

～ iPad専用電子書籍をHTML5+CSS3+JavaScript で構築 ～

『ナストマ・マガジン』 制作における トライ&エラーの共有

CSSniTE
LP17
magazines made
w/HTML5
2011.9.17

CGFM LLC. 金内透



Chapter 1

はじめに



自己紹介



@Garyuten



@Garyuten

金内 透

(かねうち とおる)

- 1975年生 36才 福岡市在住
- Blog <http://blog.cgfm.jp/garyu/>
- 合同会社CGFM 代表社員
- 職業：マークアップエンジニア
- 相方デザイナー(カネウチカズコ) の2人
SOHOスタイルで工作中

カネウチカズコ



ナストマくん



うっかりしたっていいじゃない♪

ナストマくん

4コママンガ



4コママンガ



ナストマ神社



ナストマ・ラボ



最新4コマ

2011/08/15

ナストマくん 066

作/サイト☆ムツキ

最終電車



人気4コマランキング



第44話
ICカード
共感度★★★★★(6)



第6話
探し物
共感度★★★★★(2)



第48話
ダテめがね
共感度★★★★★(2)

バックナンバー



第66話
最終電車
共感度★★★★★(0)



第65話
おつかい
共感度★★★★★(4)



第64話
冬の電気代
共感度★★★★★(2)



第63話
トマトの野望
共感度★★★★★(2)



第62話
予約の時間
共感度★★★★★(0)



第61話
歯医者帰り
共感度★★★★★(0)



第60話
レジ待ち
共感度★★★★★(5)



第59話
自動販売機
共感度★★★★★(3)



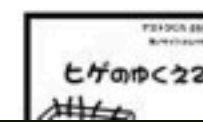
第58話
サプライズ
共感度★★★★★(3)



第57話
潮干狩り
共感度★★★★★(0)



第56話
電気毛布
共感度★★★★★(1)



第55話
ヒゲのゆくえ(その2)
共感度★★★★★(2)

自宅兼事務所のSOHO

相方にストレスが溜まる



世界中に「うっかり」を配信

精神的DV

(Domestic Violence)



気がつけば66話



ナストマくんマガジン
2011年GW合宿で制作



4月中旬トマト農家取材

GW合宿制作

5/9 公開



アプリ公開が
間にあうわけがない



そんなわけで
今回は WebClip



今回利用した技術

HTML5

CSS3

アプリケーション
キャッシュ

JS
(jQuery)

iOS

WebClip

Chapter 2

実機実演

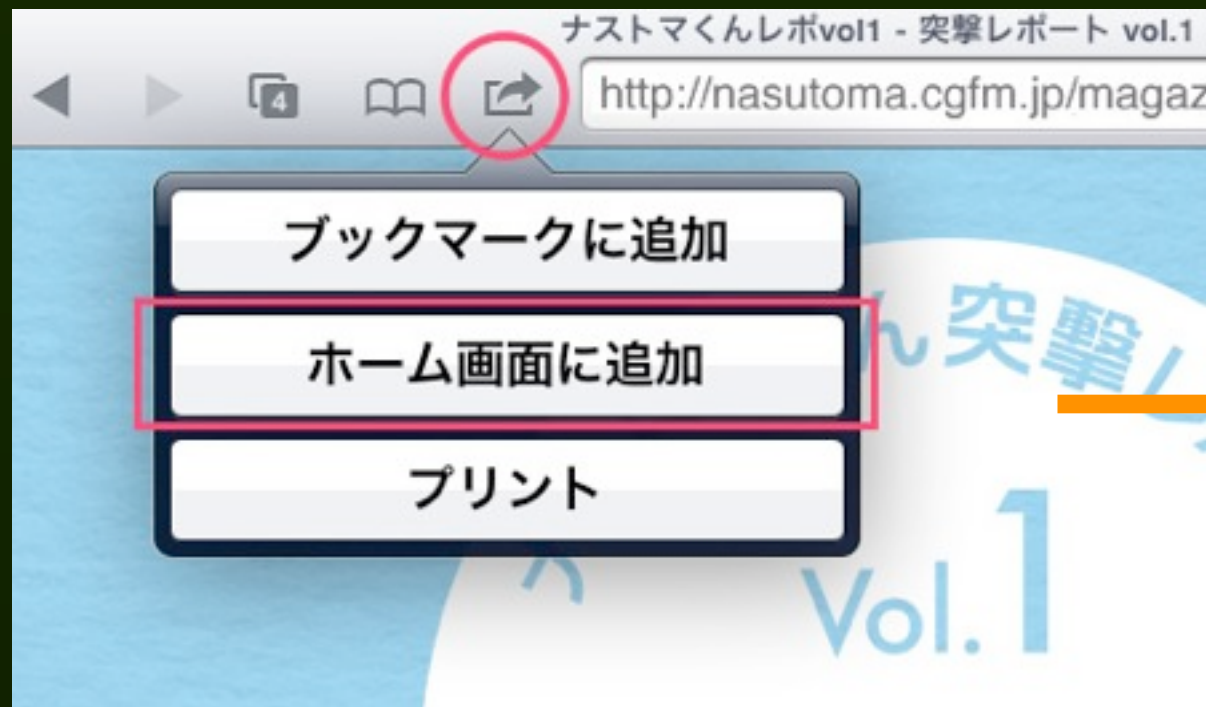


Chapter 3

WebClip と アプリケーション キャッチシュ



WebClip



HTML5

アプリケーションキャッシュ

- HTML5からの新機能
- “CACHE MANIFEST”でブラウザからローカルストレージにデータを保存
- 保存できる最大容量は未確認
(ナストマくんマガジンでは10MB前後)
- 超強力すぎるキャッシュ(後述)

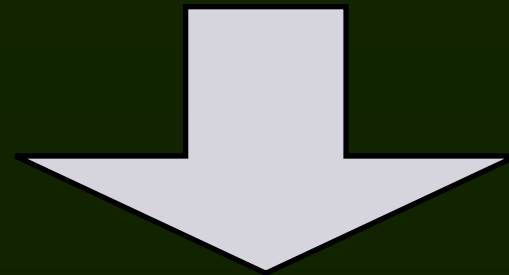
iOS

WebClip

+

HTML5

アプリケーション
ンキャッシュ

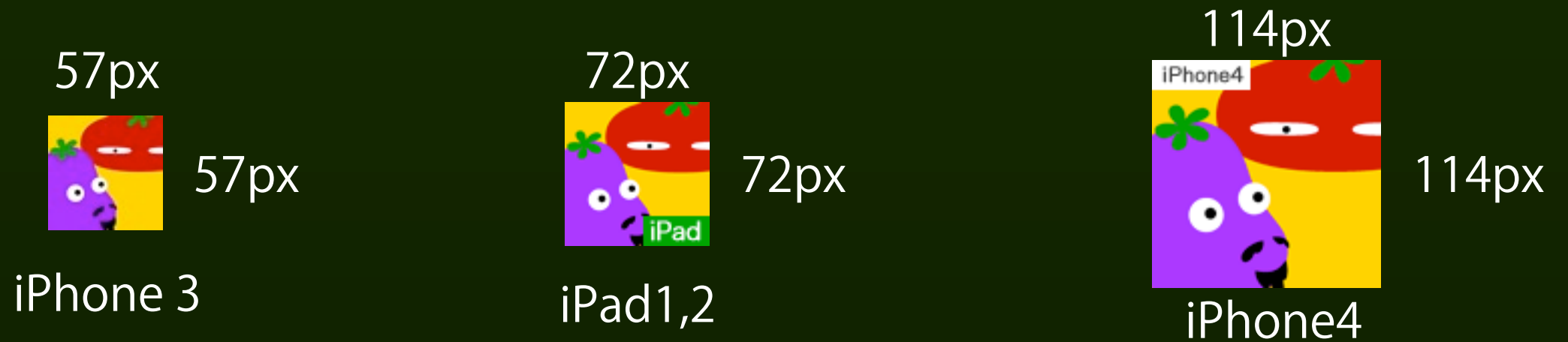


オフラインでも動作する
Webアプリケーション

WebClipで 準備すること



WebClipアイコンのサイズ



WebClipアイコンのアプリ名



ページタイトル=アプリ名
全角6文字(半角12文字)
それ以上は自動省略

- faviconとほとんど同じ
- PNG画像
- デフォルトのファイル名は「apple-touch-icon.png」
(任意変更可能)
- サイトのルートディレクトリに設置

WebClipアイコン

デフォルト



-precomposed



ファイル名を「apple-touch-icon-precomposed.png」で
自動ハイライト処理がOFFにできる
(自動角丸はOFFにできない)

WebClipアイコン - HTMLの記述

```
<!-- ▼iPhone3 -->  
<link rel="apple-touch-icon" href="iphone.png" />  
<!-- ▼iPad -->  
<link rel="apple-touch-icon" sizes="72x72" href="ipad.png" />  
<!-- ▼iPhone4 -->  
<link rel="apple-touch-icon" sizes="114x114" href="iphone4.png" />
```

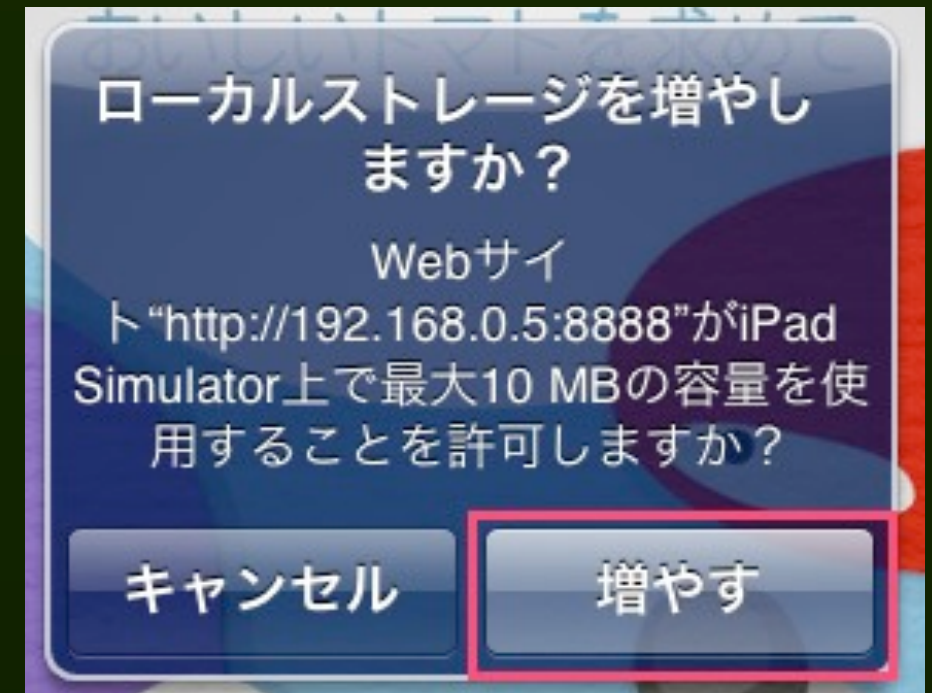
- 任意のファイル名でOK
- size属性で iPhone3、iPad、iPhone4 毎に切替可能

manifest
(.appcache)

manifestファイルでキャッシュ

HTML5でのアプリケーション
キャッシュさせるには、
マニフェストファイルの準備が必要
どのファイルをキャッシュするかを
列挙することで、Webページ表示時
に自動でキャッシュ保存してくれる

10MB以上になると警告が出る



manifestファイルの指定

HTMLタグに manifest属性で記述

```
<!DOCTYPE html>  
<html lang="ja" manifest="vol1.appcache">
```

拡張子(推奨) .manifest → .appcache

manifestファイルの中身

vol1.appache

```
CACHE MANIFEST
# Version: 0.26
# vol.1 2011/05/09 by CGFM LLC.
```

```
CACHE:
vol1.html
css/ipad-landscape.css
css/ipad-portrait.css
css/ipad.css
js/ipad.js
js/jquery-1.4.4.min.js
js/jquery.touchSwipe-1.2.4.js
js/jquery.ui.js
js/jquery.ui.transitionAnimator.js
```

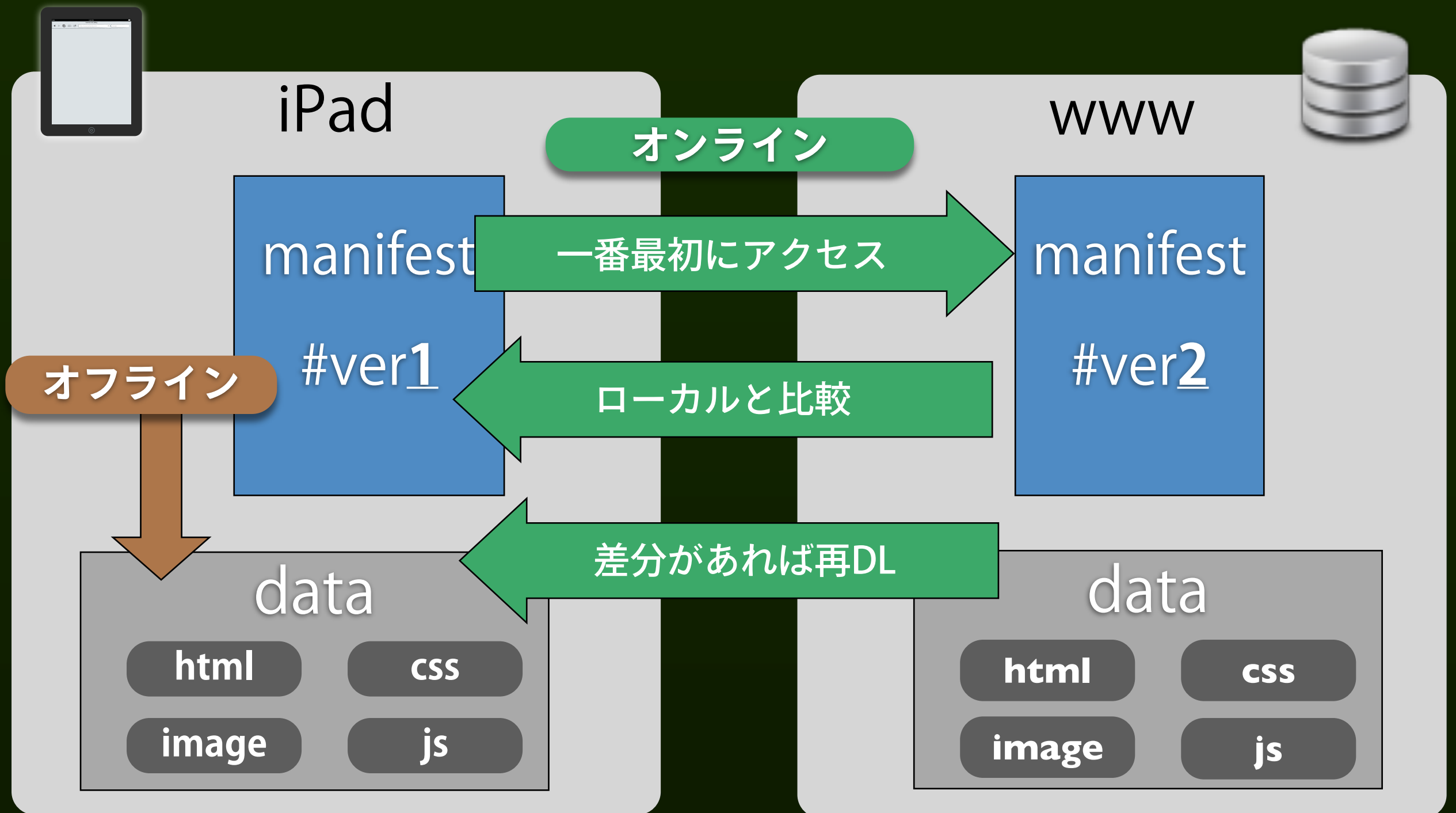
```
img/00.jpg
img/01.jpg
img/02.jpg
img/03.jpg
img/04.jpg
img/05.jpg
img/06.jpg
img/07.jpg
img/08.jpg
img/09.jpg
...
```

CACHE: キャッシュさせるファイル

NETWORK: ネットワーク経由でアクセスさせるファイル



manifestファイルの仕組み



他 特有のmeta情報

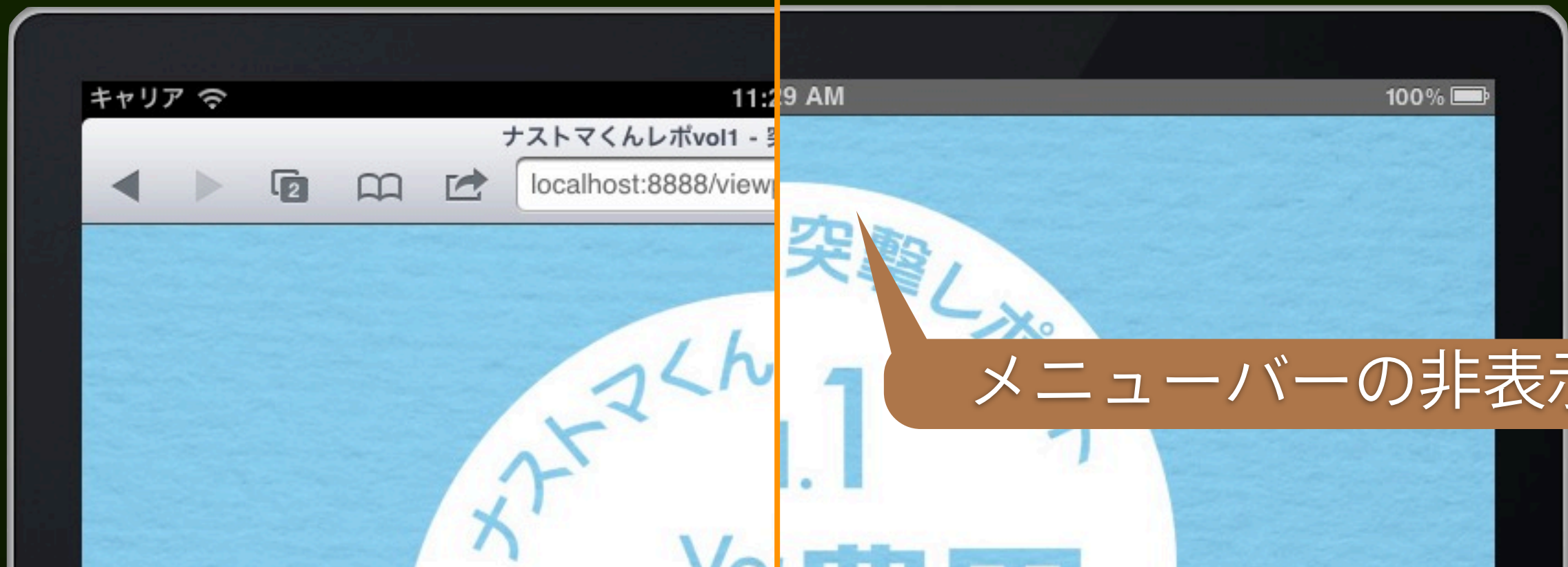
apple-mobile-web-app-capable

フルスクリーンモードを指定

```
<!-- WebClip起動時にフルスクリーン表示の指定 -->  
<meta name="apple-mobile-web-app-capable" content="yes">
```

no

yes



メニューバーの非表示

apple-mobile-web-status-bar-style

ステータスバーのスタイルを制御

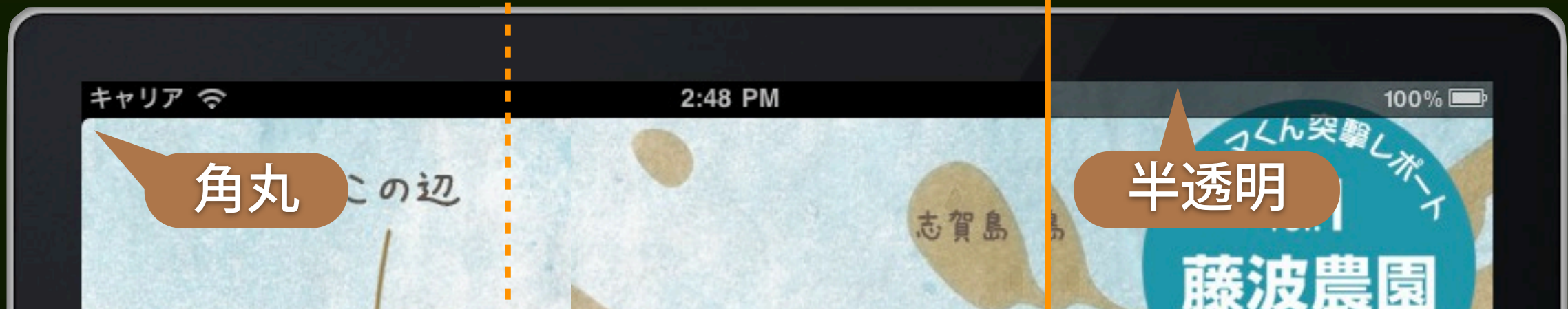
```
<meta name="apple-mobile-web-app-status-bar-style"  
      content="black-translucent">
```

- default : 不透明（デフォルト色）で表示
- black : 不透明（黒）で表示
- black-translucent : 半透明で表示

default

black

black-translucent



viewport

論理ウィンドウサイズを指定する他、
拡大縮小の許可、最大拡大/縮小率を指定

```
<meta name="viewport"  
  content="width=device-width,  
          initial-scale=1,  
          user-scalable=no,  
          maximum-scale=1.0">
```

詳細は「トライ&エラー」の章で

HTMLの記述 まとめ

```
<!DOCTYPE html>
<html lang="ja" manifest="vol1.appcache">
<head>

<meta name="viewport" content="width=device-width, initial-
scale=1,user-scalable=no,maximum-scale=1.0">

<meta name="apple-mobile-web-app-capable" content="yes">
<meta name="apple-mobile-web-app-status-bar-style"
content="black-translucent">

<link rel="apple-touch-icon" href="iphone.png" />
<link rel="apple-touch-icon" sizes="72x72"
href="ipad.png" />
<link rel="apple-touch-icon" sizes="114x114"
href="iphone4.png" />
...
```

Chapter 4

WebClipの メリット/デメリット



WebClipのメリット



Webの制作知識そのまま



Safariがバックグラウンドで起動
JavaScriptもリンクも問題なく動作
メニューバーを非表示に

ほとんど アプリに見える





**App Storeに
登録申請不要**

Web感覚で配信



オフラインで利用可能

(※外部API未使用に限る)

WebClipのデメリット





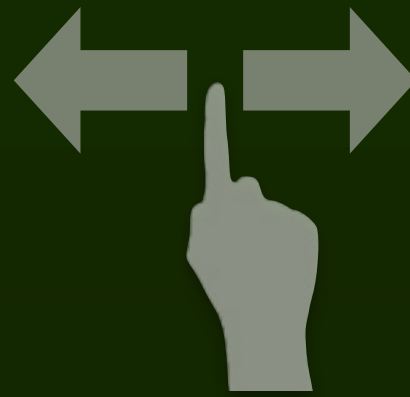
カメラ、マイク
が使えない



写真、音楽ファイルに
アクセスできない



ファイルの
アップロードができない



Safariの限界

(Swipeイベントが取れない等)





アプリの表示状態を 維持できない

(解決策はある)

他のアプリから切り替えると
必ず初期化(再読込)される



アプリの 再起動できない

更新ボタンが無い



要課金システム

シンプルな電子書籍なら
十分実用可能



Chapter 5

トライ&エラー



HTML



全て1つのHTML

HTML

#content

section



#slide

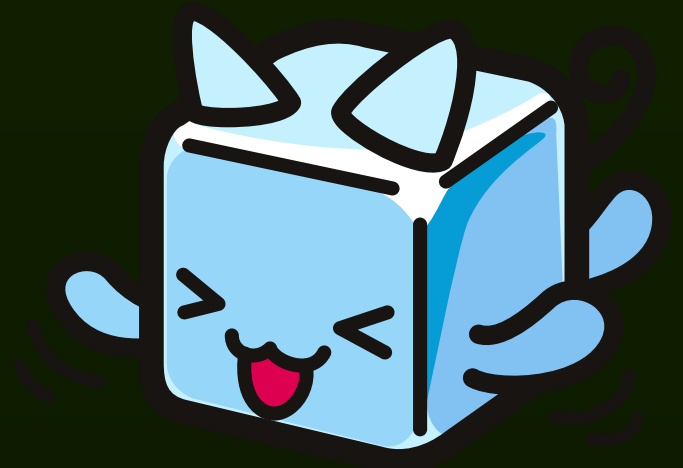
section

ul

#footer

コーディングツール

- HTMLコーディング
→ Dreamweaver + Coda
- JavaScript → NetBeans
- CSS → CSSEdit

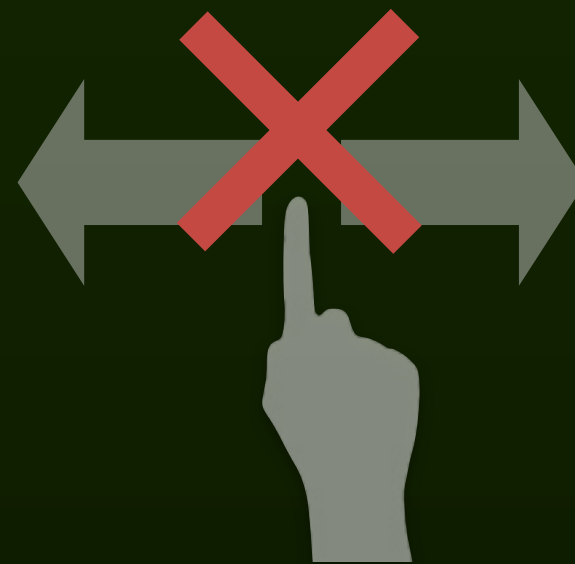


SafariのJavaScript



JSで擬似Swipeを実装

- SafariではSwipe(スワイプ)のイベントを取れない(そもそもない)



touchSwipe 便利

- touchSwipe - jQuery Plugin
<http://labs.skinkers.com/touchSwipe/>
- スクロール方向の制限も指定できる



touchSwipe Sample

```
var contentSwipeOptions= {  
  swipe: contentSwipe, //スワイプイベント時のcallback関数名  
  threshold:75, //スワイプの感知時間 ( ms )  
  allowPageScroll: "vertical" //縦方向のみスクロール  
}  
  
$('#content').swipe( contentSwipeOptions ); //スワイプ対象要素にセット  
  
function contentSwipe (e , direction , distance) {  
  switch (direction) {  
    case 'left': PageChange('R'); break; // ←方向へのスワイプ  
    case 'right': PageChange('L'); break; // →方向へのスワイプ  
    case 'up': PageChange('B'); break; // ↑方向へのスワイプ  
    case 'down': PageChange('T'); break; // ↓方向へのスワイプ  
    default: break;  
  }  
}  
  
function PageChange(n) {} //ページ遷移処理
```

onclickが使えない

- ブラウザでの確認作業が困難
- touchstart,touchendなどのイベントと合わせて指定しておく と確認作業が楽
- iOSシミュレーターでは
マウスのドラックで指のスワイプを再現
できるので簡単な動作確認が行える



操作関係のイベント

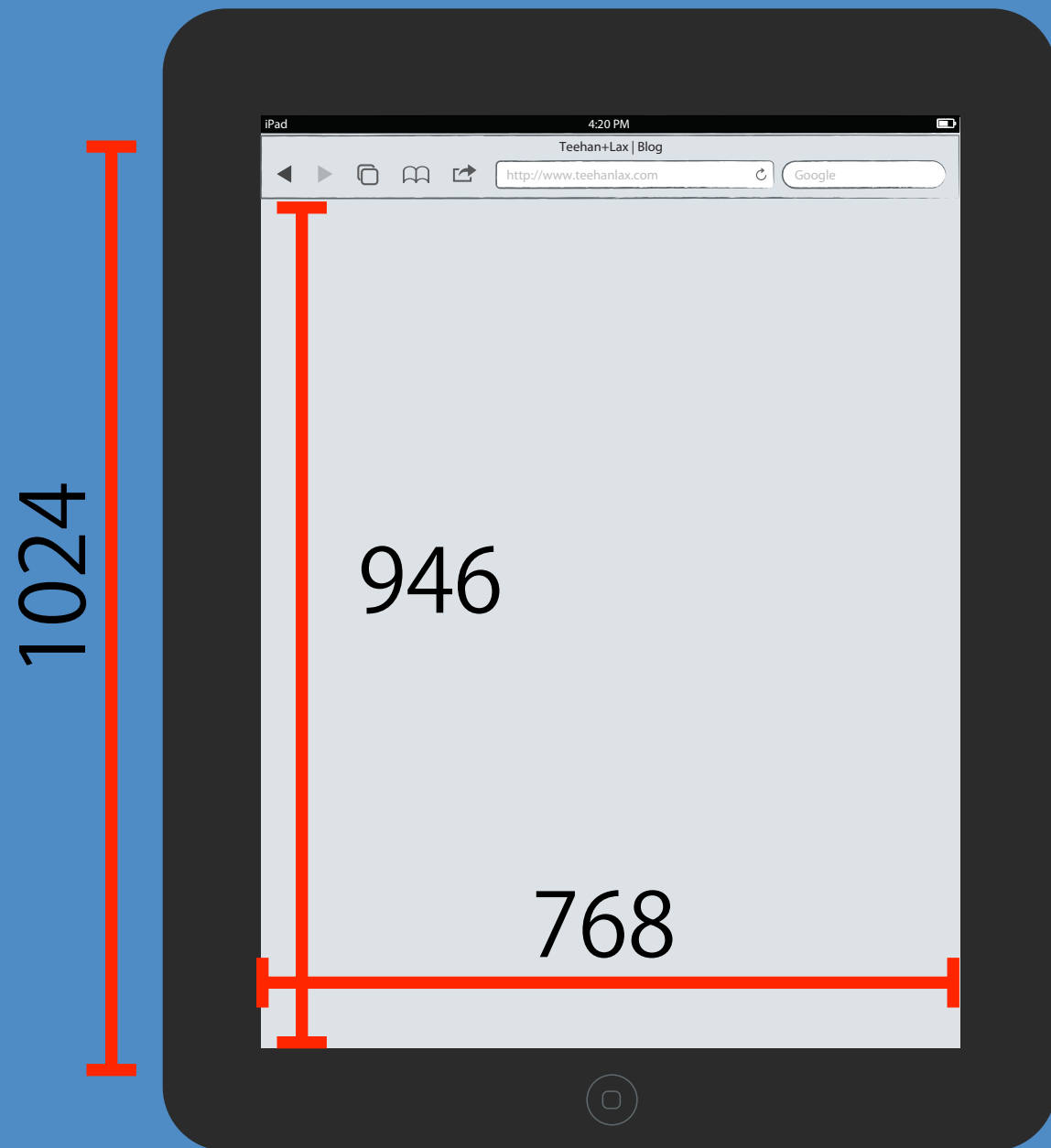
イベント名	説明	近いイベント
event.touches[n]	マウス×指の数	mouse
touchstart	指が触った	onMouseDown
touchmove	指が動いた	onMouseMove
touchend	指が離れた	onMouseUp
touchcancel	操作のキャンセル	

縦置き/横置きの切替

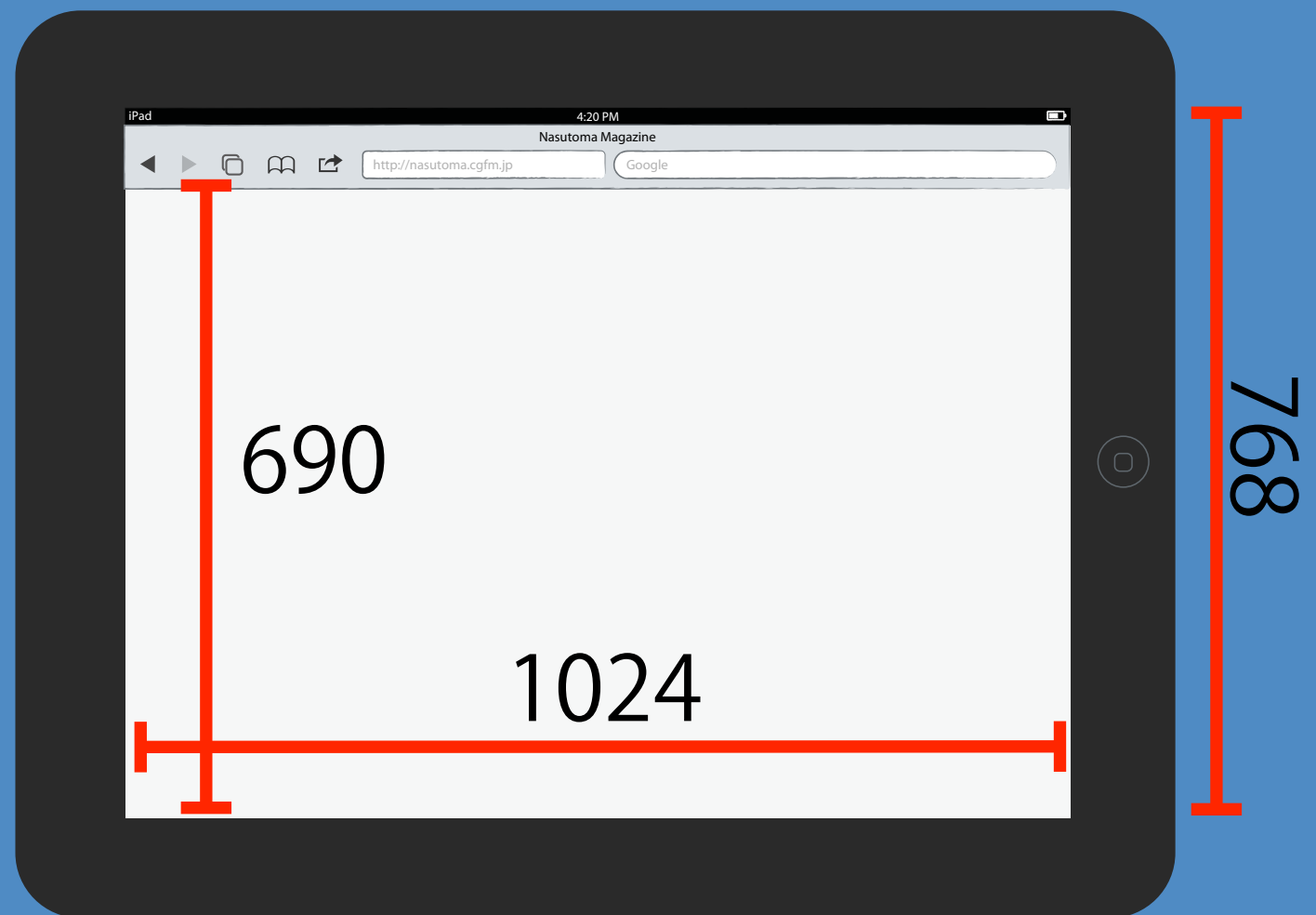


iPad

portrait



landscape



(px)

CSS

media属性で
専用CSSを適用



ipad.css

共通CSS

media="all and (orientation:**portrait**)"

media="all and (orientation:**landscape**)"



ipad-portrait.css



ipad-landscape.css



JavaScriptで縦横回転を検出

```
$(window).bind("resize load",function(){  
  
    if($(window).height()>$(window).width()){  
        //縦>横＝縦向の場合 (orientation:portrait)  
        $('#content').show(); //縦コンテンツを0n  
        $('#slide').hide(); //横コンテンツを0ff  
    }  
    else {  
        //縦<横＝横向きの場合 (orientation:landscape)  
        $('#content').hide(); //縦コンテンツを0ff  
        $('#slide').show(); //横コンテンツを0n  
    }  
})
```

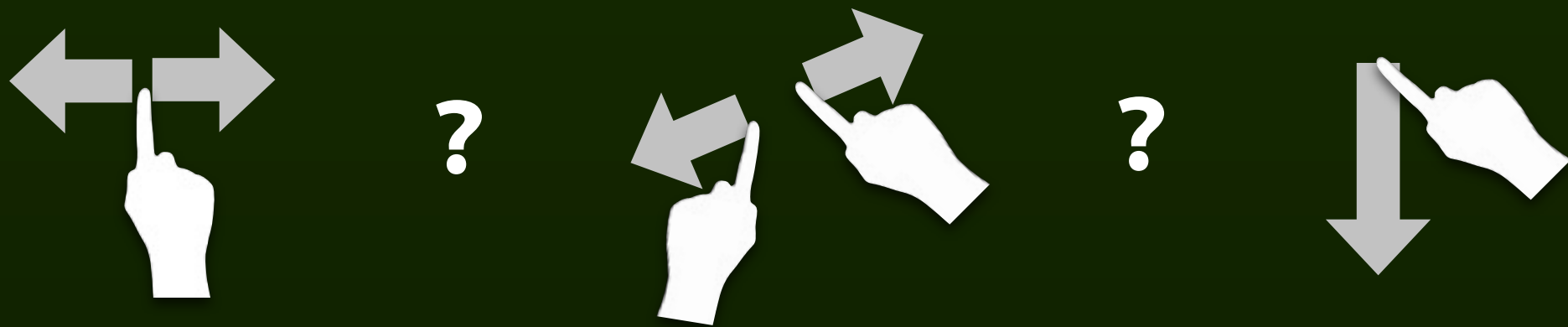
Viewport



今回のViewportの指定

name	value	comment
width	device-width	iPadの場合、 縦向: w768px 横向: w1024px
user-scalable	no	拡大縮小させない
initial-scale	1	初期拡大率=1倍
maximum-scale	1	最大拡大率=1倍

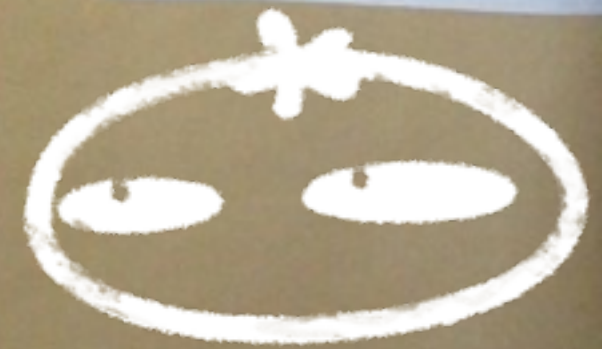
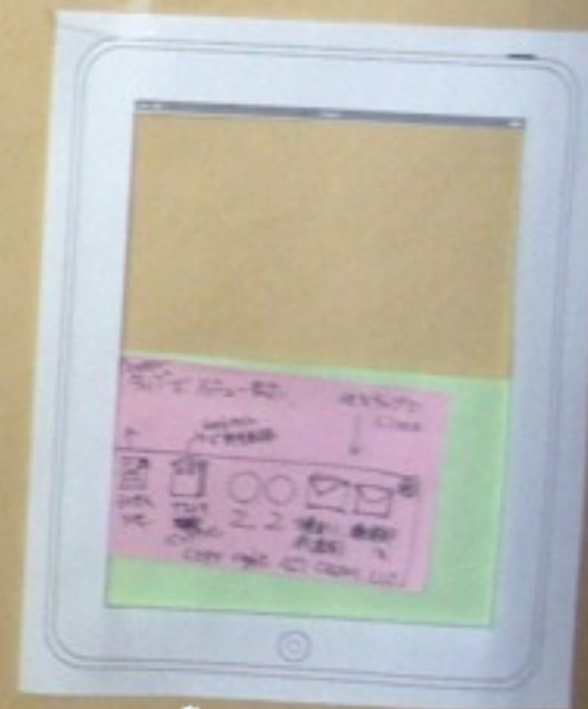
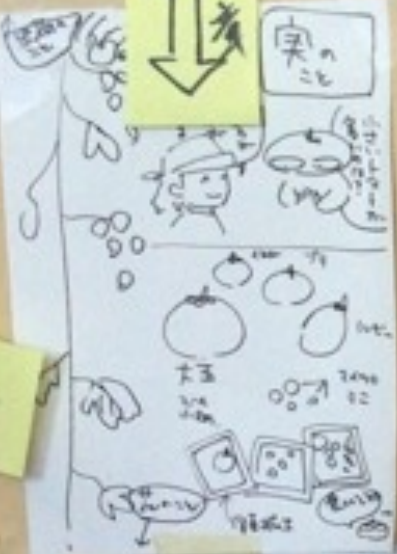
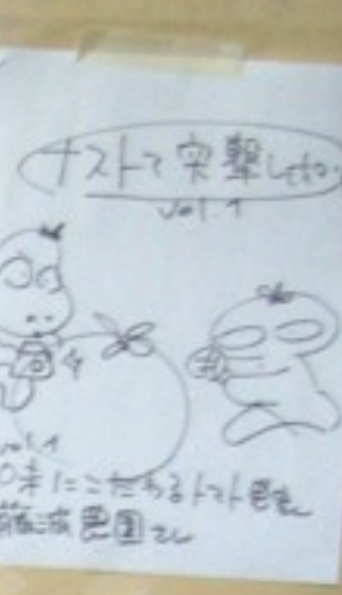
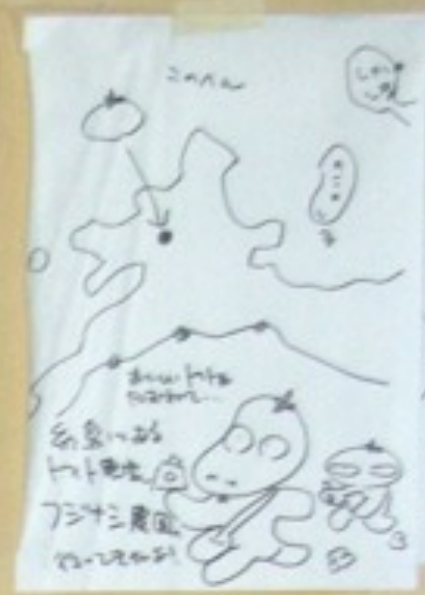
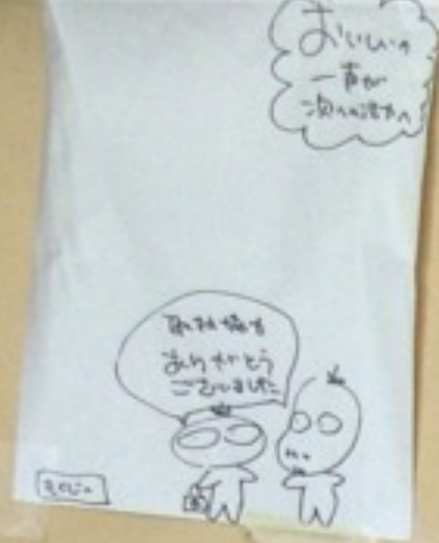
拡大・スクロール許可 + スワイプ時の注意点



- ユーザーの自由度が高まる分、**誤操作を誘発**する可能性が高まる
- 拡大時やスクロールの操作がスワイプイベントとなるので注意が必要
- コンテンツ次第で、拡大・スクロールのどちらかを禁止するのも手



ページ遷移

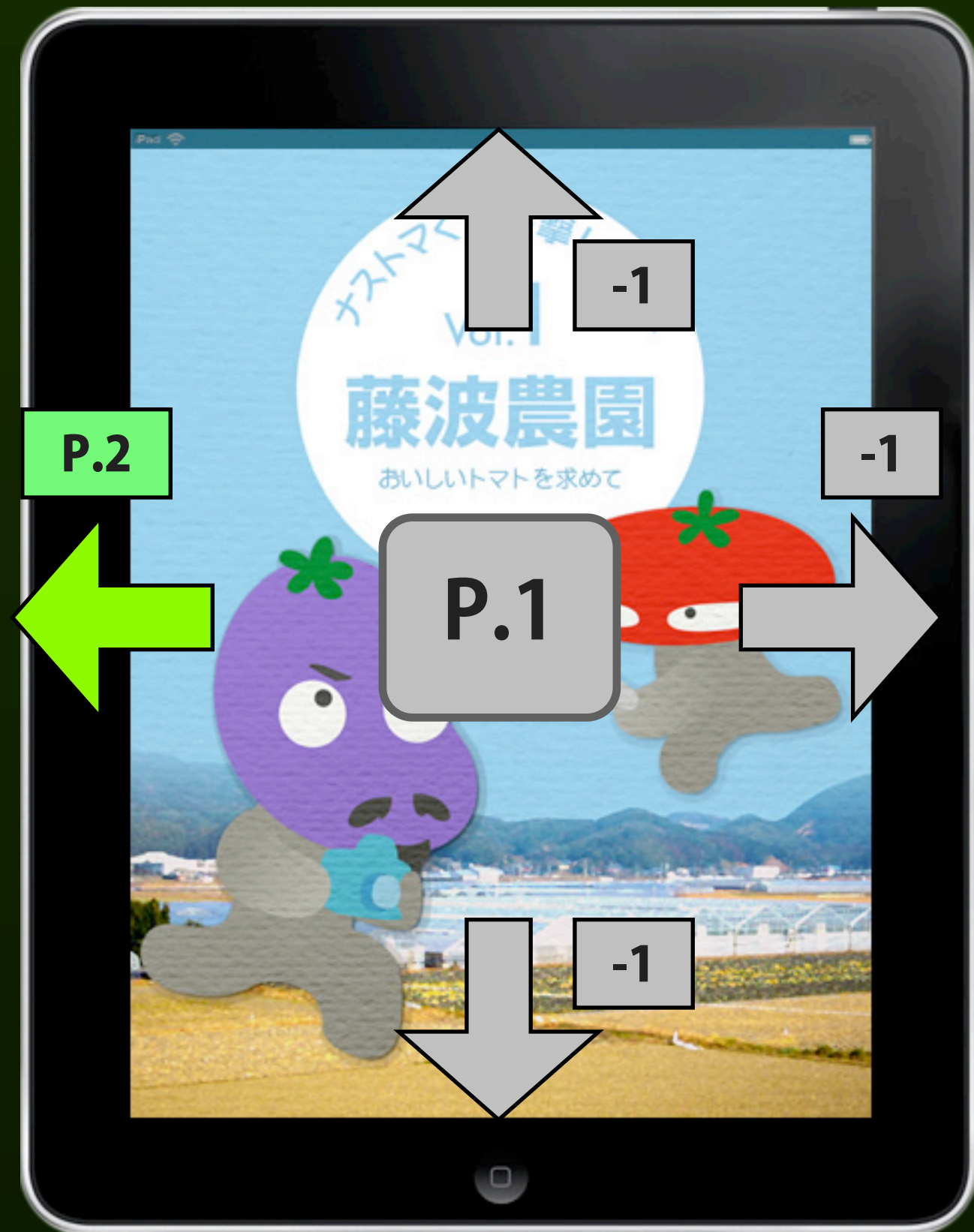


ページ遷移の制御

RPGのマップの考え方

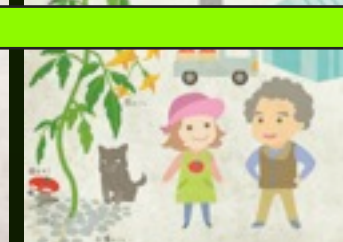
配列変数にページ毎の
上下左右に移動先の
ページNoを指定

移動できない方向には[-1]

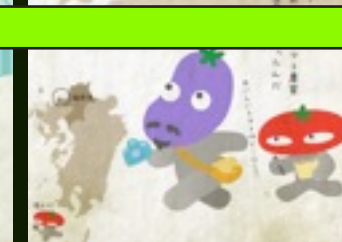
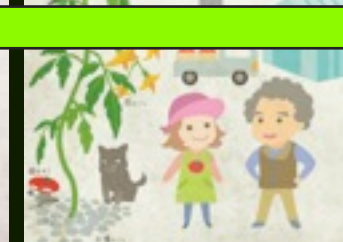
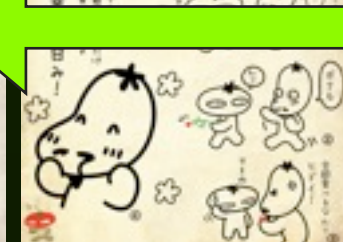
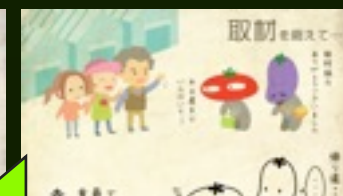




end



Start



表示状態の保持

WebClipはアプリ切替時に
毎回リロード



表示状態の保持ができない

JavaScriptのLocalStorageを利用
ページ遷移毎に表示状態を保持

アプリ切替→WebClip時に
LocalStorageのページを表示



localStorage

```
if( localStorage.NowPage > 0 ) {  
    activePage = localStorage.NowPage;  
    //指定ページの表示処理  
    pageChange(activePage);  
}
```

アニメーションはCSS3で

jQueryのエフェクトや
animateは、iPhone、iPad
では重いので極力避ける

CSS3のTransitionで
実装すると軽い



表示域外の座標計算は重い



写真スライド等で画面外に大量の写真を配置させた場合
その全て座標でレンダリング処理を行おうとして
表示処理に時間がかかる

manifestにハマる



manifestにハマる

- 強力すぎるキャッシュ
- ホームのアプリ削除、本体再起動では絶対に消えてくれない
- リロードも効かない
- [設定]-[Safari]>[キャッシュを消去]を実行
- 更にSafariの再起動も



注意！
他のWebClipしたアプリも
全て削除される



.appcache (推奨拡張子)

- HTTPヘッダの設定
- “Content-Type: text/cache-manifest”を返すように設定
- これを返さないとうまく反映されない
- お手軽なのは、.htaccess で以下の記述

```
AddType text/cache-manifest .appcache
```

任意の拡張子でもOK

.appcacheの記述の確認

- Mac版Safariの「エラーコンソール」
である程度確認できる
- iPadで実機確認する前に確実に潰しておく
(キャッシュで泣かないために)

エラー例) 記述したファイルをキャッシュできなかった場合

```
>> GET http://localhost/hoge.hoge 404 (Not Found)  
>> Application Cache update failed, because http://  
localhost/hoge.hoge could not be fetched.
```

manifest その他

- 推奨拡張子が最近[.appcache]に変わった

あくまで推奨 任意の拡張子にしてもOK
その場合、HTTPヘッダを必ず返すように設定する
“Content-Type: text/cache-manifest”

- 元々Windowsで既に使われていたファイル拡張子だった
<http://HTML5.org/tools/web-apps-tracker?from=5811&to=5812>
- Safariの[構成ファイル一覧]は確認に便利
(CSSの背景画像はカウントしないので注意)
- CotEditorなど一部のテキストエディタも便利
<http://blog.cgfm.jp/garyu/archives/2478>

確認作業



Firebug (Firefox)

Safari

iOS シミュレータ

実機(localhost+WiFi)

実機(Web+WiFi)

公開

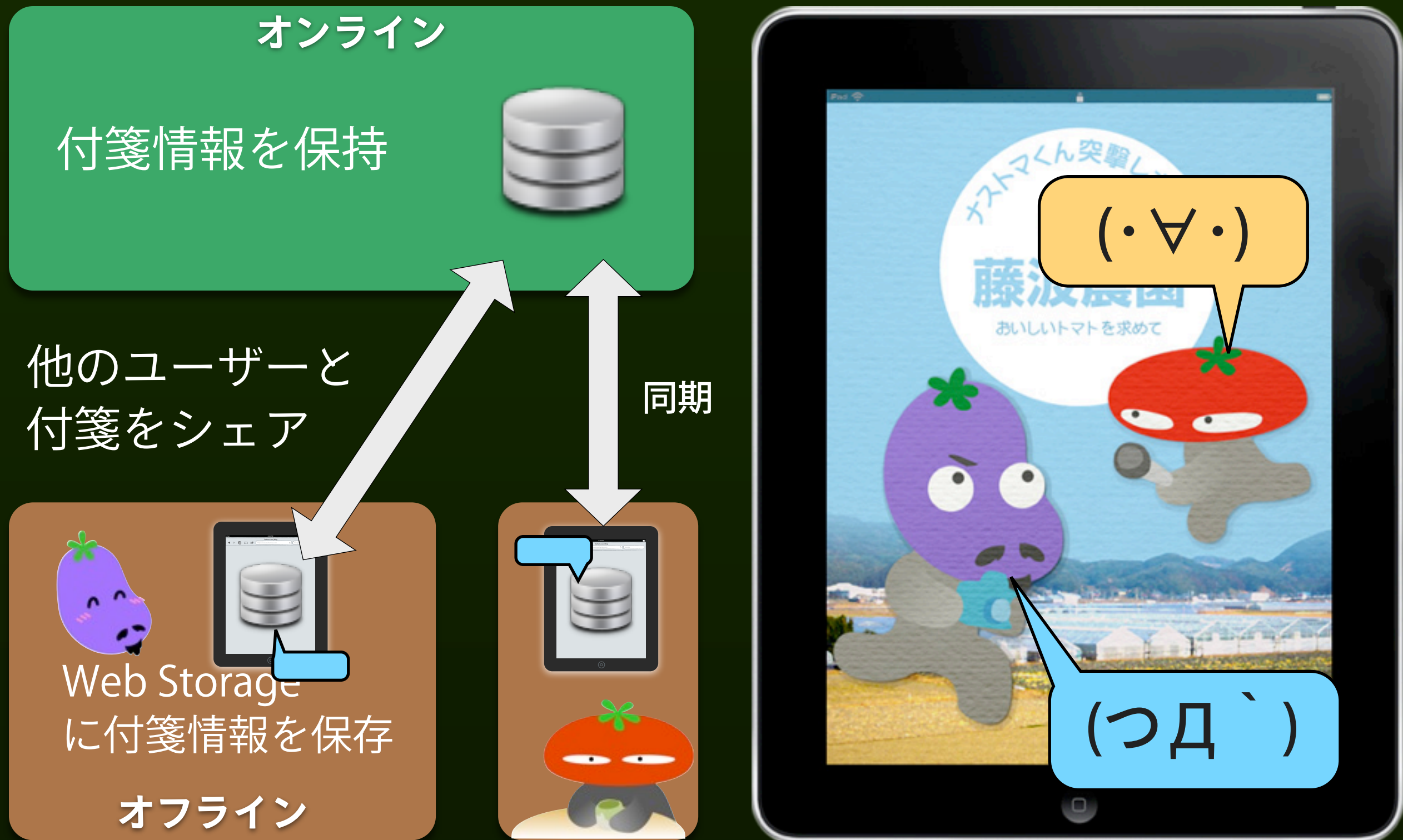
Chapter 6

今後の方向性等



オンライン時の同期処理
で広がる可能性

例) 付箋のシェア



私のトライ&エラーが
みなさんのお役に立てれば
幸いです



ありがとうございました



@Garyuten



参考サイト

- iPadでHTML5でWebアプリを作ってみました。(Yahoo! JAPAN Tech Blog)
<http://techblog.yahoo.co.jp/lab/latlonglab/yubichiz/>
- .manifest→.appcache - vantgarde - web:g
<http://web.g.hatena.ne.jp/vantgarde/20110204/1296754568>
- iPhone/iPadのホーム画面用アイコンapple-touch-icon.pngをサイトに設定しよう | Web担当者Forum
<http://web-tan.forum.impressrd.jp/e/2010/06/15/8178>
- WebのスキルでiPad/iPhoneアプリ風のWebアプリ作成のまとめ | 本を買わずに解決するWeb制作の小技
<http://ameblo.jp/linking/entry-10766619832.html>
- iOS Application Programming Guide: Build-Time Configuration Details
<http://developer.apple.com/library/ios/>