

はじめよう／

# ジフン 専用 パソコン

## 第42回 ジャバスクリプトでスクラッチを拡張しよう

前回はRubyとスクラッチを使って、ブロック型とテキスト型のプログラミング言語の違いについて考えてみた。でも、どのようなときにテキストでプログラムを書けばよいのか、まだピンときてないんじゃないかな。そこで今回は、ジャバスクリプト(JavaScript)というテキスト型のプログラミング言語を使って、ブロックだけではプログラムを書くのが難しいときに、スクラッチ自体を拡張する例を紹介しよう。

ジブン専用パソコン特設サイト  
[https://prog.kodomonokagaku.com/  
category/raspberrypi](https://prog.kodomonokagaku.com/category/raspberrypi)

### Webのために生まれた 「ジャバスクリプト」

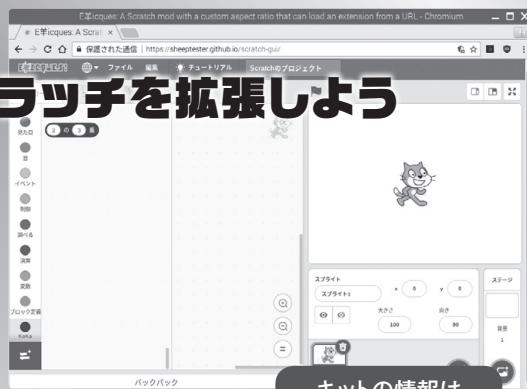
ジャバスクリプトは、アメリカのブレンダン・アイクさんが1995年に開発したプログラミング言語だ。当時のWebブラウザは、動きのない、決まった内容の文字や画像しか表示できなかった。しかし、ユーザーの操作で表示する内容を変えたり、動かしたりしたいという要求が高まり、Webブラウザが直接プログラミング言語を実行するというアイディアが生まれた。そのための言語としてつくられたのがジャバスクリプトだ。

ジャバスクリプトを実行するしくみは、今ではWebブラウザだけでなく、サーバーでWebアプリケーションなどを動かすためにも使われている。実は、スクラッチ 3.0もジャバスクリプトだ。

ゲーム 工作 プログラミング



## ラズベリーパイ Raspberry Pi 大活用



キットの情報は  
KoKa Shop!へ  
定期購読者特典割引あり  
購入ページ



今回の内容はPythonが  
インストールされていれば  
普通のパソコンでもできるよ。

トで書かれているんだよ。

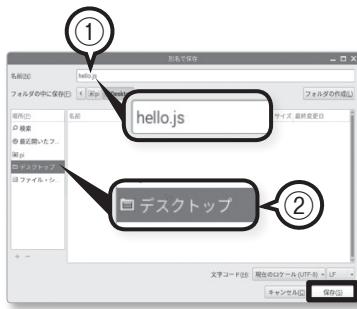
### ジャバスクリプトのプログラムを 書いてみよう

ジャバスクリプトのプログラムはテキストで書くので、スタートメニューの「アクセサリ」の中にある「Text Editor」を使おう。

エディターが開いたら、半角で次のように入力しよう。これは、「hello, world」という文字列をアラート(警告)のダイアログ(ウィンドウ)に表示するプログラムだ。

```
alert('hello, world');
```

ただ にゅうりょく かく にん  
正しく入力できたことを確認したら、「ファイル(F)」メニューから「保存(S)」を選んで保存しよう。名前には「hello.js」と入力(①)し、場所は「デスクトップ」を選ぼう(②)。



続いて、このプログラムを実行するWebページをHTML<sup>※1</sup>で書くよ。「ファイル(F)」メニューから「新規(N)」を選んだら、新しく開いたウィンドウに下のソースコードⒶを入力しよう。

※1 HTMLについては、2018年8月号(第17回)を見てみよう。以下のサイトから見ることもできるぞ。

<https://prog.kodomonokagaku.com/jibun/images/1808raspi.pdf>

正しく入力できたら、さっそく同じように「ファイル(F)」メニューから「保存(S)」を選んで保存する。今度は名前を「index.htm」にしよう(③)。場所は同じ「デスクトップ」だ(④)。



デスクトップにあるアイコン「index.htm」をダブルクリックすると、Webブラウザが開いて実行されるよ。「hello, world」とアラートが表示されたかな？ うまく動かない場合は、

hello.jsとindex.htmの内容をよく確認し、もし間違いが見つかったら、それを直して保存してから、index.htmを再読み込みしよう。



## Webサーバーで動かす

今はindex.htmを直接実行したけど、Webサーバー経由で実行することもできる。サーバーはターミナルから起動するので、スタートメニューの「アクセサリ」から「LXTerminal」を選ぼう。タスクバーの「>\_」のアイコンをクリックしてもいいよ。

ターミナルが開いたら、「cd」コマンドを使って、ディレクトリを「Desktop」に移動するよ。「pi@raspberrypi:~ \$」の後の「■」に次のように入力して「Enter」キーを押そう。

```
cd Desktop
```

移動したら、「pi@raspberrypi:~/Desktop \$」と返ってくるはずだ。続いて、パイソン(Python)に組み込まれているWebサーバーを起動するコマンドを入力しよう。「8000」はポート番号だ。

```
python -m SimpleHTTPServer 8000
```

起動に成功すると、以下が表示されるよ。

```
Serving HTTP on 0.0.0.0 port 8000 ...
```

### 【ソースコードⒶ】

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
    <script src="hello.js">
    </script>
  </body>
</html>
```

…この文書はHTMLであることを宣言  
…ここからHTML  
…ここからヘッダー(ここでは入れていないけどタイトルなど)  
…ヘッダーはここまで  
…ここから本体  
…「hello.js」というスクリプト(プログラム)を埋め込む  
…スクリプトはここまで  
…本体はここまで  
…HTMLはここまで

「…」の後ろの部分は説明なので入力しなくていいよ。

表示を確認したらWebブラウザーに戻り、アドレス欄に「localhost:8000」と入力して開こう。「localhost」は、今使っているコンピューターのこと、「8000」はさっき指定したポート番号だ。



## スクラッチの限界と拡張機能

スクラッチのプログラミングに慣れてくると、がんばれば何でもつくれる気がするよね。でも、スクラッチだけでは難しいこともある。今やったようなアラートを出すことはできないし、「( )の絶対値」のような返り値を返すブロック(レポーターブロック)を定義でつくることもできない。例えば、スクラッチにはべき乗がないので、2を3乗したいときは、「(((2)\*(2))\*(2))」のようにブロックをたくさん重ねる必要がある。

そこで用意されているのが拡張機能だ。この連載でも、「翻訳」や「音声合成」の拡張機能を使ったことがあるね※2。

拡張機能は、最初から用意されているものだけでなく、自分でつくることもできる。例えば、2020年8月号別冊付録「QRコードプログラミングBOOK」で紹介されていた「QRコード」は杉浦学先生がつくったものだ。このような拡張機能もジャバスクリプトでつくられているぞ。

※2 2019年6月号(第27回)を見てみよう。以下のサイトから見ることができるぞ。  
<https://prog.kodomonokagaku.com/jibun/images/1906raspi.pdf>

## 拡張機能をつくるてみよう

拡張機能を開発するための環境を用意するのはなかなか大変だ。そこで今回は、Sheep\_

makerさんという人がつくった「E羊icques」※3というmod(改造版)を使ってみよう。E羊icquesを使うと、ジャバスクリプトで書いた拡張機能のアドレスを入れるだけで、それを実行することができる。

今回は、べき乗を計算する拡張機能をつくりてみよう。「Text Editor」で「ファイル(F)」メニューから「新規(N)」を選んで、新しく開いたウィンドウに右ページのソースコード①を入力して「koka.js」という名前で保存しよう。

※3 文字化けではなく本当にこういう名前。

## E羊icquesの起動

Webブラウザーのアドレス欄に次のアドレスを入力して、E羊icquesを起動したら、左下の「拡張機能を追加」アイコンをクリックしよう。

<https://sheeptester.github.io/scratch-gui/>



拡張機能の一覧が開いたかな?  
次に、左上の「Choose an extension」をクリックしよう。



拡張機能のアドレスを入力するダイアログが開くので、次のように入力だ。

<http://localhost:8000/koka.js>

## 【ソースコード⑧】

```

class KoKa {
    constructor() {}

    getInfo() {
        return {
            id: 'koka',
            name: 'KoKa',
            blocks: [
                {
                    opcode: 'power',
                    blockType: Scratch.BlockType.REPORTER,
                    text: '[X] の [N] 乗',
                    arguments: {
                        X: {
                            type: Scratch.ArgumentType.NUMBER,
                            defaultValue: '2'
                        },
                        N: {
                            type: Scratch.ArgumentType.NUMBER,
                            defaultValue: '3'
                        }
                    }
                }
            ]
        }
    }

    power(args) {
        return Math.pow(args.X, args.N); ...Mathオブジェクトに定義されているpow関数を使ってべき乗を計算
    }
}

Scratch.extensions.register(new KoKa()); ...KoKaのインスタンス(実体)をつくるて拡張機能を登録

```

すると、「KoKa」というカテゴリーが追加され  
て(5)、「(2) の (3) 乗」というブロックができ  
ている!



このブロックをクリックすると、「8」が返ってく  
るね。もちろん、数字を変えたり、他のブロック  
と組み合わせて使ったりすることもできるよ。



普段、ブロックでプログラムを書いている人に  
は、ちょっと難しい内容だったかもしれないけれ  
ど、拡張機能の可能性と、それをテキスト言語  
で書く意味がわかつてもらえたかな。自分で拡  
張機能をつくれるようになると、ほとんどどんな  
ことでもスクラッチでできるようになるぞ。今  
のスクラッチの機能に不満がある人はぜひチャレ  
ンジしてほしい。

本格的に拡張機能をつくるてみたい人には、石  
原淳也さんが書いた「Scratch を改造しよう」※4  
がおすすめだ。もちろん、ジャバスクリプトなど  
の知識も必要になるので、その入門書もあわせ  
て読もう。

※4 <https://otona-scratch.champierre.com/books/1/posts>  
(有料コンテンツです)

今回の記事は、石原淳也さんが書いた以下のブログ記事を参考に  
しました。  
<https://blog.champierre.com/1133>