



# C++リファレンスサイトcpprefjp でのドキュメンテーション

高橋 晶 (Akira Takahashi)

faithandbrave@GitHub

Preferred Networks, Inc.

2017/11/18 (土) OSSドキュメンテーション勉強会

# cpprefjpとは

- ・ プログラミング言語C++の日本語リファレンスサイト
- ・ 特徴は、標準ライブラリの全てのクラス、全ての関数にサンプルコードを提供するのを目標にしていること
- ・ 多くのユーザーにとって重要なのは概要文章とサンプルコードである、という考えが基本としてある

# サイト規模

- ・ 現在、3,000ページ超の情報を提供中
- ・ 標準ライブラリのリファレンスがメインだが、各所からの強い要望にともなって言語機能の解説もはじめている
- ・ C++14までは、ほぼほぼ全体を網羅している
  - ・ マイナーなライブラリ機能は保留中
  - ・ 言語機能はC++11以降の新機能のみ
- ・ 現在、C++17とC++20に対応中（次期バージョン）

# 執筆環境

- ・ GitHubリポジトリでMarkdownでドキュメントを書いて、
- ・ それをHTMLに自動変換して  
GitHub Pagesのリポジトリにデプロイしている
- ・ Markdown -> HTMLの変換サーバーは、サイト管理者の一人であるめるぽんの個人サーバーを使用

# マネタイズ

- ・ なし
- ・ 管理者、執筆者含めて全てボランティア
- ・ 法人化して企業スポンサーを募ることも考えられたが、手が回らず停滞中
- ・ サーバー代はめるぽん個人が負担している。  
彼が運営しているWandboxサービスが収入源

# 議論場所

- ・ GitHub Issueのみ
- ・ GitterやSlackのようなチャットを用意する予定は今のところない
- ・ 議論の履歴が追いにくくなると困るから
- ・ 気軽に要望を言えすぎても対応しきれないから、  
ある程度のハードルとしてIssueを議論場所になっている
- ・ 正式な議論の場以外 (たとえばTwitter) は、コントリビュータの合意を得られる場ではないので議論場所とは見なしていない

# GitHub + Markdownの何が いい か

- ・ ブラウザで編集するWikiと違って、ローカル環境に全てのドキュメントデータを持てる
- ・ 全てプレーンテキストだからgrep、一括置換、解析などが楽
- ・ 文書構造を統一しやすい
- ・ ページ単位の編集ではなくコミット単位の編集なので、関連した修正を追跡しやすい

# Markdown拡張

- ・ Markdownは表現力がそれほど高くない
- ・ しかし、プログラミング関係のドキュメントには、いろいろな表現が必要になる
- ・ cpprefjpでは、以下のような拡張記法を定義している
  - ・ コードブロック中識別子の修飾(リンク、文字色など)
  - ・ (主に) SEO用のページメタ情報
  - ・ 数式 (Mathjax)



# コードブロック中識別子の修飾

- コードブロックのすぐ下に、箇条書きで修飾リストを並べる

```
```cpp
#include <vector>

int main()
{
    std::vector<int> v = {1, 2, 3};
    v.emplace_back(4);
}
```

* emplace_back[color ff0000]
* std::vector[link /reference/vector.md]
```

# ページメタ情報 1/3

- ・ ページ名 (見出し1) のすぐ下に、箇条書きでメタ情報を並べる

```
# emplace_back
* vector[meta header] 所属するヘッダファイル
* std[meta namespace] 所属する名前空間
* vector[meta class] 所属するクラス
* function[meta id-type] 機能の分類 (クラス、関数、定数など)
* cpp11[meta cpp] どのバージョンから入った機能か

```cpp
template <class... Args>
void emplace_back(Args&&... args);
```
```

# ページメタ情報 2/3

🏠 / リファレンス / vector / `emplace_back`

"std::...", "<header>", etc.



最終更新日時: 2017年05月11日 05時18分47秒

Akira Takahashi が更新

🕒 履歴

✎ 編集

*function*

<vector>

std::vector::**emplace\_back** ⚡ C++11

```
template <class... Args>
void emplace_back(Args&&... args);

template <class... Args>
void vector<bool>::emplace_back(Args&&... args); // C++14
```

## 概要

直接構築で新たな要素を末尾に追加する。

この関数の引数 `args...` は、要素型 `T` のコンストラクタ引数である。当関数の内部で要素型 `T` のコンストラクタを呼び出し、追加する要素を構築する。

# ページメタ情報 3/3

```
<meta charset="UTF-8">
<title>vector::emplace_back - cpprefjp C++日本語リファレンス</title>
<meta name="viewport" content="width=device-width,initial-scale=1">
<meta name="keywords" content="
  C++,標準ライブラリ,リファレンス,ドキュメント,STL,std,cpp11,vector,std,vector,function
">
<meta name="title" content="vector::emplace_back - cpprefjp C++日本語リファレンス">
<meta itemprop="name" content="vector::emplace_back - cpprefjp C++日本語リファレンス">
<meta property="og:title" content="vector::emplace_back - cpprefjp C++日本語リファレンス">
<meta property="og:url" content="https://cpprefjp.github.io/reference/vector/emplace_back.html">
<meta property="og:site_name" content="cpprefjp - C++日本語リファレンス">
<meta property="og:type" content="article">
<meta property="og:description" content="直接構築で新たな要素を末尾に追加する。">
<meta name="twitter:card" content="summary">
<meta name="twitter:title" content="vector::emplace_back - cpprefjp C++日本語リファレンス">
<meta name="twitter:url" content="https://cpprefjp.github.io/reference/vector/emplace_back.html">
<meta name="twitter:description" content="直接構築で新たな要素を末尾に追加する。">
```

# 数式 1/2

- ・ ページ単位でMathjaxを有効にし、TeX記法で数式を記述する

```
# hypot
* cmath[meta header]
* std[meta namespace]
* function[meta id-type]
* [mathjax enable] このページでMathjaxを有効にする
* cpp11[meta cpp]
```

...

この関数の効果である以下の式は、三平方の定理の式変形である：

$$f(x, y) = \sqrt{x^2 + y^2}$$

## 数式 2/2

この関数は、「三平方の定理」によって、直角三角形の斜辺の長さを求める  
よび斜辺cがあったとき、辺の長さは、三平方の定理によって以下の関係が成

$$a^2 + b^2 = c^2$$

この関数の効果である以下の式は、三平方の定理の式変形である：

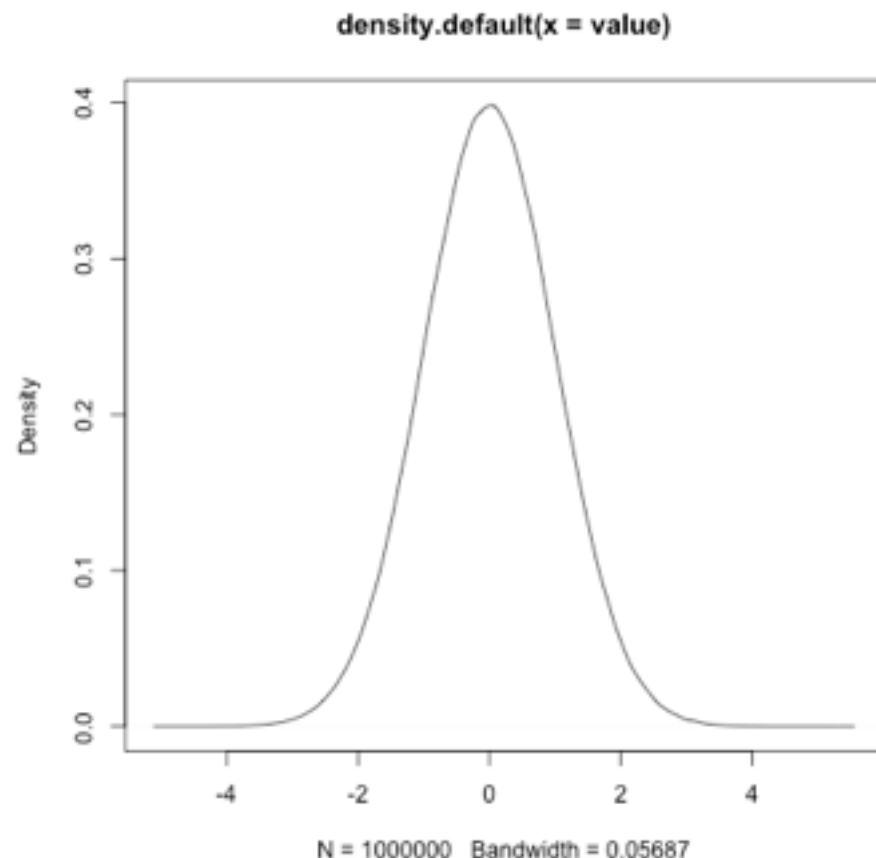
$$f(x, y) = \sqrt{x^2 + y^2}$$

$a^2 + b^2$  は  $c^2$  と等しくなるため、2乗の和に対する平方根を求めることで、  
する辺aとbの長さを与えることで、戻り値として斜辺cの長さが返される。

# チャート管理

- ・ 値の遷移や統計などを可視化したチャートは、画像管理用のリポジトリに入れている
- ・ チャートはR言語で書いて、そのソースコードもリポジトリに入れている
- ・ これによって、サイト内画像の書き方も、ある程度統一できている

このプログラムによってある時に得られた結果 ([normal\\_distribution.tsv](#)) を図示する。



```
# normal_distribution.R
png("normal_distribution.png")

x <- read.table("normal_distribution.tsv")
value = x$V1

plot(density(value))
```

# 権限付与 1/3

- ・ サイトの編集を手伝っていただける方は、常に募集している
- ・ Pull Requestを送ってくれてマージまでしたら権限付与



# 権限付与 2/3

- Webの編集画面からPull Requestを送れる

[リファレンス](#) / [optional](#) / optional

"std::...", "<header>", etc.

最終更新日時: 2017年11月02日 03時15分12秒  
Akira Takahashi が更新

[履歴](#) [編集](#)

*class template*

std::optional ⚡ C++17

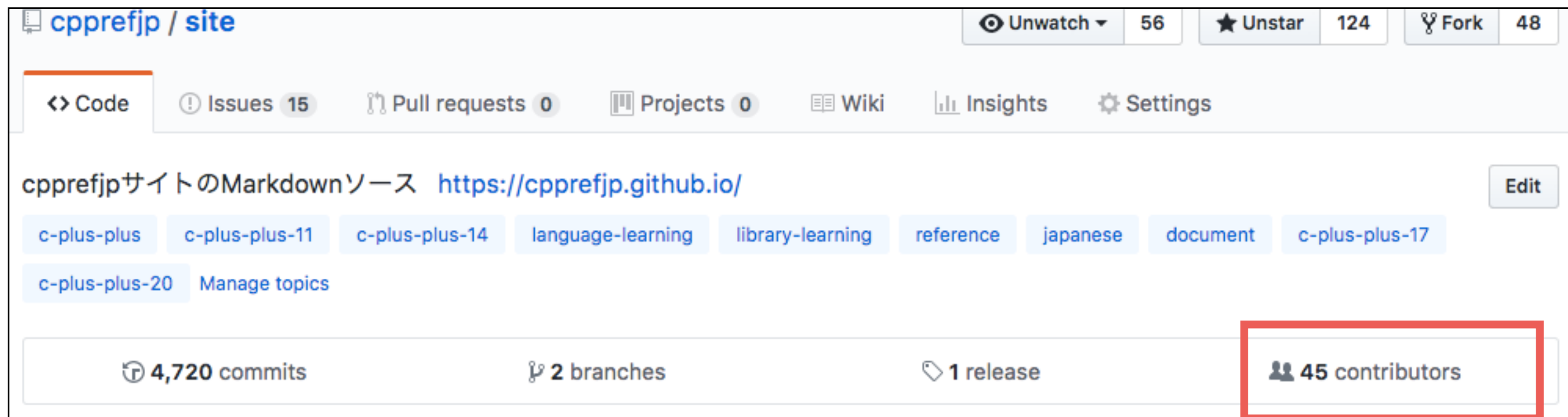
```
namespace std {  
    template <class T>  
    class optional;  
}
```

概要

optional クラスは、任意の型 `T` の値を有効値として、あらゆる型に共通の無効値状態を表現できる

# 権限付与 3/3

- 現在のコントリビューター状況 (GitHubに移行してから)

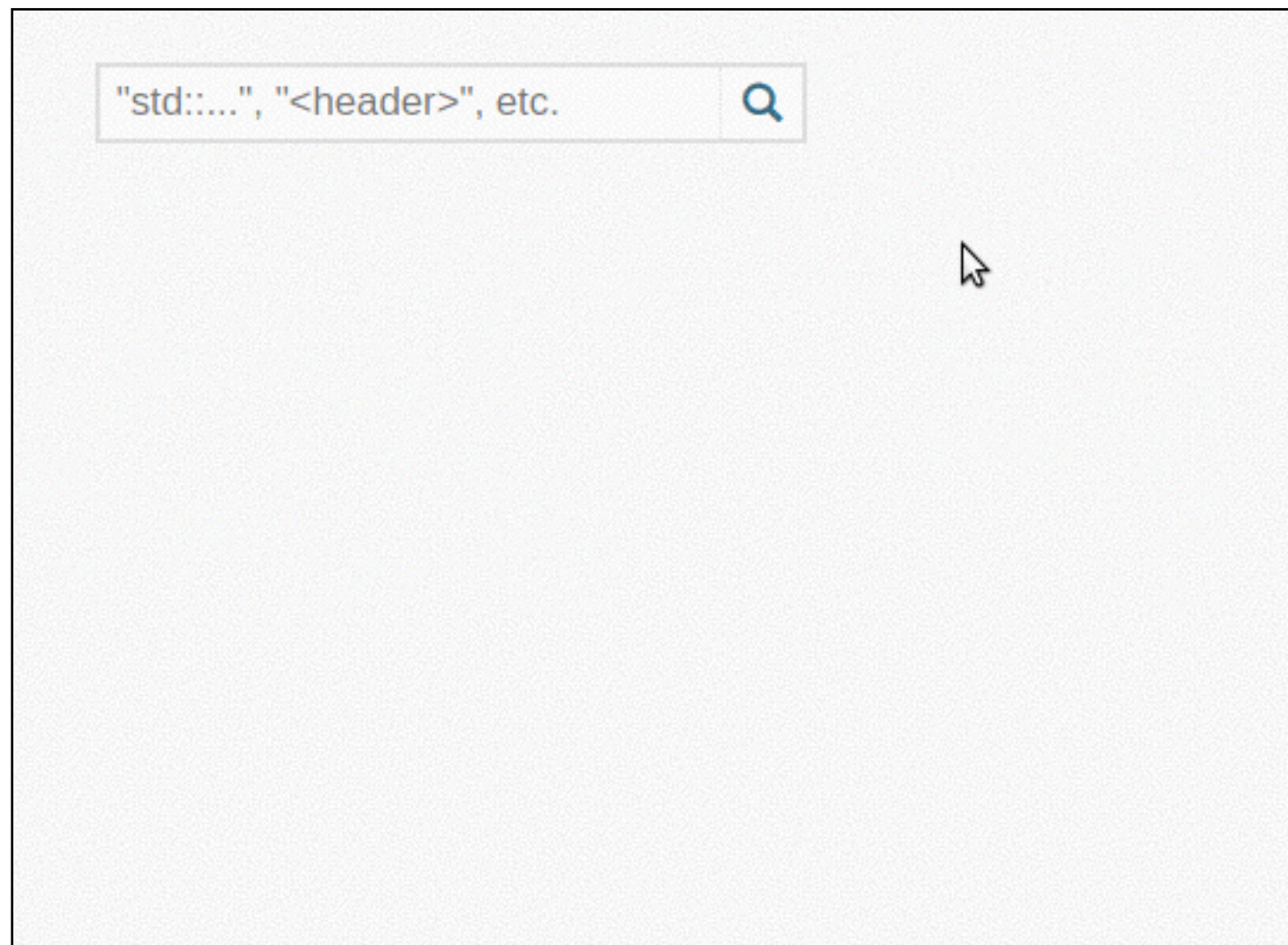


The screenshot displays the GitHub repository page for `cpprefjp / site`. At the top, there are buttons for `Unwatch` (56), `Unstar` (124), and `Fork` (48). Below these are tabs for `Code`, `Issues` (15), `Pull requests` (0), `Projects` (0), `Wiki`, `Insights`, and `Settings`. The main content area shows the repository name `cpprefjp` and the file `site`, with a link to the Markdown source: <https://cpprefjp.github.io/>. Below this is a list of topics: `c-plus-plus`, `c-plus-plus-11`, `c-plus-plus-14`, `language-learning`, `library-learning`, `reference`, `japanese`, `document`, `c-plus-plus-17`, and `c-plus-plus-20`, along with a `Manage topics` link. At the bottom, there are statistics for `4,720 commits`, `2 branches`, `1 release`, and `45 contributors`. The `45 contributors` link is highlighted with a red box.

<https://github.com/cpprefjp/site/graphs/contributors>

# 検索エンジン

- ・ 最近 saki7@GitHub さんが作ってくれた



# FAQ 1 / 3

- ・ Q. ボランティアベースでいつまで続けられそうか？
- ・ 私が生きてる限りは、一人でも続けるつもり。  
心が折れたらわからない (何度も折れてる)

# FAQ 2/3

- ・ Q. 最新・次期規格への対応は、どれくらいのスピードでやるつもり？
- ・ のんびり。そのさらに次の規格がでるまでには終わらせたい気持ち

# FAQ 3/3

- ・ Q. 言語入門やチュートリアル、言語機能の網羅的な解説を書く予定は？
- ・ ない。継続的なコントリビュータがやってくれるのであれば検討するが、言語バージョンごとに用意するのもたいへん
- ・ 需要があるからできるわけではないのと、作っておわりではないのが難しい

# まとめ

- ・ GitHub + Markdownで管理すると、いろいろラク
- ・ Markdownの独自拡張はどうしても必要になる
- ・ 画像もプレーンテキストで書けば管理できる
- ・ コントリビュータのみなさん、ありがとう