

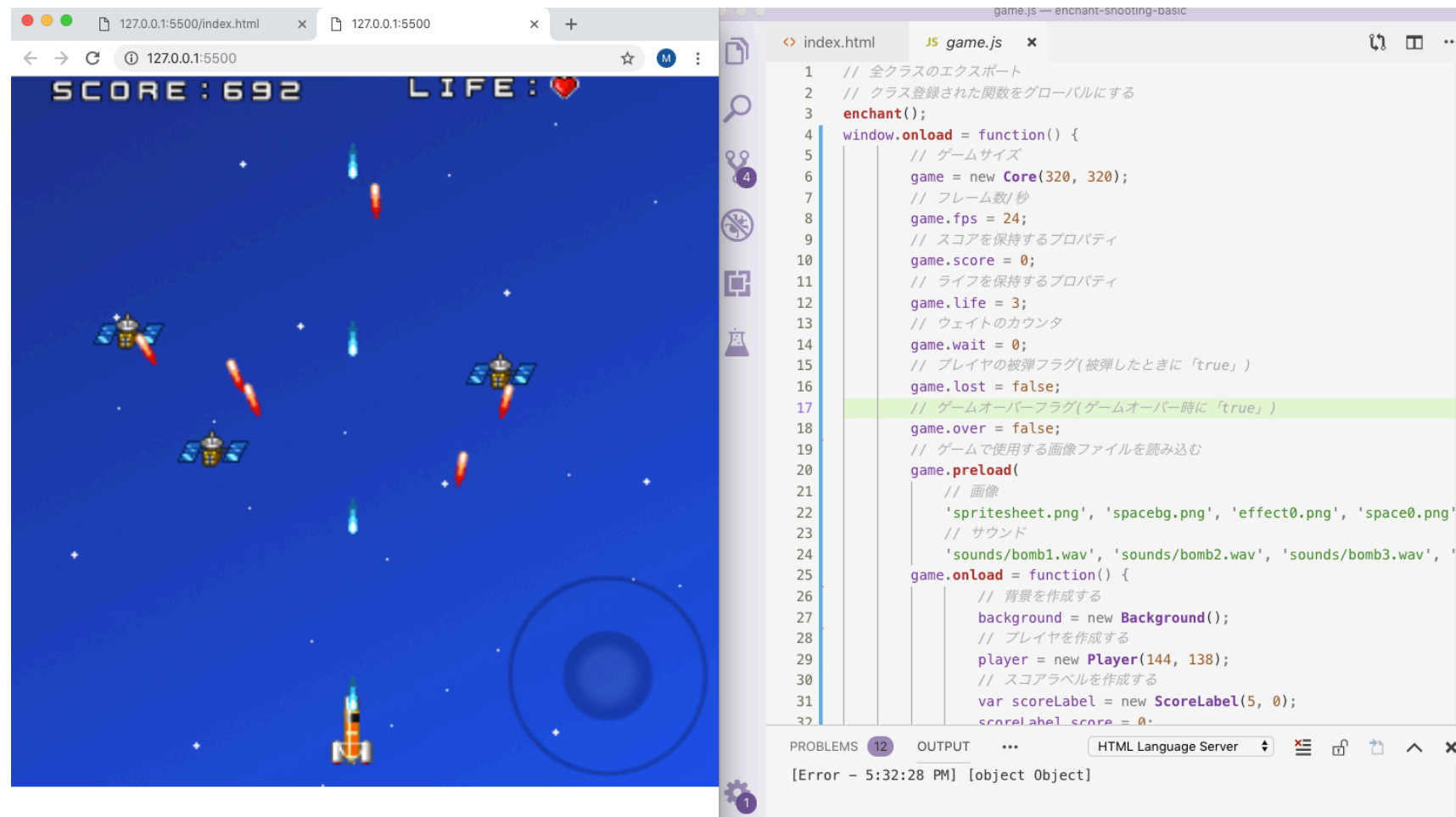
JavaScriptでシューティングゲーム を作ろう！

第2回

コードをながめる

更新日: 2019.3.21

CoderDojo Nada



JavaScriptのオブジェクト

これがオブジェクトの基本です

{ }

これがオブジェクトの作り方

```
let a = {}
```

オブジェクトの中身が x という名前で 1 という値をセット

```
let a = {x : 1}
```

JavaScriptのオブジェクト

Chromeの開発者ツールのConsoleで

```
let a = {}
```

```
a = {x : 1}
```

```
a.x
```

```
b = {y : function(){console.log("hello")}}
```

```
b.y()
```

```
let crab = {}
```

```
crab = {say:function(m){console.log(m)}}
```

```
crab.say()
```

```
crab.say2 = function(m){console.log(m+m)}
```

```
crab.say2("abc")
```

```
> let a = {}  
< undefined  
  
> a = {x:1}  
< ▶ {x: 1}  
  
> a.x  
< 1  
  
> b = {y:function(){console.log("hello")}}  
< ▶ {y: f}  
  
> b.y()  
hello  
< undefined  
  
> let crab = {}  
< undefined  
  
> crab = {say:function(m){console.log(m)}}  
< ▶ {say: f}  
  
> crab.say("abc")  
abc  
< undefined  
  
> crab.say2 = function(m){console.log(m+m)}  
< f (m){console.log(m+m)}  
  
> crab.say2("abc")  
abccabc
```

ゲームスタート！

1. ゲーム部品の準備
2. イベント: window.onload (ウィンドウがロードされる時)
 - 2.1 ゲームの初期設定
 - 2.2 イベント: game.onload (ゲームがロードされる時)
 - 2.2.1 背景を作る
 - 2.3 ゲームをスタートする

```
1 // 1 ゲーム部品の準備
2 enchant(); // 全クラスのエクスポート クラス登録された関数をグローバルにする
3 let game, background; // グローバル変数の定義
4 // 2 イベント: window.onload (ウィンドウがロードされる時)
5 window.onload = function() {
6     // 2.1 ゲームの初期設定
7     game = new Core(320, 320); // ゲームサイズ
8     game.fps = 24; // フレーム数/秒
9     game.preload('spacebg.png'); // 画像のロード
10    // 2.2 イベント: game.onload (ゲームがロードされる時)
11    game.onload = function() {
12        // 2.2.1 背景を作る
13        background = new Background();
14    };
15    // 2.3 ゲームをスタートする
16    game.start();
17 };
```

背景クラスを定義する

3 背景クラスをenchant.Spriteクラスを元に定義する

3.1 スプライトの初期設定をする

3.1.1 イベント: enterframe(新しいフレームを描画する時):

3.1.1.1 スプライトのy座標を1ずつ下げる(プラスする)

3.1.1.2 ウィンドウの左上までいったら、最初の位置に戻す

3.1.2 ゲームに背景を追加する

```
18 // 3 背景クラスをenchant.Spriteクラスを元に定義する
19 ㊦ let Background = enchant.Class.create(enchant.Sprite, { // enchant.Spriteを継承
20     // 3.1 スプライトの初期設定をする
21     ㊦ initialize: function() {
22         enchant.Sprite.call(this, 320, 640);
23         this.x = 0; // x座標
24         this.y = -320; // y座標
25         this.frame = 0; //先頭のフレーム
26         this.image = game.assets['spacebg.png'];
27         // 3.2 イベント: enterframe(新しいフレームを描画する時):
28     ㊦ this.addEventListener('enterframe', function() {
29         // 3.2.1 スプライトのy座標を1ずつ下げる(プラスする)
30         this.y++;
31         // 3.2.2 ウィンドウの左上までいったら、最初の位置に戻す
32         // y座標が 0 以上になったら、y座標を最初の位置 -320 に戻す
33         if (this.y >= 0) this.y = -320;
34     });
35     // 3.3 ゲームに背景を追加する
36     game.rootScene.addChild(this);
37 }
38 });
```

背景をつくる

「new Background」とすることで、
「Background」オブジェクトの「initialize」関数を呼び出す

```
18 // 3 背景クラスをenchant.Spriteクラスを元に定義する
19 let Background = enchant.Class.create(enchant.Sprite, { // enchant.Spriteを継承
20 // 3.1 スプライトの初期設定をする
21 initialize: function() {
22     enchant.Sprite.call(this, 320, 640);
23     this.x = 0; // x座標
24     this.y = -320; // y座標
25     this.frame = 0; // 先頭のフレーム
26     this.image = game.assets['spacebg.png'];
27 // 3.2 イベント: enterframe(新しいフレームを描画する時):
28 this.addEventListener('enterframe', function() {
29     // 3.2.1 スプライトのy座標を1ずつ下げる(プラスする)
30     this.y++;
31     // 3.2.2 ウィンドウの左上までいったら、最初の位置に戻す
32     // y座標が 0 以上になったら、y座標を最初の位置 -320 に戻す
33     if (this.y >= 0) this.y = -320;
34 });
35 // 3.3 ゲームに背景を追加する
36 game.rootScene.addChild(this);
37 }
38 });
```

```
1 // 1 ゲーム部品の準備
2 enchant(); // 全クラスのエクスポート クラス登録された関数をグローバルにする
3 let game, background; // グローバル変数の定義
4 // 2 イベント: window.onload (ウィンドウがロードされる時)
5 window.onload = function() {
6     // 2.1 ゲームの初期設定
7     game = new Core(320, 320); // ゲームサイズ
8     game.fps = 24; // フレーム数/秒
9     game.preload('spacebg.png'); // 画像のロード
10    // 2.2 イベント: game.onload (ゲームがロードされる時)
11    game.onload = function() {
12        // 2.2.1 背景を作る
13        background = new Background();
14    };
15    // 2.3 ゲームをスタートする
16    game.start();
17 };
```

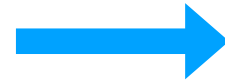
イベント処理

window.onload
ウィンドウがロード



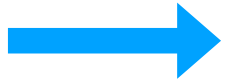
ウィンドウがロードされた直後にする処理

game.onload
ゲームがロード



ゲームオブジェクトがロードされた直後にする処理

enterframe
新しいフレームが描画

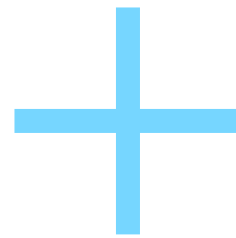


新しいフレームが描画される直前にする処理
フレーム：画面を更新する単位

クラスをつくる

```
enchant.Class.create(親オブジェクト, { 子独自の拡張機能 })
```

親オブジェクトの機能

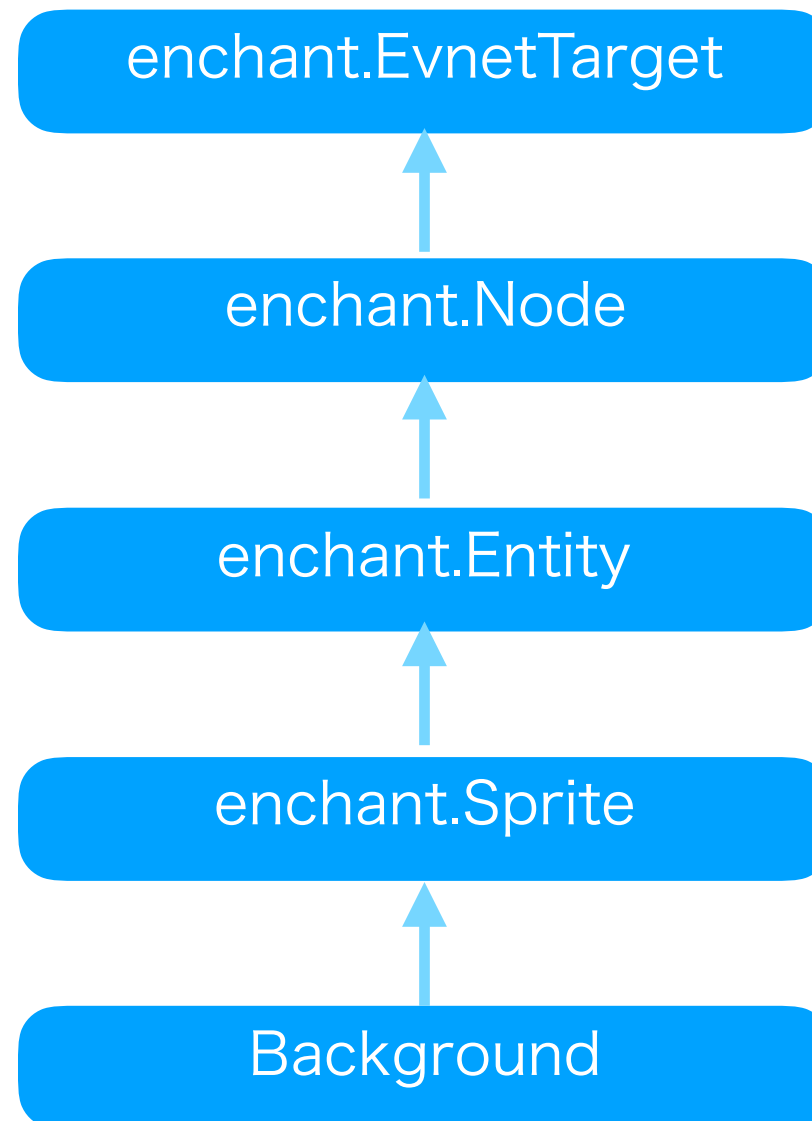


子独自の拡張機能

継承をたどってみよう

Visual Studio Codeの左側にある「エクスプローラ」で
「enchant.js」を開いてみよう

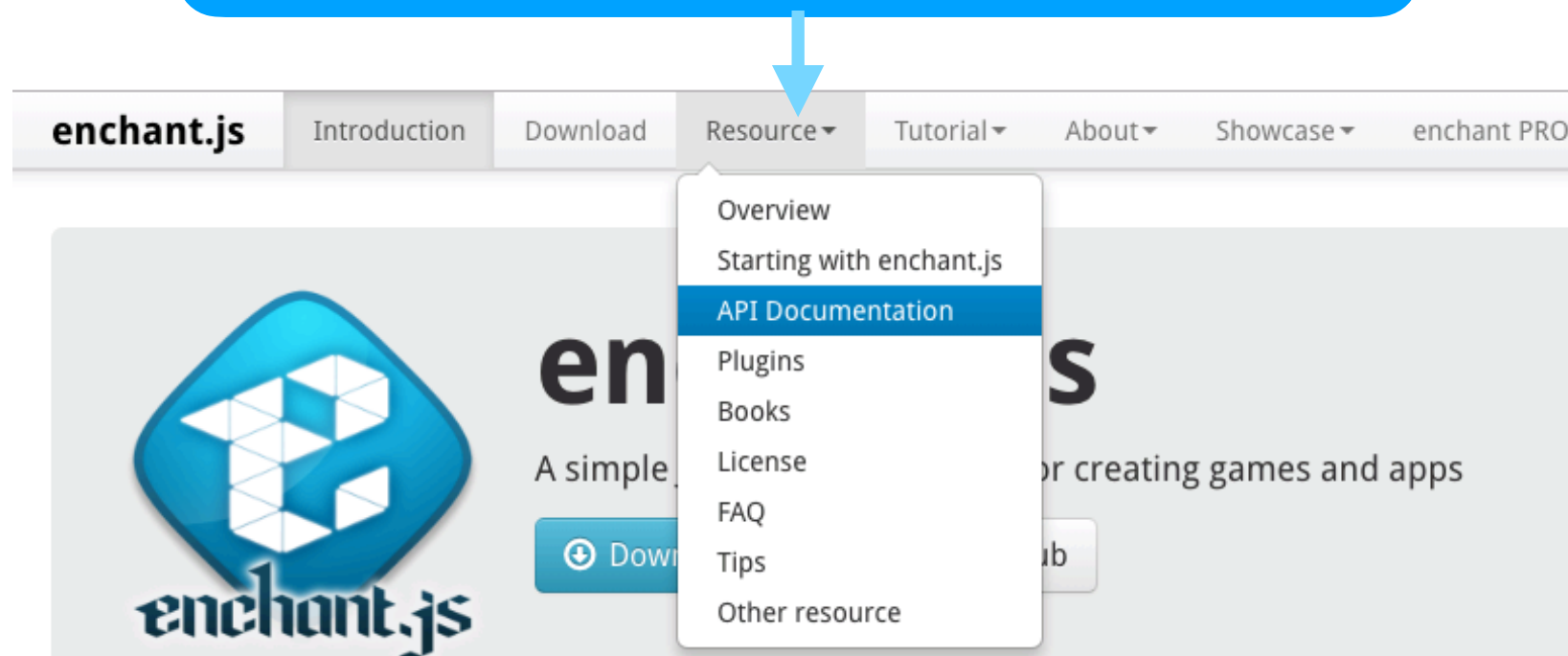
```
enchant.Class.create(親オブジェクト, { 子独自の拡張機能 })
```



調べる

enchant.jsのWebサイト
<http://enchantjs.com/ja/>

クラス、APIなど（enchant.jsの中身について）



API Documentation

Document (v0.8.3, latest)

- Japanese
 - <http://wise9.github.com/enchant.js/doc/core/ja/>
 - <http://wise9.github.com/enchant.js/doc/plugins/ja/> (with plugins)
- English
 - <http://wise9.github.com/enchant.js/doc/core/en/>
 - <http://wise9.github.com/enchant.js/doc/plugins/en/> (with plugins)

このリンクから入る

調べる

Classes

classes | files

[_global_](#)
[enchant . Action](#)
[enchant . ActionEventTarget](#)
[enchant . BinaryInputManager](#)
[enchant . BinaryInputSource](#)
[enchant . CanvasLayer](#)
[enchant . CanvasScene](#)
[enchant . Class](#)
[enchant . Core](#)
[enchant . Deferred](#)
[enchant . DOMScene](#)
[enchant . DOMSound](#)
[enchant . Easing](#)
[enchant . Entity](#)
[enchant . ENV](#)
[enchant . Event](#)
[enchant . EventTarget](#)
[enchant . Game](#)
[enchant . Group](#)
[enchant . InputManager](#)
[enchant . InputSource](#)
[enchant . KeyboardInputManager](#)
[enchant . KeyboardInputSource](#)
[enchant . Label](#)
[enchant . LoadingScene](#)
[enchant . Map](#)
[enchant . Node](#)
[enchant . ParallelAction](#)
[enchant . Scene](#)
[enchant . Sprite](#)
[enchant . Surface](#)
[enchant . Timeline](#)

さっき出てきたクラス
を調べよう！
enchant.Sprite
enchant.Entity
enchant.Node
enchant.EventTarget

Class

enchant.Sprite

画像表示機能を持ったクラス. Entity を継承している.

Defined in: [enchant.js](#)

Extends [enchant.Entity](#)

Class Summary

Constructor Attributes

Constructor Name and Description

[enchant.Sprite](#)(width, height)

Field Summary

Field Attributes

Field Name and Description

[frame](#)
表示するフレームのインデックス.

[image](#)
Spriteで表示する画像.

Fields borrowed from class [enchant.Entity](#):

[backgroundColor](#), [buttonMode](#), [buttonPressed](#), [compositeOperation](#), [debugColor](#), [height](#), [opacity](#), [originX](#), [originY](#), [rotation](#), :

Fields borrowed from class [enchant.Node](#):

[age](#), [parentNode](#), [scene](#), [x](#), [y](#)

Method Summary

Methods borrowed from class [enchant.Entity](#):

[disableCollection](#) [enableCollection](#) [intersect](#) [rotate](#) [scale](#) [within](#)

Methods borrowed from class [enchant.Node](#):

[moveBy](#) [moveTo](#)

Methods borrowed from class [enchant.EventTarget](#):

[addEventListener](#) [clearEventListener](#) [dispatchEvent](#) [on](#) [removeEventListener](#)

利用する（ダウンロード）

enchant.jsのWebサイト
<http://enchantjs.com/ja/>

enchant.js Introduction Download Resource Tutorial About Sho

Download

このタブを選ぶ

Download

Download latest version of enchant.js (v0.8.3, recommended)

zip アーカイブには画像ファイル、ドキュメントが含まれています。

github

github repository page: <https://github.com/wise9/enchant.js>

The latest repository can also be cloned via git, using the following code (Non-prebuilt files, you need to `npm install` and `grunt`):

```
git clone https://github.com/wise9/enchant.js.git
```

or [Download zipball of development branch](#)

過去のバージョン

github から過去のバージョンにアクセスすることができます:
<https://github.com/wise9/enchant.js/tags>

画像素材

enchant.js プロジェクトでは、ゲーム開発にすぐに取り掛かれるような画像素材ライブラリを提供しています。

これらの画像は、上記のリンクからダウンロードできるパッケージに含まれています。詳しくは [Image Materials](#) のページをご覧ください。



音声素材

音声素材のダウンロードはこちらからどうぞ (zipファイル)。

This post is also available in: [英語](#), [フランス語](#), [スペイン語](#), [ポルトガル語 \(ブラジル\)](#)