환하면 Handlebars는 블록을 렌더링하지 않습니다.

template

```
<div class="entry">
{{#if author}}
<h1>{{firstName}} {{lastName}}</h1>
{{/if}}
</div>
```

다음 입력을 위 템플릿에 전달하면

input

```
{
  author: true,
  firstName: "Yehuda",
  lastName: "Katz",
}
```

다음과 같은 결과를 생성합니다:

output

```
<div class="entry">
<h1>Yehuda Katz</h1>
</div>
```

입력이 빈 JSON 객체 `{}` 이면, `author` 는 `undefined` 가 되어 `if` 조건이 실패하고 다음과 같은 출력이 생성됩니다:

html

```
<div class="entry"></div>
```

블록 표현식을 사용할 때, 표현식이 거짓 값을 반환하면 실행할 템플릿 섹션을 지정할 수 있습니다. `else` 로 표시된 이 섹션은 "else 섹션"이라고 합니다.

template

```
<div class="entry">
{{#if author}}
<h1>{{firstName}} {{lastName}}</h1>
{{else}}
<h1>Unknown Author</h1>
{{/if}}
</div>
```

#each

내장 `each` 헬퍼를 사용하여 리스트를 반복할 수 있습니다. 블록 내부에서는 `this` 를 사용하여 반복 중인 요소를 참조할 수 있 습니다.

template
```
<ul class="people_list">
  {{#each people}}
    <li>{{this}}</li>
  {{/each}}
</ul>
```

이 컨텍스트가 사용될 때:

input
```
{
  people: [
    "Yehuda Katz",
    "Alan Johnson",
    "Charles Jolley",
  ],
}
```

다음과 같은 결과를 생성합니다:

output
```
<ul class="people_list">
    <li>Yehuda Katz</li>
    <li>Alan Johnson</li>
    <li>Charles Jolley</li>
</ul>
```

`this` 표현식을 사용하여 현재 컨텍스트를 참조할 수 있습니다.

리스트가 비어 있을 때만 표시될 `else` 섹션을 선택적으로 제공할 수 있습니다.

template
```
{{#each paragraphs}}
<p>{{this}}</p>
{{else}}
<p class="empty">No content</p>
{{/each}}
```

`each` 에서 항목을 반복할 때, {{@index}}를 통해 현재 반복 인덱스를 참조할 수 있습니다.

handlebars
```
{{#each array}} {{@index}}: {{this}} {{/each}}
```

또한 객체를 반복할 때 `{{@key}}` 는 현재 키 이름을 참조합니다:

handlebars
```
{{#each object}} {{@key}}: {{this}} {{/each}}
```

> 배열을 반복할 때 첫 번째와 마지막 단계는 `@first` 와 `@last` 변수를 통해 표시됩니다.
>
> 중첩된 `each` 블록은 깊이 기반 경로를 통해 반복 변수를 액세스할 수 있습니다. 예를 들어, 부모 인덱스에 접근하려면 `{{@../index}}` 를 사용할 수 있습니다.

출처 : https://handlebarsjs.com/ko/guide

2. 예약어

Handlebarsjs 문법으로 작성시 다음 예약어를 사용하실 수 있습니다.

예약어	설명
{{id}}	위젯 ID
{{meta}}	레이아웃 메타 오브젝트
{{data}}	바인딩 된 데이터 오브젝트

3. BIX5Helper 함수

BIX5Helper 는 뷰 템플릿 엔진인 Handlebarsjs 의 registerHelper 라이브러리입니다. BIX5 는 위젯 레이아웃을 데이터에 따라 동적으로 작성하기 위하여 뷰 템플릿 엔진인 Handlebarsjs 를 내장하고 있습니다. 레이아웃 작성 시 자주 사용되는 helper 함수들을 제품에서 제공하고 있습니다. 대시보드 미리보기, 편집화면에서는 기본적으로 해당 라이브러리를 불러오고 있습니다.

기본적으로 RegisterHelper 는 다음과 사용법을 따릅니다.
{{헬퍼명 속성 1="값 1" 속성 2="값 2" 속성 3="값 3" …}}

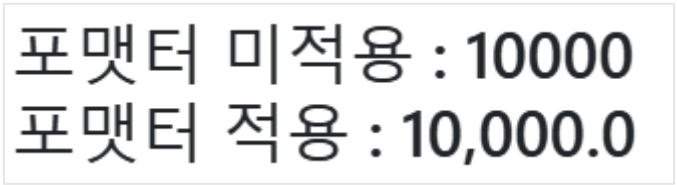
3.1. NumberFormatter

숫자를 표시하는 포맷터 헬퍼입니다.

사용법

```
<h1>
        포맷터 미적용 : {{data.0.value}}
</h1>
<h1>
        포맷터 적용 : {{NumberFormatter data.0.value precision="1"}}
</h1>
```

결과



속성

속성명	타입	설명	기본값
precision	number	출력할 소수점 자리수	0
thousandsSeparatorTo	string	천단위 구분 기호 문자	,

decimalSeparatorTo	string	소수점 구분 기호 문자	.
applyPrecisionToInteger	boolean	정밀도 적용 여부	“true”
useNegativeSign	boolean	마이너스 기호 사용여부	“true”
showDecimalZero	boolean	소수점 이하 자리수가 0 일 경우 보여주는 여부	“true”
nanSign	string	NaN 값인 경우 출력할 문자	“”

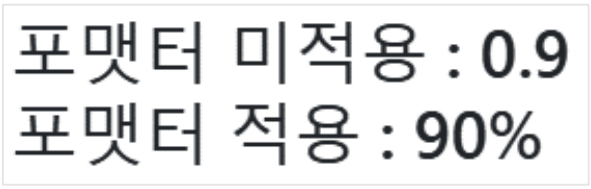
3.2. PercentFormatter

퍼센트를 표시하는 포맷터 헬퍼입니다.

사용법

```
<h1>
        포맷터 미적용 : {{data.0.value}}
</h1>
<h1>
        포맷터 적용 : {{PercentFormatter data.0.value multiplyHundred="true"}}
</h1>
```

결과



속성

2.1 NumberFormatter 를 상속받습니다.

속성명	타입	설명	기본값
divideHundred	boolean	변환할 값을 100 으로 나누어 계산할지 여부	“false”
multiplyHundred	boolean	변환할 값을 100 으로 곱하여 계산할지 여부	“false”
percentSymbol	string	백분율표시 심볼	“%”

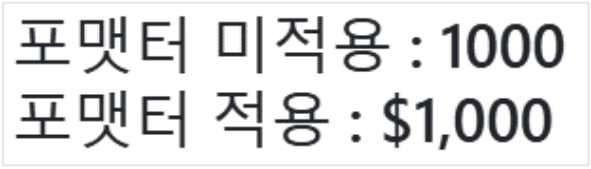
3.3. CurrencyFormatter

화폐를 표시하는 포맷터 헬퍼입니다.

사용법

```
<h1>
        포맷터 미적용 : {{data.0.value}}
</h1>
<h1>
        포맷터 적용 : {{CurrencyFormatter data.0.value}}
</h1>
```

결과



속성

2.1 NumberFormatter 를 상속받습니다.

속성명	타입	설명	기본값
currencySymbol	string	화폐표시 심볼	“$”
alignSymbol	string	화폐표시 심볼의 위치 유효값:left, right	“left”

3.4. ifCond

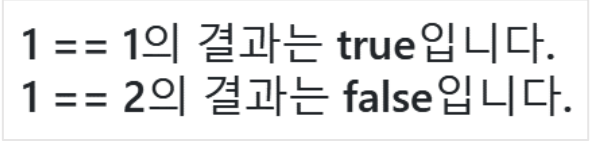
비교 연산을 위한 헬퍼입니다.

사용법

```
{{#ifCond 1 "==" 1}}
<h1>
        1 == 1 의 결과는 true 입니다.
</h1>
{{else}}
<h1>
        1 == 1 의 결과는 false 입니다.
</h1>
{{/ifCond}}

{{#ifCond 1 "==" 2}}
<h1>
        1 == 2 의 결과는 true 입니다.
</h1>
{{else}}
<h1>
        1 == 2 의 결과는 false 입니다.
</h1>
{{/ifCond}}
```

결과



속성

속성명	타입	설명	기본값
값 1		(필수) 비교할 값	
operator	string	(필수) 비교 연산자. 유효값 ==, ===, !=, !==, <, <=, >, >=, &&, ||	-
값 2		(필수) 비교할 값	

3.5. DataFieldFormatter

컬럼 데이터에서 필드를 꺼낼 때 사용하는 포맷터 헬퍼입니다.

사용법

레이아웃 예시

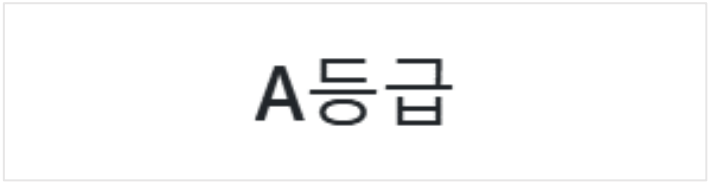
```
{{#if data}}
    <h1>{{DataFieldFormatter data.[0] meta.columnVal}}</h1>
{{else}}
-
{{/if}}
```

컴포넌트 옵션 예시(프로퍼티옵션에 의해 사용자 메뉴가 만들어지고 사용자가 특정 필드를 지정한 상황)

```
{
  "propertyOptions": [
    {
      "label": "컬럼",
      "control": "ColumnComboBox",
      "attribute": "columnVal"
    }
  ],
  "layoutMetaData": {
      "columns": [
        {
          "id": 201,
          "name": "등급",
          "label": "등급",
          "dataType": "string",
          "memo": "등급",
          "aggregate": "",
          "alias": "등급",
          "category": "dimension"
        }
```

```
        ],
        "columnVal": {
          "id": 201,
          "name": "등급",
          "label": "등급",
          "dataType": "string",
          "memo": "등급",
          "aggregate": "",
          "alias": "등급",
          "category": "dimension"
        }
    },}
```

결과



속성

속성명	타입	설명	기본값
data	object	(필수)	-
field	object	(필수) 연산 대상 값	-

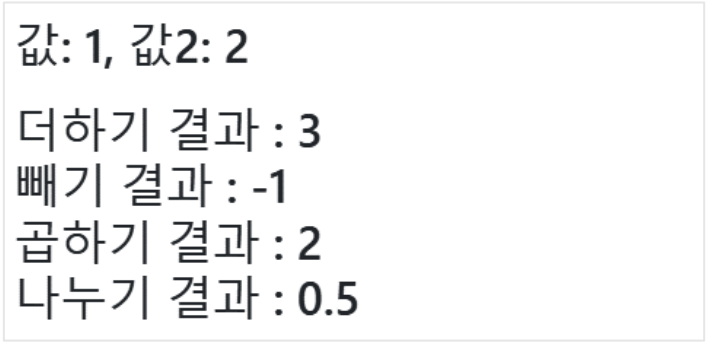
3.6. 사칙연산

간단한 사칙연산 헬퍼입니다.

사용법

```
<h1>
        값: {{meta.value1}}, 값 2: {{meta.value2}}
</h1>
<br>
<h1>
        더하기 결과 : {{plus meta.value1 meta.value2}}
</h1>
<h1>
        빼기 결과 : {{minus meta.value1 meta.value2}}
</h1>
<h1>
        곱하기 결과 : {{multiply meta.value1 meta.value2}}
</h1>
<h1>
        나누기 결과 : {{divide meta.value1 meta.value2}}
</h1>
```

결과



속성

속성명	타입	설명	기본값
값 1	number	(필수) 연산 대상 값	-
값 2	number	(필수) 연산 대상 값	-

감사합니다