

テーマ1

enchant.js で ゲーム作り

(3)

2015/05/12

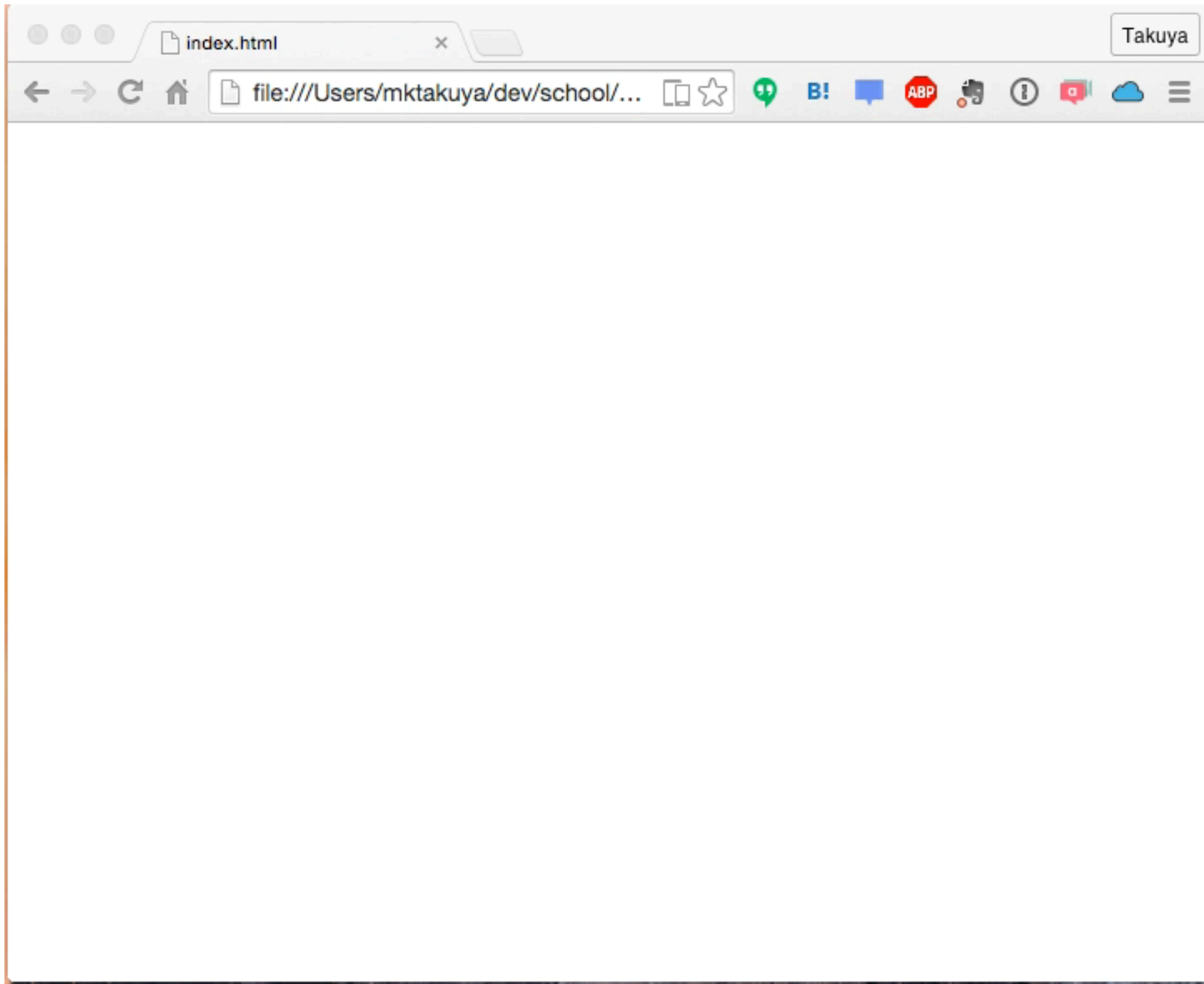
苫小牧高専 ソフトウェアテクノロジー部

今日やること

- ・ 前回のまとめ
- ・ インデントの話
- ・ ラベルを表示しよう

前回のまとめ

クマさん端っここで消える問題



クマさん端っここで消える問題

- ・ 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？

if文

```
if (条件) {  
    処理;  
}
```

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();
}
```

if文

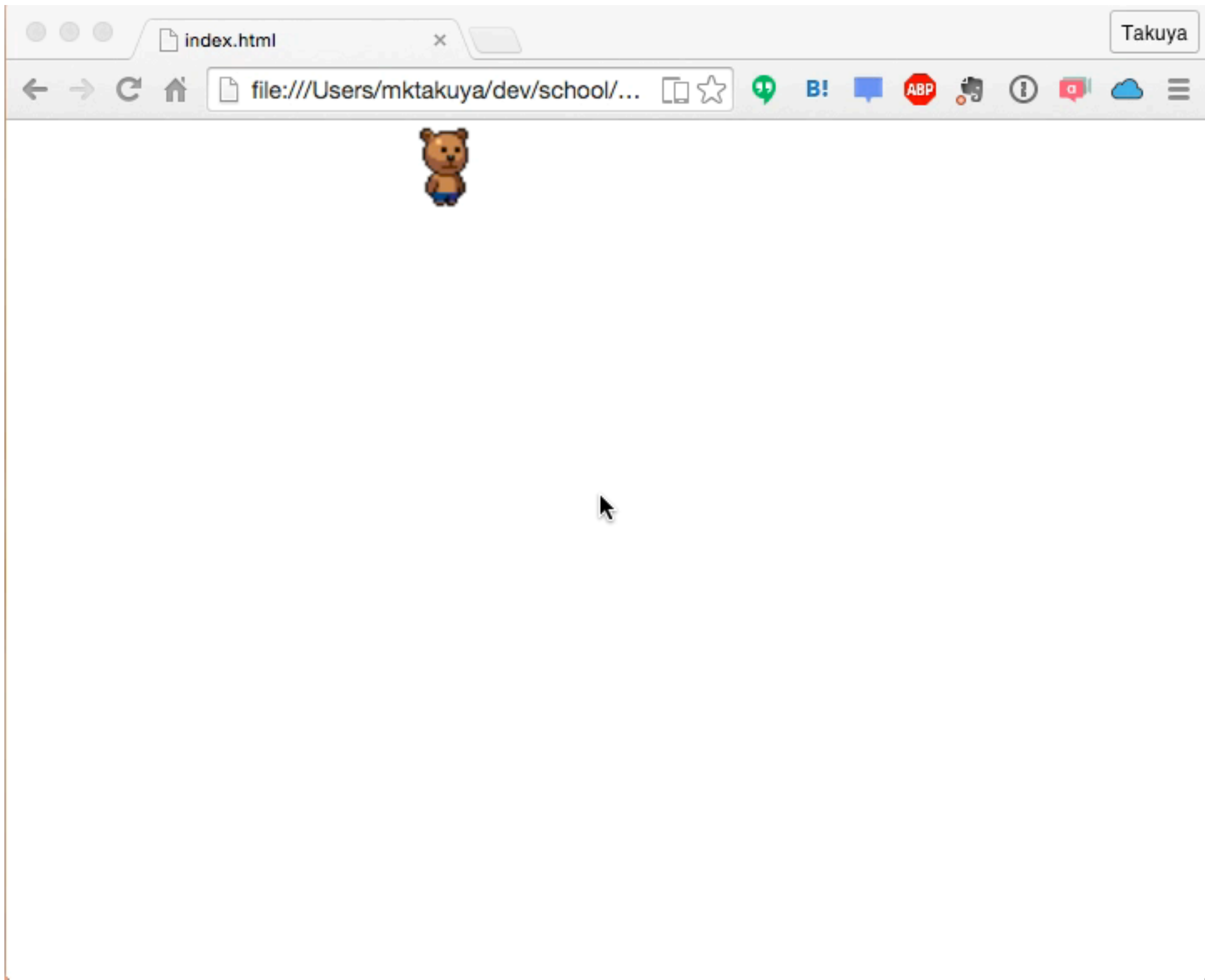
↓ this.x が 320を超えた時

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

↑ this.x に 0 を設定する。

クマさん端っここで消える問題

☆解☆決☆



前回のテーマ(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;

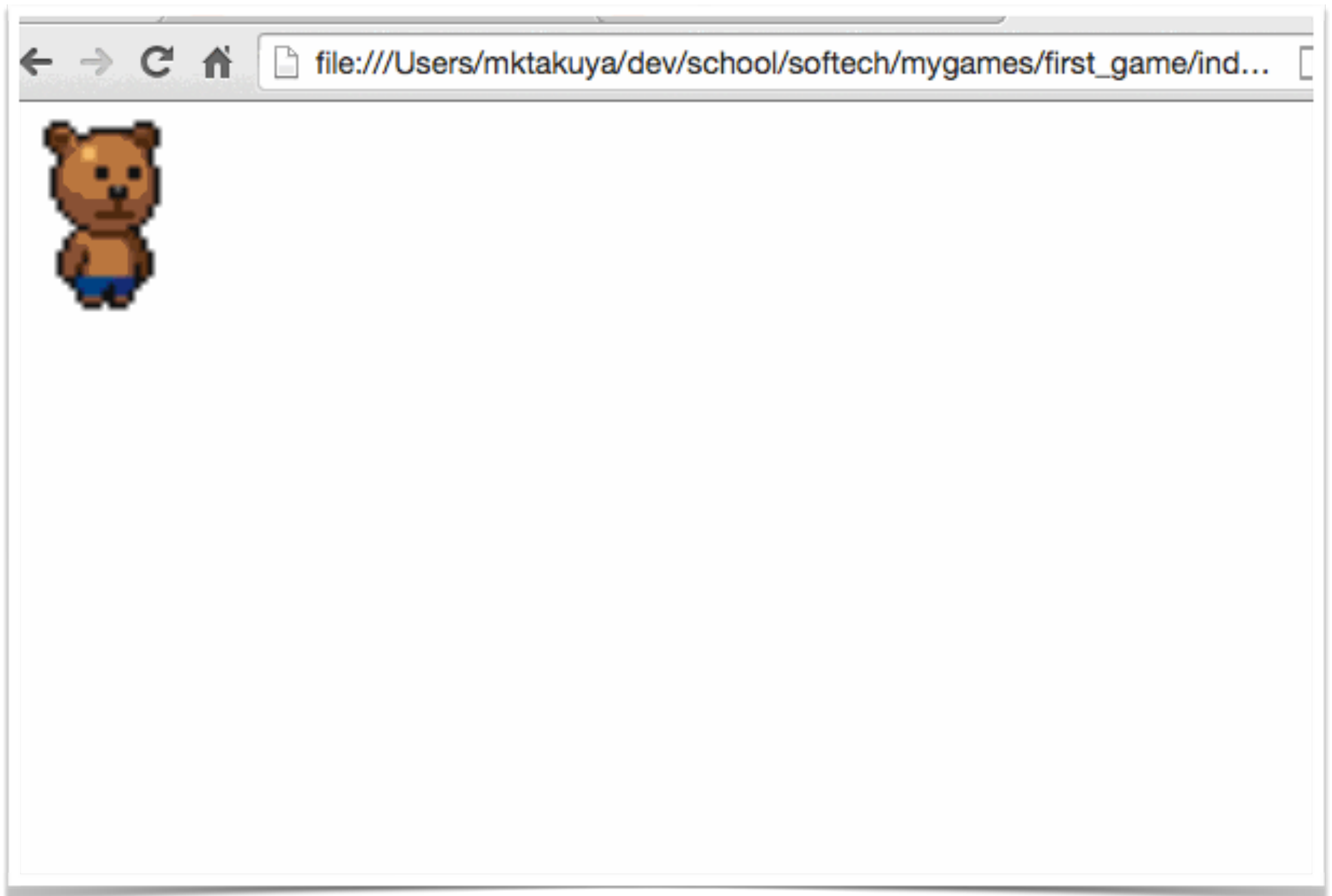
    core.rootScene.addChild(bear);

    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！

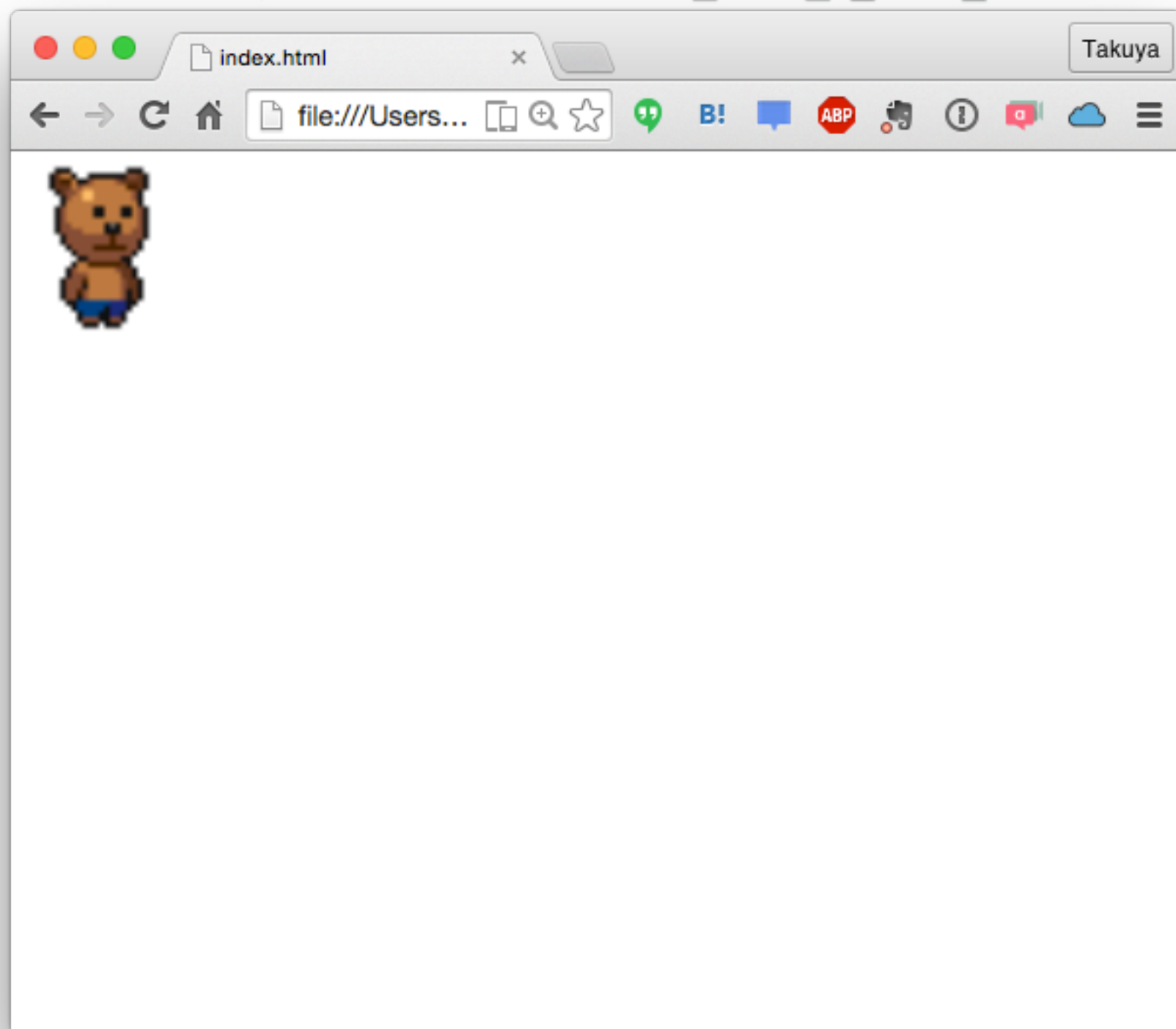


画像のフレーム について

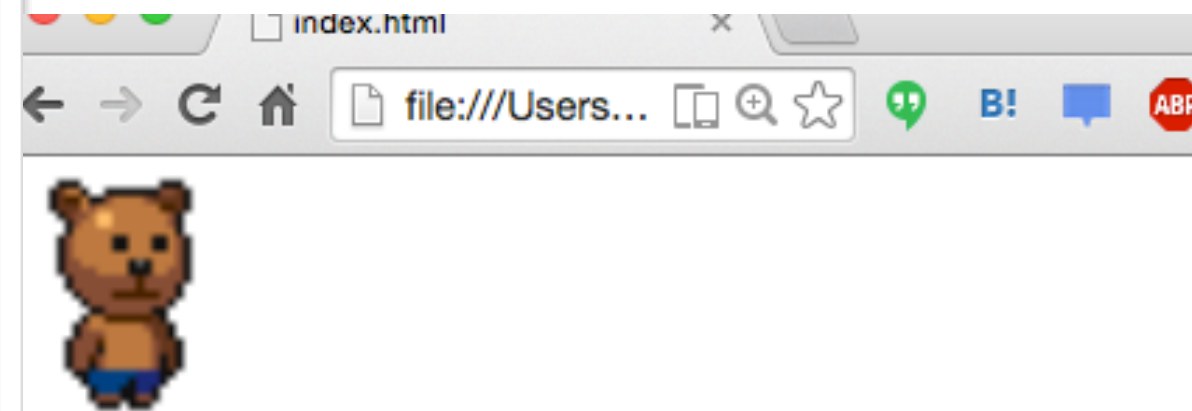
chara1.png



ゲーム画面

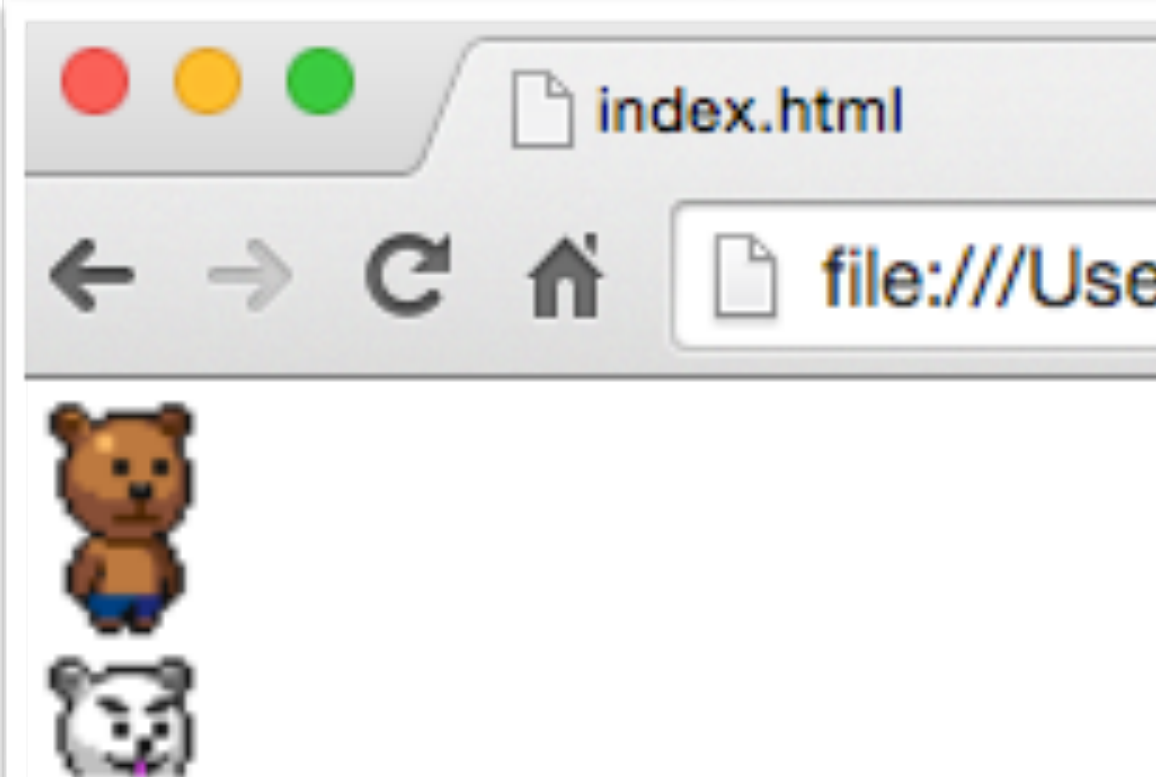


chara1.pngの 一部分（フレーム）を切り抜いて表示している



```
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
```

前回のテーマ(2)

「キャラクターを操作してみる」

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、キーボードの〇〇が押されたら、〇〇方向へ移動する。」
- ・ 「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)

インデント

どっちが見やすい？

インデントあり

```
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);
    }
    core.start();
}
```

インデントなし

```
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);
    }
    core.start();
}
```

インデントとは

- ・ インデント = 字下げ
- ・ インデントを使って、
入れ子になっている部分をわかりやすくするべし。
- ・ インデントは、キーボードの[tab]キーで！
(スペースキー連打はダメよ)
- ・ 自分と、他人（3日後の自分も他人！）が
見やすいコードを書くよう心がけよう！

エディタの設定を いじってみよう

- ・ プログラマーを目指すなら、
道具を自分仕様に改造しよう！
- ・ プログラマにとっての道具 → エディタ！
- ・ TeraPadのオートインデントを有効にしよう！
- ・ 自分のPCがある人は、
ぜひTeraPad以外のエディタを使ってみよう！
(最近だとSublimeTextがオススメ？)

本題

今日のテーマ(1)

ラベルを表示してみる

～ストップウォッチを表示してみよう～

main.js

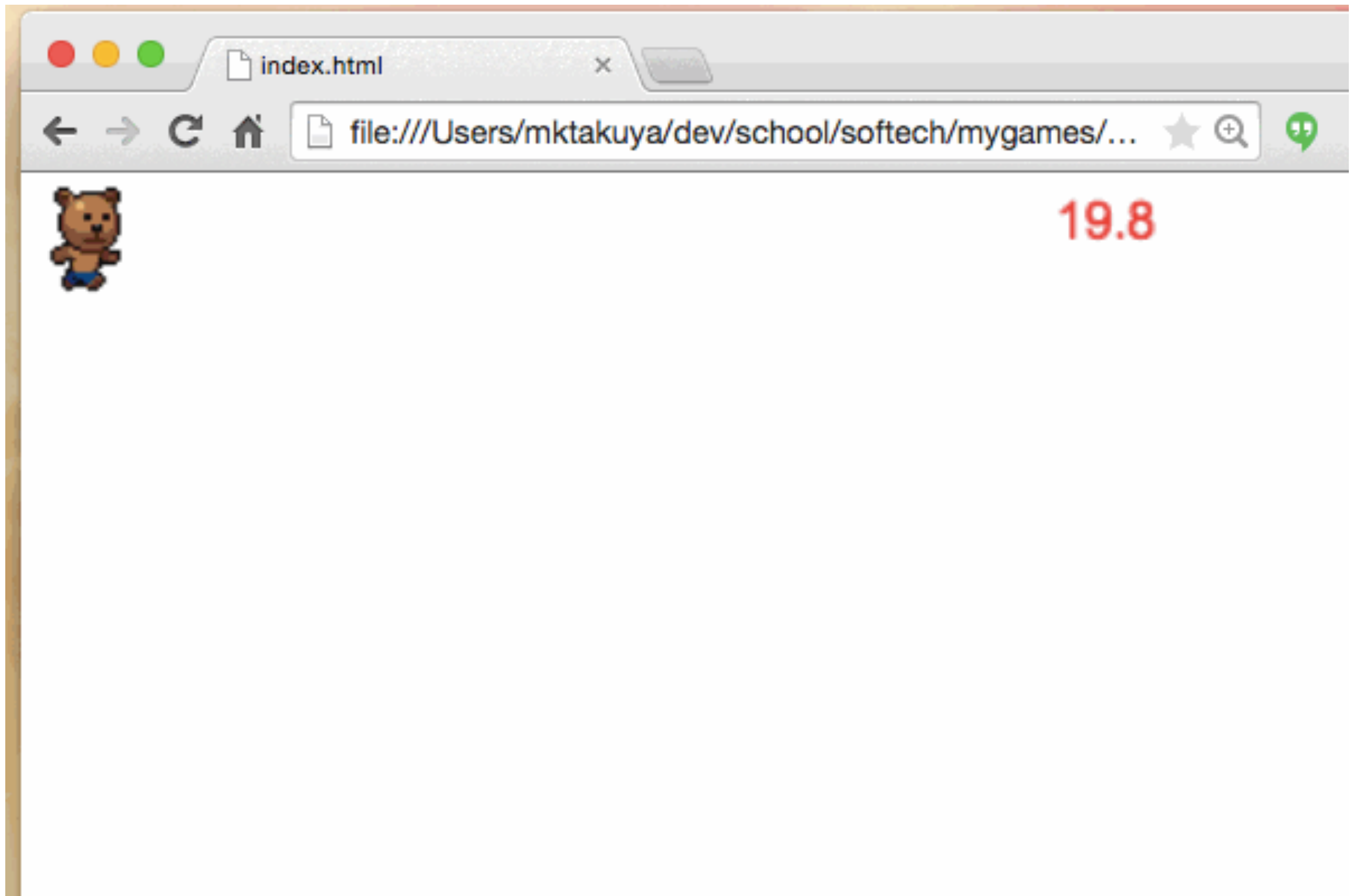
```
// 省略
if (core.input.down) {
    this.y += 3;
}
});

var label = new Label();
label.x = 280;
label.y = 5;
label.color = 'red';
label.font = '14px "Arial"';
label.text = '0';

label.on('enterframe', function() {
    label.text = core.frame / core.fps;
});

core.rootScene.addChild(bear);
core.rootScene.addChild(label);
}
core.start();
}
```

ストップウォッチ的な



main.js

```
// 省略
if (core.input.down) {
    this.y += 3;
}
});
```

```
var label = new Label(); ← labelを作成
label.x = 280;           ← 座標指定（おなじみ）
label.y = 5;
label.color = 'red'; ← 色指定
label.font = '14px "Arial"'; ← フォントとサイズ
label.text = '0';

label.on('enterframe', function() { ← たぶんaddEventListenerと同じ
    label.text = core.frame / core.fps;
});
```

```
core.rootScene.addChild(bear);
core.rootScene.addChild(label); ← addChildを忘れずに！
}
core.start();
}
```

label.on

```
label.on('enterframe', function() {  
    label.text = core.frame / core.fps;  
});
```

- ・ たぶんaddEventListenerと同じ
- ・ label.text は labelに表示されるテキスト。
そこに core.frame / core.fps を格納している。
- ・ 現在のフレーム数をfpsで割れば秒数が出るよ！！

課題1

1. カウントアップではなく、
カウントダウンしてみよう！
2. ラベルを整数だけの表示にしよう！

【ヒント】

`5 / 2;`

`#=> 2.5` と表示される

`Math.floor(5 / 2);`

`#=> 2` と表示される

今日のテーマ(2)

衝突判定をやってみる

～いよいよゲームっぽくなってきたかも～

衝突判定とは？

- ・ キャラクターとキャラクターが衝突した事（イベント）を判定する処理。
- ・ ゲームプログラミングには必須の処理！



まずは敵を作ろう

```
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;
```

```
    var enemy = new Sprite(32, 32);  
    enemy.image = core.assets['chara1.png'];  
    enemy.x = 280;  
    enemy.y = 50;  
    enemy.frame = 5;
```

← 敵は白いクマさん（フレーム番号5）にする

```
    bear.addEventListener('enterframe', function() {  
      // 以下略
```

addChildするのも忘れずに！

衝突判定の処理を書こう

```
// 省略  
if (core.input.down) {  
    this.y += 3;  
}
```

```
if (this.within(enemy, 15)) {  
    label.text = 'HIT!';  
}
```

```
});
```

↑ 衝突したらlabel.text を“HIT!”にする

```
var label = new Label();  
label.x = 280;  
label.y = 5;  
label.color = 'red';  
label.font = '14px "Arial"';  
label.text = '0';  
  
label.on('enterframe', function() {  
    label.text = core.frame / core.fps;  
});  
// 以下略
```

within

```
if (this.within(enemy, 15)) {  
    label.text = 'HIT!';  
}
```

- thisはbear
- this.within(enemy, 15)
thisの中心とenemyの中心が15px近づいたらHIT!
- 数値を変えれば衝突判定を厳しくしたり、ゆるくしたりできる。

課題2

1. 敵にアニメーションをつけよう！

2. 敵も動かしてみよう！

【適当に動かす？】

frameが進むごとにxやyを変えてみる？

【鬼ごっこゲーム？】

矢印キー以外を割り当てて、

2人で対戦できるようにする？

