

---

# Breathe Sample Documentation

*Release 1.0.0*

**Eevee**

**Mar 10, 2019**



<b>1</b>	<b>Getting Started</b>	<b>1</b>
<b>2</b>	<b>Classes</b>	<b>3</b>
2.1	Sum . . . . .	3
<b>3</b>	<b>Functions</b>	<b>5</b>
3.1	accumulate . . . . .	5
3.2	average . . . . .	5
3.3	average_float . . . . .	5



# CHAPTER 1

---

## Getting Started

---

Defined in `stream/stream.hpp`



### 2.1 Sum

```
template <typename T>
struct Sum
    Sum struct to accumulate the values.
```

#### Public Functions

```
Sum ()
    Constructor.
```

```
T Zero ()
    Get the initial value.
```

**Return** The initial value.

```
void operator() (T n)
    Call operator to accumulate a value.
```

#### Parameters

- *n*: a value to be accumulated.

#### Public Members

```
T sum
    The sum value.
```





### 3.1 accumulate

**template** <typename It, typename Result = typename std::iterator\_traits<It>::value\_type>

Result stream::: **accumulate** (It *first*, It *last*)

Computes the sum of the given values.

**Return** The result.

**Parameters**

- *first*: the range first.
- *last*: the range last.

### 3.2 average

**template** <typename It, typename Result = typename std::iterator\_traits<It>::value\_type>

Result stream::: **average** (It *first*, It *last*)

Computes the average of the given values.

**Return** The result.

**Parameters**

- *first*: the range first.
- *last*: the range last.

### 3.3 average\_float

**template** <typename It>

`float stream: :average_float` (It *first*, It *last*)  
Computes the average(float) of the given values.

**Return** The float result.

**Parameters**

- `first`: the range first.
- `last`: the range last.

## S

`stream::accumulate (C++ function), 5`  
`stream::average (C++ function), 5`  
`stream::average_float (C++ function), 5`  
`stream::Sum (C++ class), 3`  
`stream::Sum::operator () (C++ function), 3`  
`stream::Sum::Sum (C++ function), 3`  
`stream::Sum::sum (C++ member), 3`  
`stream::Sum::Zero (C++ function), 3`