



ブラウザの枠を飛び越えろ！**Node.js**セミナー

## Chapter01

# Node.jsの概要と バージョン

# Node.jsとは

Node.jsはGoogle Chromeと同じ「V8」という実行エンジンを使って作られたJavaScriptの実行環境です。

従来JavaScriptはブラウザでしか動作できませんでしたが、このNode.jsを使うことで、JavaScriptを様々な場所で実行できるようになります。

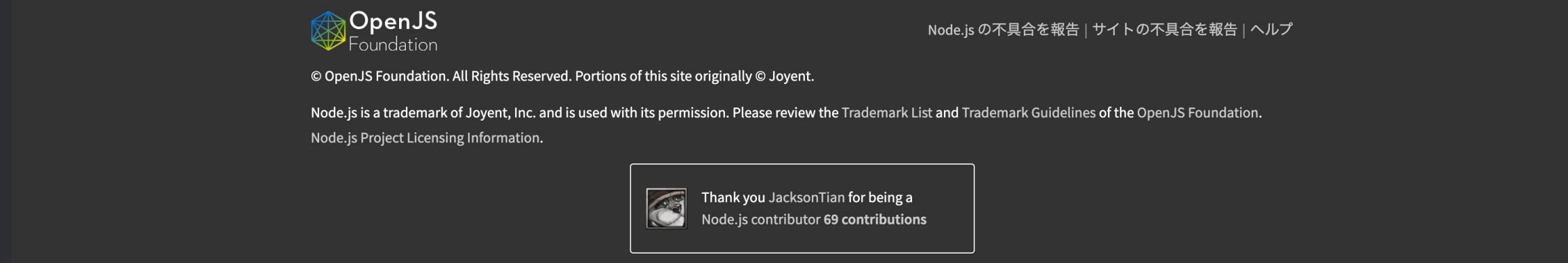
サーバーサイドの開発やフロントエンドの開発環境などに用いられています。



# Node.jsの2つのバージョン

Node.jsはWebサイトから無料で利用することができる。

ただしNode.jsには2種類のバージョンがありそれぞれ性質が違う。



## ○ 偶数バージョン

「LTS (Long Term Supportの略)」と呼ばれるバージョン。  
長期サポートが保証されている。

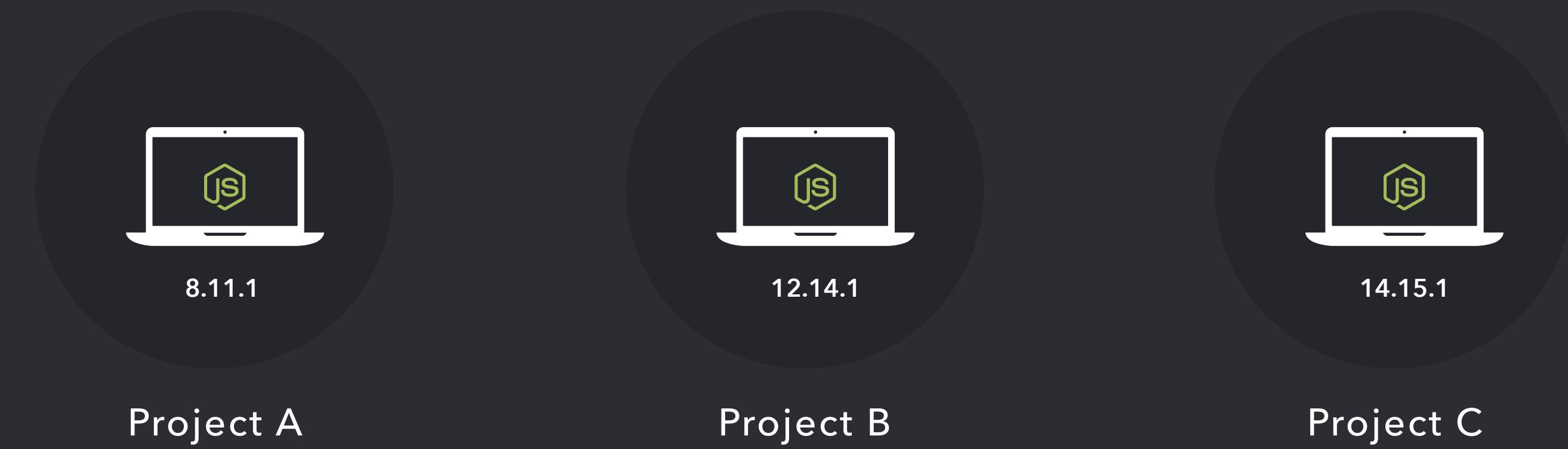
## ○ 奇数バージョン

新しい技術を積極的に取り組んでいくバージョン。  
1年ほどでサポートされなくなったりする。

# Node.jsのインストール

Node.jsのインストールは公式サイトからダウンロードしてインストールをすることができる。

ただしNode.jsはバージョンに依存することが多く様々な案件やチームで開発を行う場合、様々なバージョンのNode.jsが必要になるケースがある。



案件によってNodeのバージョンを変更する必要がある。

# Node.jsのバージョン管理

Node.jsには様々なバージョン管理を行うツールがあるので、自分の環境にあったバージョン管理ツールを使う。

基本的にはどのバージョン管理のツールもコマンドラインを使ってバージョンを変更したりインストールしたりを行う。

バージョン管理ツール	対応OS	概要
nodebrew	Mac	日本での利用者が多く定番の管理ツール
nvm	Mac	人気があるバージョン管理ツール
nvm-windows	Windows	nvm のWindows版のバージョン管理ツール
nodenv	Mac	プロジェクトディレクトリに移動時にバージョンを自動で変える
nodist	windows	nodenvやnに影響を受けているWindows用のバージョン管理ツール
n	Mac	システムのNode.jsのバージョン管理を行う

# nodenvをインストールしてみる (macのみ)

---

nodenvをインストールするにはコマンドラインツールを利用する。

ただし、今後node.js以外の「env」系のバージョン管理ツール（PHPやPythonなど）を使う場合は、

それらを管理する「anyenv」というパッケージを入れると一元管理できる。

また、「anyenv」以外のパッケージ等もまとめて管理するならば「Homebrew」というパッケージ管理ツールもあると便利。

## ○ nodenv

node.jsのバージョンを管理するツール  
(今回のメイン)

## ○ anyenv

様々なenv系ツールをまとめれるパッケージ  
(PHPやPythonなど)

## ○ Homebrew

macの様々なパッケージを管理できるツール  
(gitなど)

# Homebrewをインストールしてみる

Homebrewはmacの様々なパッケージを管理できるツール。

nodeenvをインストールするためだけでは必須ではないが、macにインストールするパッケージを管理できて便利。

インストールはサイトにアクセスし表示されるコマンドをターミナルにコピペすることでインストールできる。



## バージョンの確認

```
brew -v
```

Terminal

# anyenvをインストールしてみる

anyenvは様々なenv系のツールを管理するためのパッケージ。

今後他の言語など（PHPやPythonなど）のバージョン管理などを行う場合は入れておいてもいい。

nodenvのインストールも比較的楽にできる。

Homebrew経由でインストール

```
brew install anyenv
```

Terminal

GitHubからインストール

```
git clone https://github.com/anyenv/anyenv ~/.anyenv
```

Terminal

# pathを通す

anyenvでインストールしたものをどこからでもコマンドで使えるようにするために path を通す。

pathはどのシェルを使ってるかによって設定ファイルが異なる。

macOS Catalina からはデフォルトのシェルが「zsh」となっているがそれより前のmacでは「bash」が採用されていた。

## bashの場合

Terminal

```
echo 'export PATH="$HOME/.anyenv/bin:$PATH"' >> ~/.bash_profile
echo 'eval "$(anyenv init -)"' >> ~/.bash_profile
```

## zshの場合

Terminal

```
echo 'export PATH="$HOME/.anyenv/bin:$PATH"' >> ~/.zshrc
echo 'eval "$(anyenv init -)"' >> ~/.zshrc
```

bashを通したあとはシェルを再起動して設定を有効にする。

Terminal

```
exec $SHELL -l
```

# nodenvのインストール

いよいよnodenvをインストールする。ここではanyenvでインストールする方法を紹介する。

anyenvが使えるかバージョンを確認する

```
anyenv --version
```

Terminal

env系を入れるためのプラグインをインストール

```
anyenv install --init
```

Terminal

マニフェストディレクトリを作るか？聞かれるので「y」（yes）を入力して作成する

```
Manifest directory doesn't exist: /Users/○○/.config/anyenv/anyenv-install
Do you want to checkout ? [y/N]: y
```

Terminal

nodenvをインストール

```
anyenv install nodenv
```

Terminal

シェルを再起動して設定を有効にする。

```
exec $SHELL -l
```

Terminal

# anyenvのアンインストール

anyenvで入れた env 系のアンインストールと、anyenv自体のアンインストールを紹介。

インストールした nodenv をアンインストールする場合

```
anyenv uninstall nodenv
```

Terminal

Homebrewで入れた anyenv をアンインストール

```
brew uninstall anyenv
```

Terminal

GitHubから入れた anyenv をアンインストール

```
rm -rf $(anyenv root)
```

Terminal

# Node.jsのインストール

いよいよNode.jsをインストールする。

インストール可能なバージョンを表示する

```
nodenv install -l
```

Terminal

バージョンを指定して Node.js をインストール

```
nodenv install バージョン番号
```

Terminal

指定したバージョンをグローバルに設定する

```
nodenv global バージョン番号
```

Terminal

指定したバージョンをローカルに設定する（cd でバージョンを指定するディレクトリに移動してから）

```
nodenv local バージョン番号
```

Terminal

インストールされているなnodeのリスト

```
nodenv versions
```

Terminal

# Node.jsのバージョン確認

いよいよNode.jsがインストールできたら、  
Node.jsが利用できるかバージョンを確認する。

## Node.jsのバージョン確認

```
node -v
```

Terminal

## Nodeのパッケージマネージャー (npm) のバージョン確認

```
npm -v
```

Terminal