テーマ1
enchant.js で ゲーム作り
(2)

2015/04/21

苫小牧高専 ソフトウェアテクノロジー部

プログラミングの授業にて

・授業のプログラミング（C言語）の流れ

1. コードを書く

2. コンパイルする

```
gcc -o hello hello.c
```

3. 実行する

```
./hello
```

コンパイラ言語

・ソースコードをコンパイル（翻訳）して実行する。

・C言語、C++、Java、C#など

・プラットフォームに依存する
→ Windowsでコンパイルしたファイルは
Macじゃ動かない

JavaScript

- コンパイル等することなく
  ブラウザ（Internet Exploler）が直接実行する
- プラットフォーム非依存
  → WindowsでもMacでも、IEでもChromeでも
  Firefoxでも（基本的には）動く。

授業と部活で学んだことが
ごちゃごちゃにならないように……

- C言語とJavaScriptは似ている部分も多いが、
  基本的に異なるということに注意！

- 今後、みなさんの授業の進度に合わせて
  補足説明していきます。

前回のまとめ

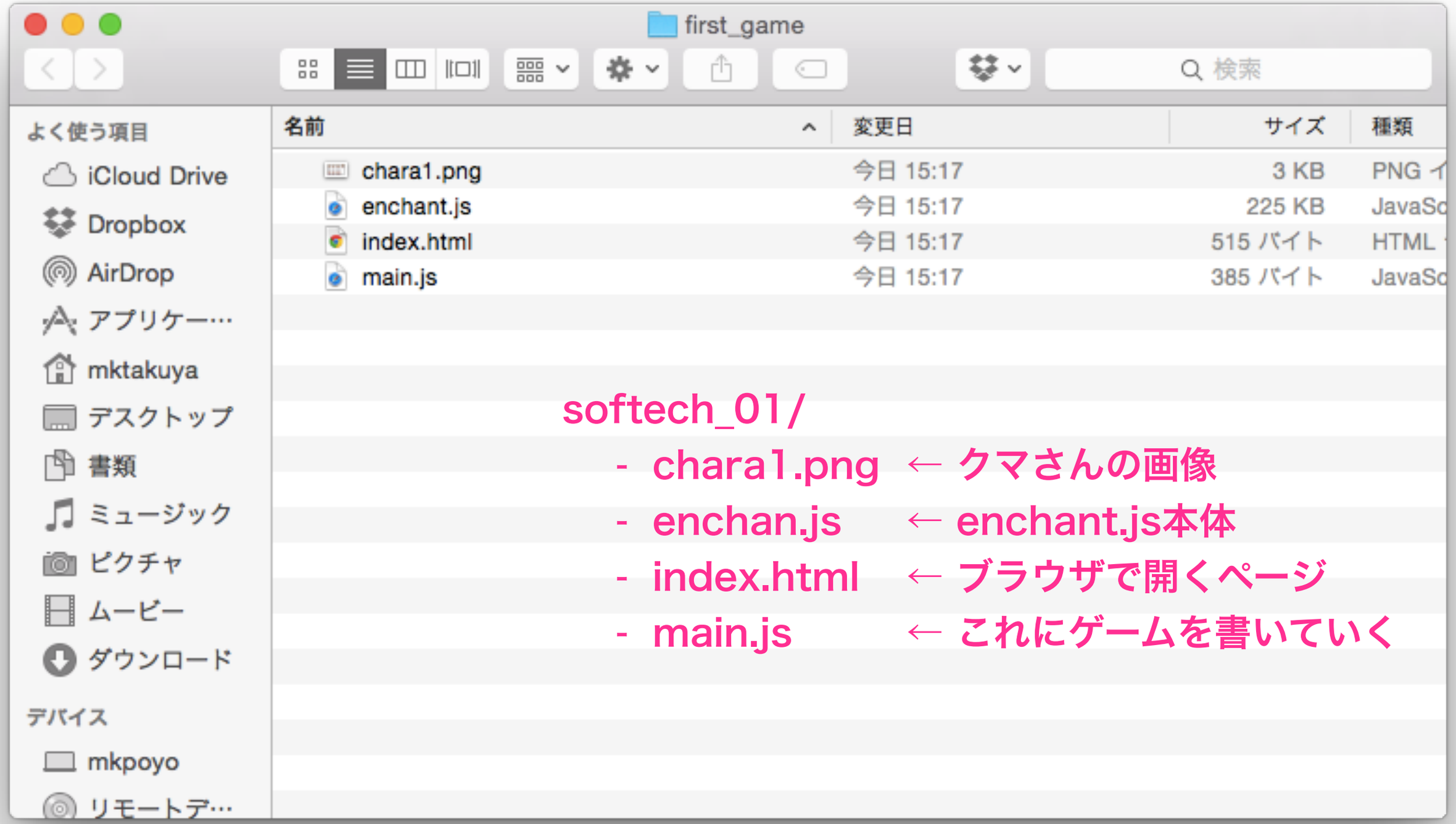
開発に必要なツールについて

- enchant.js本体
  前回で配布済み。

- エディタ（プログラムを書くソフトウェア）
  TeraPad を使います。

- ブラウザ（プログラムを実行するソフトウェア）
  Internet Exproler 11

配布したもの


first_game
検索
よく使う項目
iCloud Drive
Dropbox
AirDrop
アプリケー…
mktakuya
デスクトップ
書類
ミュージック
ピクチャ
ムービー
ダウンロード
デバイス
mkpoyo
リモートデ…
名前
変更日
サイズ
種類
chara1.png 今日 15:17 3 KB PNG イ
enchant.js 今日 15:17 225 KB JavaSc
index.html 今日 15:17 515 バイト HTML
main.js 今日 15:17 385 バイト JavaSc
softech_01/
- chara1.png ← クマさんの画像
- enchan.js ← enchant.js本体
- index.html ← ブラウザで開くページ
- main.js ← これにゲームを書いていく

ブラウザで開くファイル
index.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset=“UTF-8">
    <title>Takuyaのゲーム</title> ← ページのタイトル
    <meta http-equiv="x-ua-compatible" content="IE=Edge">
    <meta name="viewport" content="width=device-width, user-scalable=no">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <script type="text/javascript" src="./enchant.js"></script>
    <script type="text/javascript" src="./main.js"></script>
    <style type="text/css">
      body {
        margin: 0;
        padding: 0;
      }
    </style>
  </head>
  <body>
  </body>
</html>
```

ゲーム本体
main.js

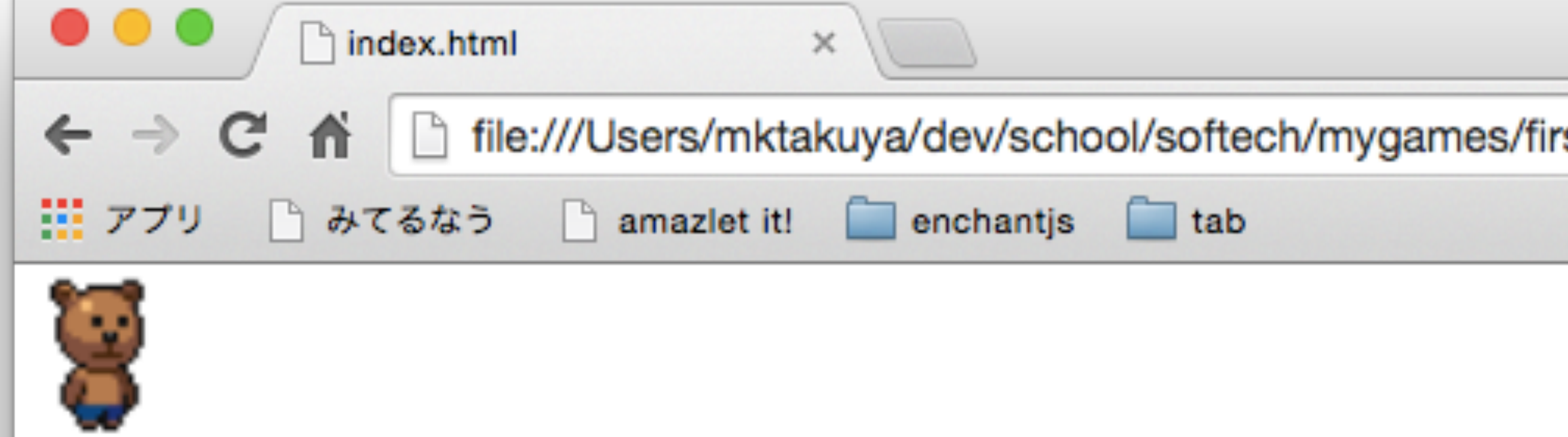
```
enchant(); ← enchant.js を使うためのおまじない。

window.onload = function() {
    var core = new Core(320, 320); ← ゲームのCore（核）となる部分

    core.onload = function() {
        ここに処理を書いていく。
    }
    core.start(); ← ゲームスタート！
}
```

前回のテーマ

「画像を動かしてみる」

main.js
～クマさんを表示する～

```
enchant();

window.onload = function() {
    var core = new Core(320, 320);
    core.preload('chara1.png');    ← ① 画像を読み込む

    core.onload = function() {
        var bear = new Sprite(32, 32);                ← ② 変数を定義する
        bear.image = core.assets['chara1.png'];    ← ③ ファイル名を指定する
        bear.x = 0;  ← ④ 座標を指定する
        bear.y = 0;

        core.rootScene.addChild(bear);  ← ⑤ シーンにbearを追加する
    }
    core.start();
}
```


index.html
file:///Users/mktakuya/dev/school/softech/mygames/firs
アプリ
みてるなう
amazlet it!
enchantjs
tab

クマさん現る！

画像を表示する
まとめ

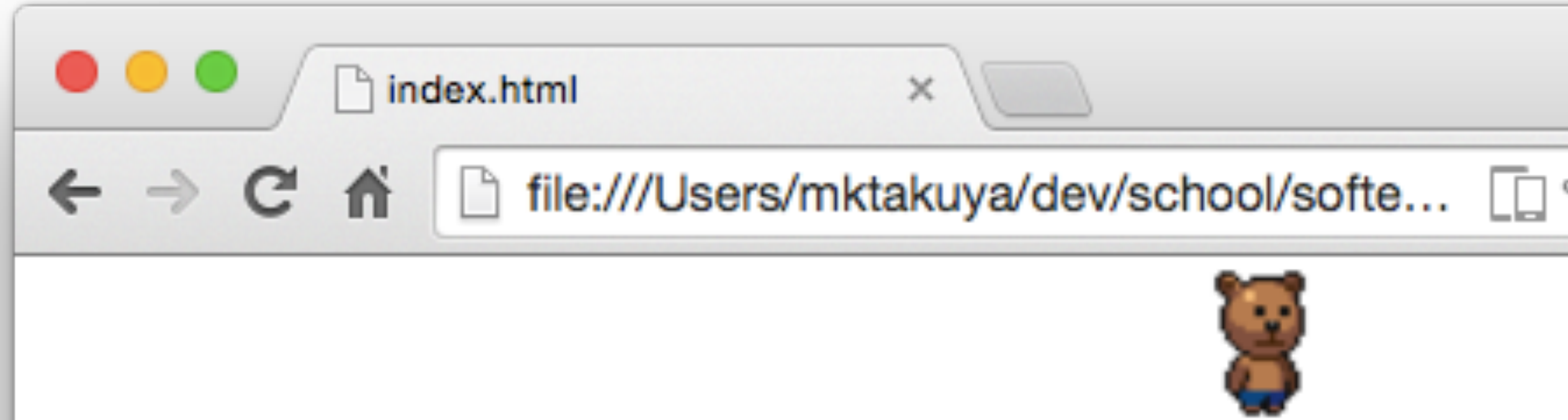
- ```
  core.onload = function() {
      // ここに処理を書いていく
  }
  ```

- ゲーム画面に画像を表示するには、
  ① 使いたい画像を読み込む
  ② Sprite()を使って変数を作る
  ③ 画像を変数にセット
  ④ 座標をセット
  ⑤ rootSceneに追加
  ```

画像を動かす

こいつ…動くぞ！？

main.js
画像を動かす

```
enchant();

window.onload = function() {
    var core = new Core(320, 320);
    core.preload('chara1.png');

    core.onload = function() {
        var bear = new Sprite(32, 32);
        bear.image = core.assets['chara1.png'];
        bear.x = 0;
        bear.y = 0;

        core.rootScene.addChild(bear);

        bear.addEventListener('enterframe', function() {
            this.x += 3;
        });
    }
    core.start();
}
```

← ① フレームが進むごとに実
（おまじない）

← ② bearのX座標を
3進める



クマさん動く！

画像を動かす
まとめ

- ゲームプログラミングでは、「あるイベントが発生した時のみ実行する処理」がよく出てくる。

- フレームが進んだ時の処理は、

```
bear.addEventListener('enterframe', function(){
        // ここに処理を書いていく
});
```

画像をいろいろ動かす

回転させたり、拡大させたり

画像をいろいろ動かす
まとめ

- ```
  core.onload = function() {
      // ここに処理を書いていく
  }
  ```
- 移動（this.xやthis.y）以外にも、this.rotate()やthis.scale()などを使って画像を操作することができる。
  ```

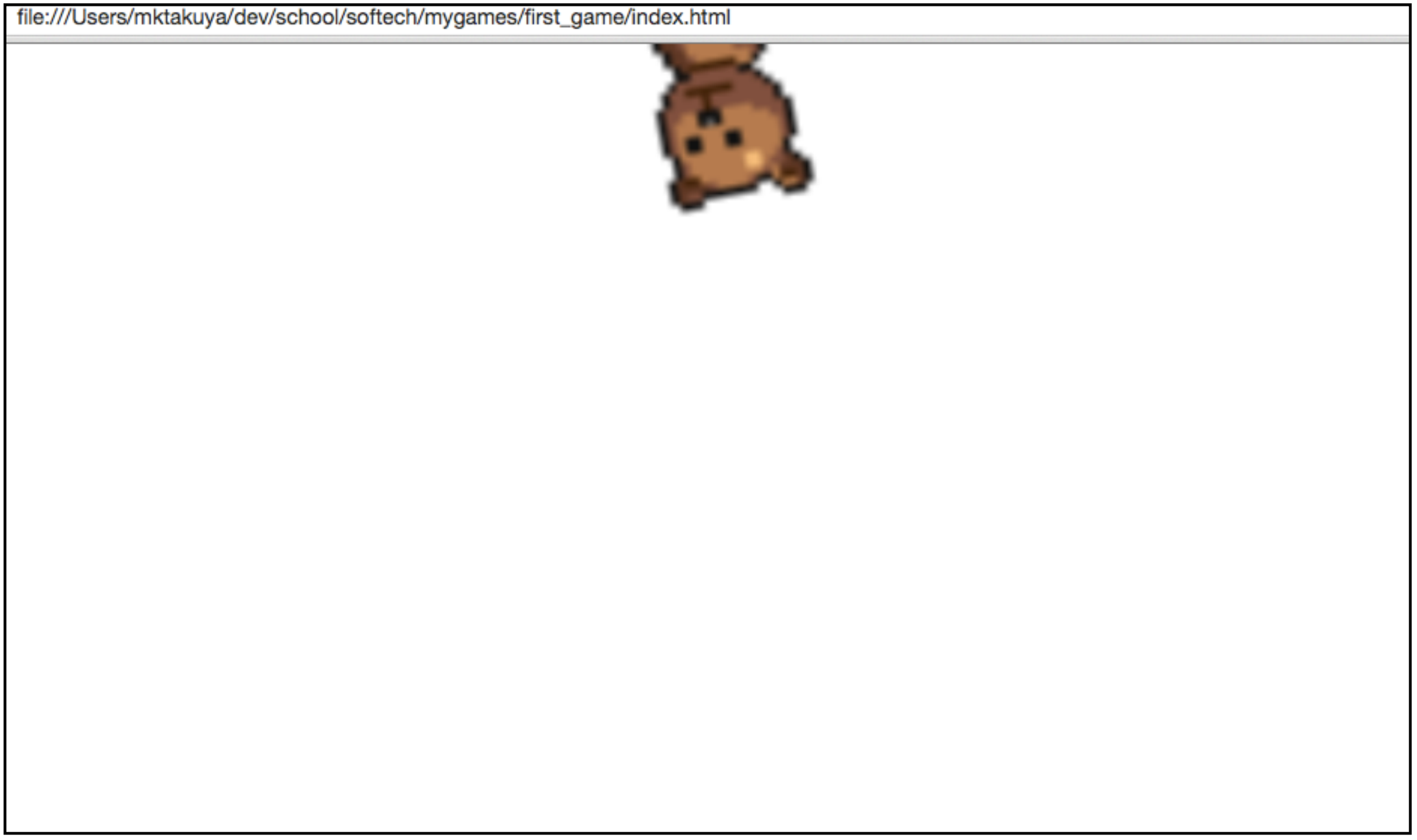


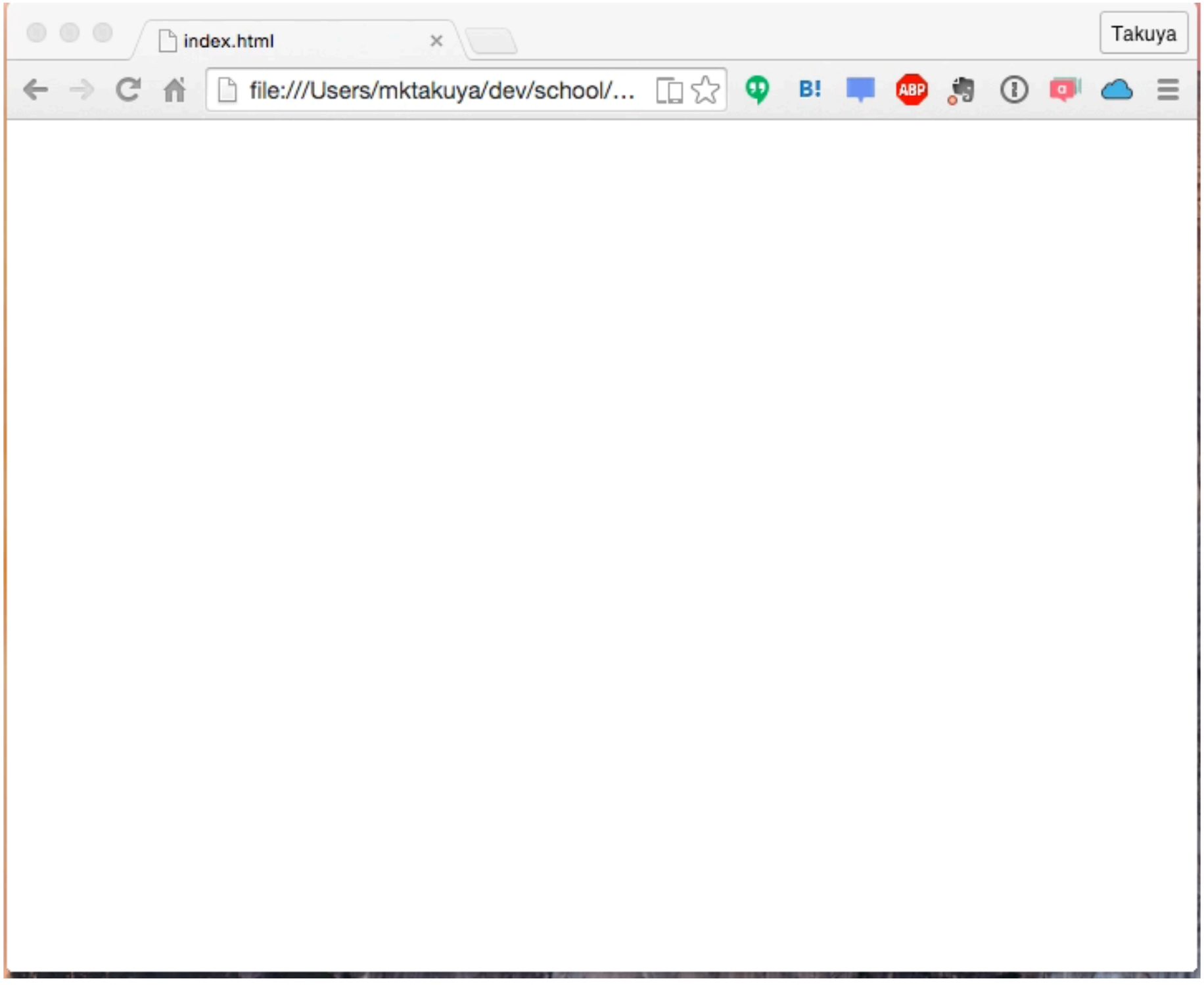
図1．回転しながら巨大化していく不審物

「クマさんが端っこで消えちゃう問題」

が

まだ未解決な件

クマさん端っこで消える問題



クマさん端っこで消える問題

- bear.x が 320を超えた時、
  bear.xを0にリセットすれば良いのでは？

クマさん端っこで消える問題

- もし、「bear.x が 320を超えた」ならば

- bear.xを0にリセットする

  bear.x = 0;

クマさん端っこで消える問題

・もし、「bear.x > 320」なら

・bear.xを0にリセットする

bear.x = 0;

if文

```
if（条件）{
   処理;
}
```

クマさん端っこで消える問題

・もし、「**bear.x > 320**」なら

↑ **条件**

・bear.xを0にリセットする

bear.x = 0;

↑ **処理**

if文

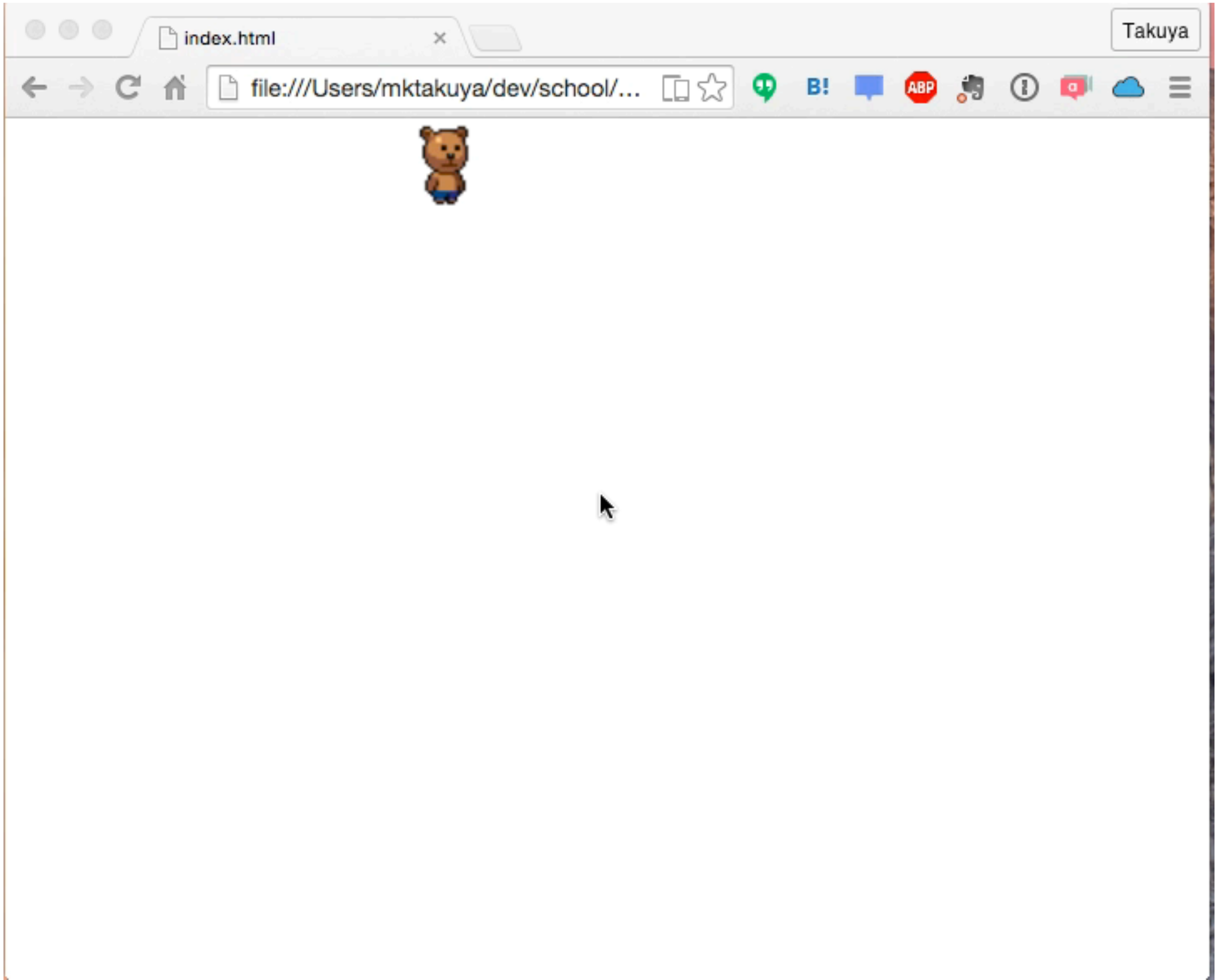
```
                ↓ this.x が 320を超えた時
if (this.x > 320) {
   this.x = 0;
                ↑ this.x に 0 を設定する。
}
```

bearのX座標が320を超えたら0にリセット

```
enchant();

window.onload = function() {
    var core = new Core(320, 320);
    core.preload('chara1.png');

    core.onload = function() {
        var bear = new Sprite(32, 32);
        bear.image = core.assets['chara1.png'];
        bear.x = 0;
        bear.y = 0;

        core.rootScene.addChild(bear);

        bear.addEventListener('enterframe', function() {
            this.x += 3;

            if (this.x > 320) {
                this.x = 0;
            }
        });
    }
    core.start();
}
```

クマさん端っこで消える問題

☆解☆決☆



課題2

課題2

1. クマさんを縦（y軸方向）にも動かしてみよう。

2. クマさんが上下方向に見きれた時の処理を追加してみよう。

今日のテーマ(1)

キャラクターの

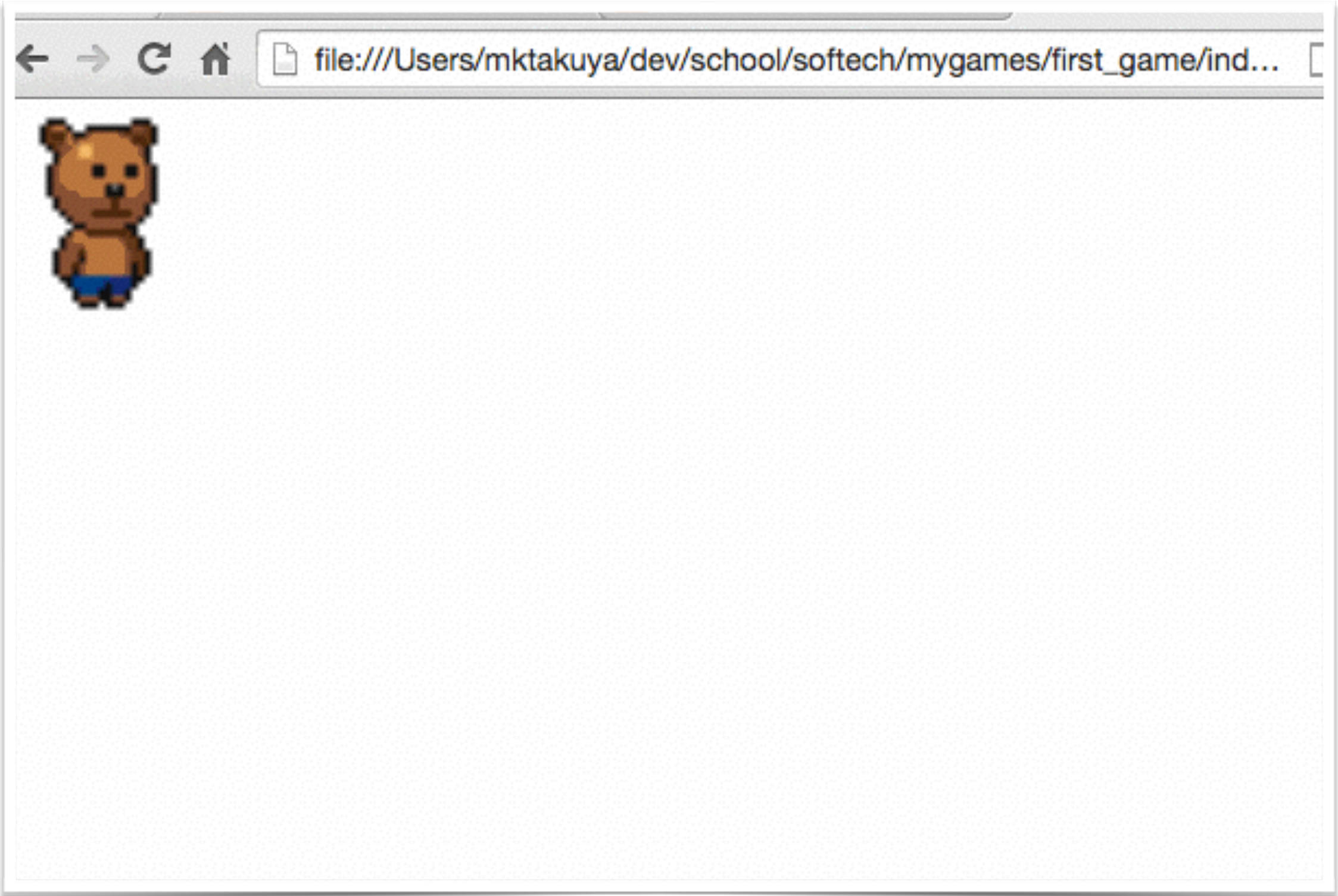
アニメーションを作ってみる

main.js

```
enchant();

window.onload = function() {
    var core = new Core(320, 320);
    core.preload('chara1.png');

    core.onload = function() {
        var bear = new Sprite(32, 32);
        bear.image = core.assets['chara1.png'];
        bear.x = 0;
        bear.y = 0;

        core.rootScene.addChild(bear);

        bear.addEventListener('enterframe', function() {
            this.x += 3;
            this.frame = this.age % 3;

            if (this.x > 320) {
                this.x = 0;
            }
        });
    }
    core.start();
}
```

```
this.frame = this.age % 3;
```

クマが歩いた！

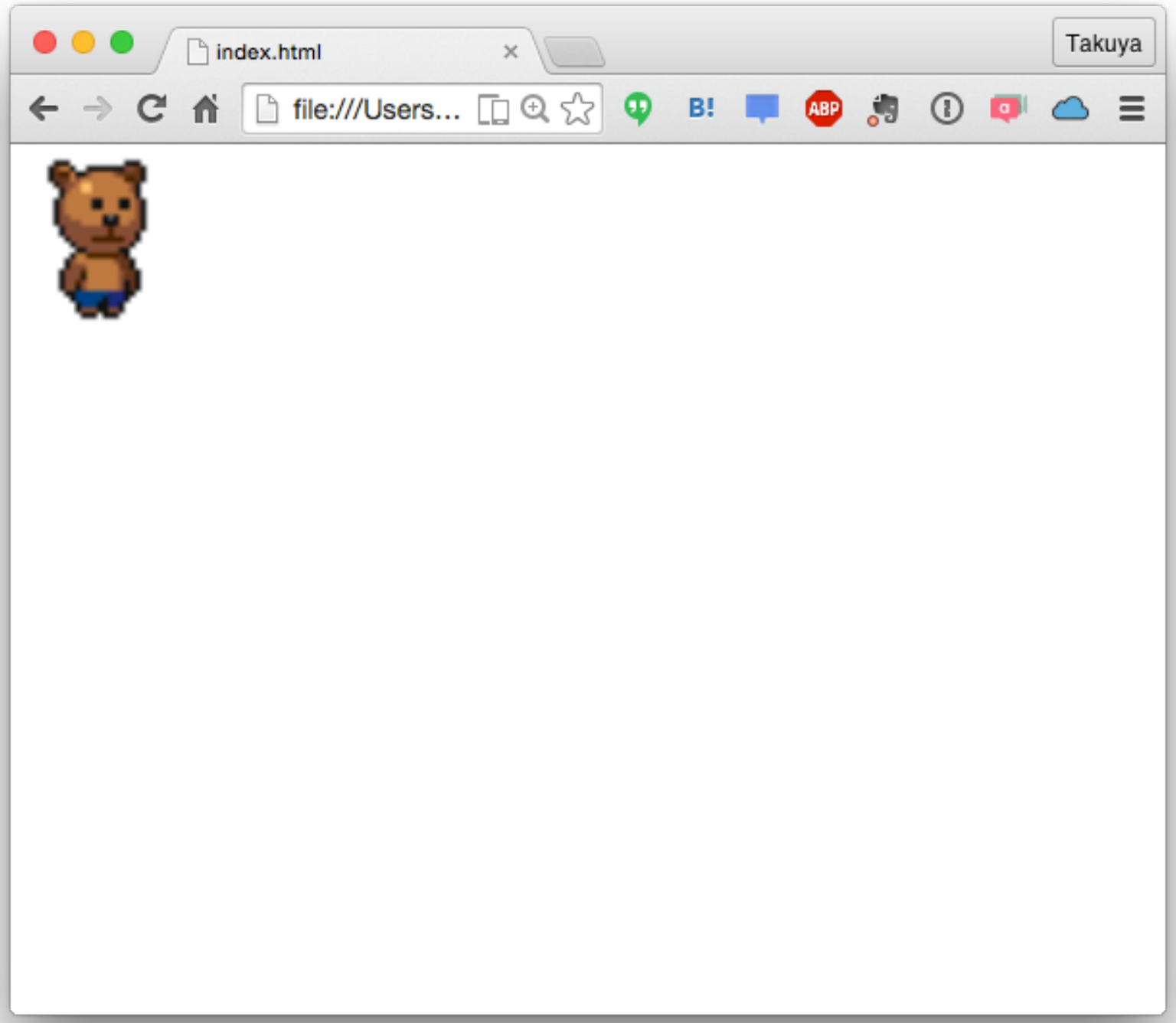

file:///Users/mktakuya/dev/school/softech/mygames/first_game/ind…

画像のフレーム

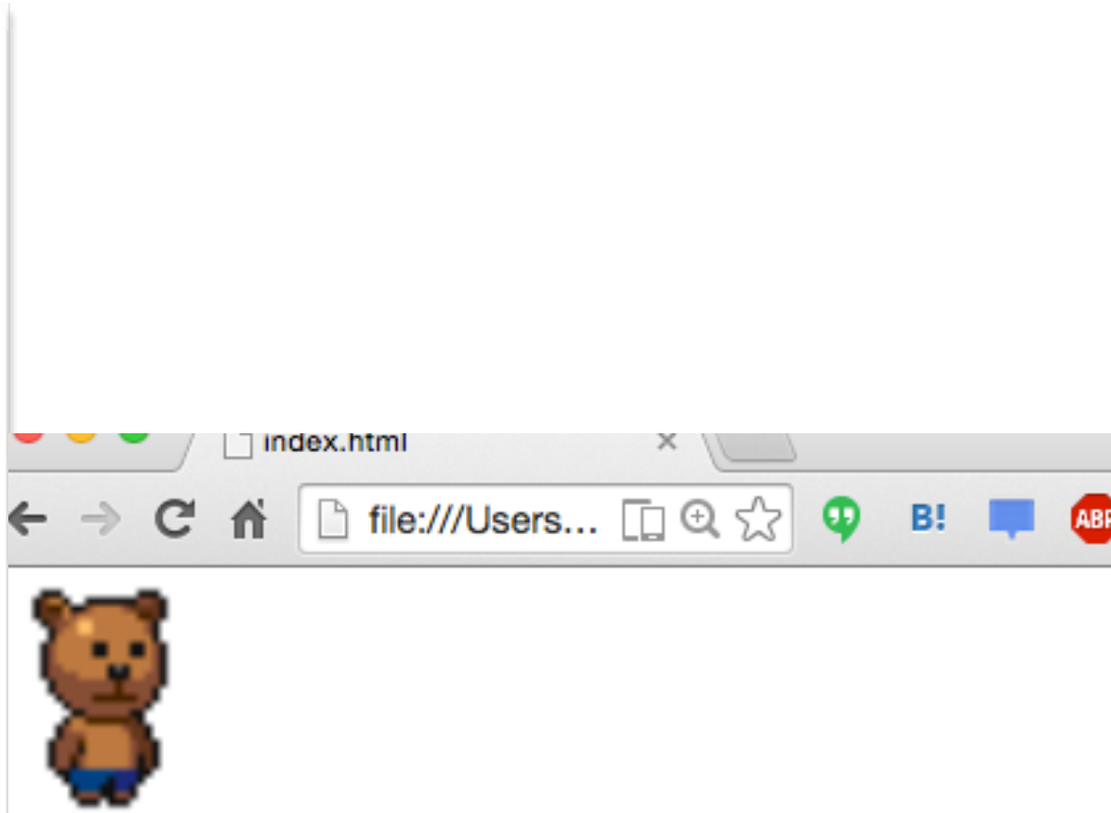
について

chara1.png

ゲーム画面

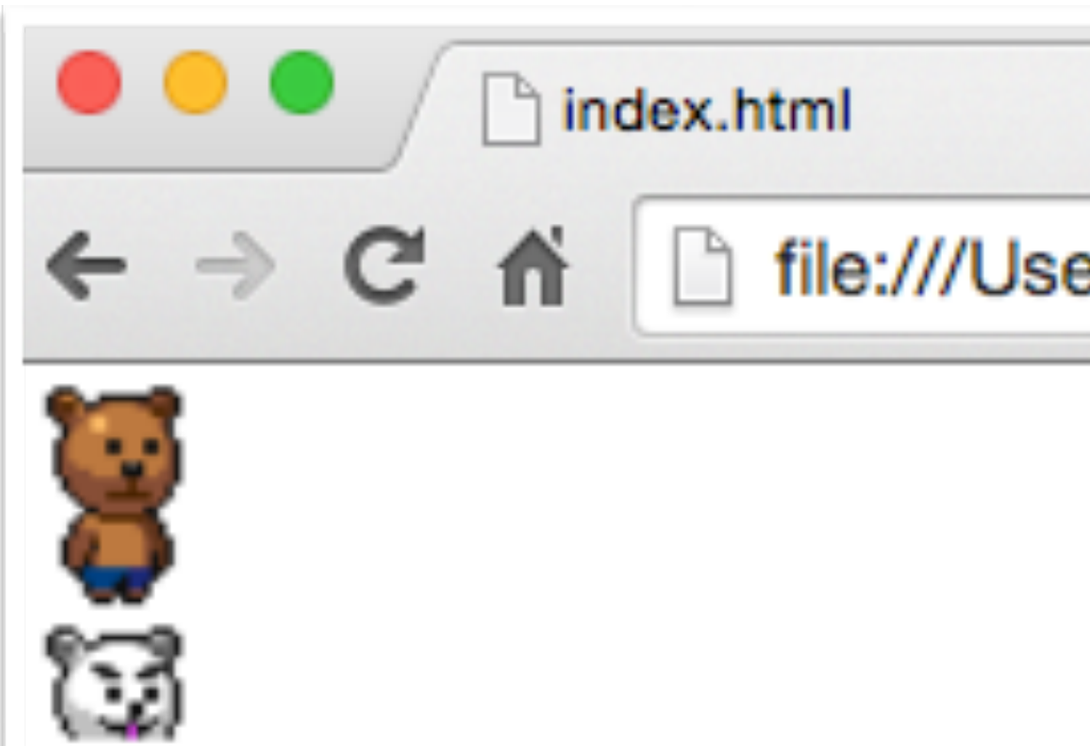


chara1.pngの
一部分（フレーム）を切り抜いて表示している




```
var bear = new Sprite(32, 32);
// この画像は 32px で切り取ればOK
```

chara1.pngの
一部分（フレーム）を切り抜いて表示している




```
var bear = new Sprite(32, 50);
// 切り抜くサイズを間違えると大変なことになるので注意
```

フレームという概念について

- bear.frame の 値をいじると、32pxで切り取ったうちのどこの部分を表示するかを選べる。

- 初期値は0

- bear.frame = [数字];
  で指定する

- また、bear.ageには
  キャラクタが動き始めてから
  何フレーム目かが入っている。

- bear.frameを0 → 1 → 2 → 0 → 1 → 2 → …… と繰り返せば、
  クマさんが走っているように見える。

main.js

```
this.frame = this.age % 3;


this.frameに、
this.age / 3のあまりを入れている。


/*
  “A ÷ B = Q あまり R”とすると、
  Qは A / Bで、
  Rは A % Bで求められる。
*/
```

課題

1. 白クマさんを表示してみよう

2. 白クマさんを歩かせてみよう

今日のテーマ(2)

「キャラクターを操作してみる」

enchant.jsの操作

- キーボードからの操作（PC向け）
- タッチでの操作（スマホ向け）
- 今回は簡略化のため、キーボードからの操作のみにします。

仕様決め

- キーボードの矢印キー（←↓↑→）で
  操作することにする。

- キーボード入力は core.input.xxx で
  受け取ることができる。

- 矢印キーは、
  core.input.left
  core.input.right
  core.input.up
  core.input.down

main.js
～クマさんをコントロールする～

```
// 省略

        bear.addEventListener('enterframe', function() {
            this.frame = this.age % 3;

                if (core.input.left) {
                    this.x -= 3;
                }
                if (core.input.right) {
                    this.x += 3;
                }
                if (core.input.up) {
                    this.y -= 3;
                }
                if (core.input.down) {
                    this.y += 3;
                }
        });
// 省略
```

main.js
～クマさんをコントロールする～

- 「もし、キーボードの○○が押されたら、○○方向へ移動する。」

- 「もし、～なら、～する。」は if文

- if (core.input.left) {
  this.x -= 3;
  }

- 座標系に気をつけて！
  （X軸は右向きに正、Y軸は下向きに正）

課題

1. 画面（core）の枠を超えて移動することができないようにしよう。

2. 画面（core）の枠を超えたら反対側に行くようにしてみよう。
（パックマンみたいな感じ）

今日のテーマ(3)

ラベルを表示してみる