

各画面と対応するファイル一覧

-css

--default.css . . . アプリケーション全体で共通の css ファイルです。

--link-categories.css . . . アプリケーション全体で共通の css ファイルです。

--top-base.css . . . 3つのトップページで共通の css ファイルです。

-images . . . アプリケーションに必要な画像が入っています。

-js

--default.js . . . アプリケーションの起動と共通の動作が定義されています。

--navigator.js . . . ページ遷移を実行するクラスが定義されています。

--zoomable-view.js . . . セマンティックズームに必要なクラスが定義されています。

-pages

--detail . . . トップから遷移する個別のページを格納するディレクトリです。

--top-1 . . . トップページのデザインテンプレート 1 です。

--top-2 . . . トップページのデザインテンプレート 1 です。

--top-3 . . . トップページのデザインテンプレート 1 です。

-default.html . . . アプリケーションのルートページです。すべてのページはこのページ内に展開されます。

-package.appxmanifest . . . アプリケーションをストアに登録するための設定を記述するファイルです。

Windows ストアアプリ固有のスタイル制限について

アプリレイアウト

windows8 アプリとウェブページの大きな違いの一つとしてアプリレイアウトがあります。

次の2つの例をご覧ください。



[フルスクリーン・ランドスケープ]



[スナップ]

windows ストアアプリではユーザーがアプリケーションの画面領域を自由に変更することができるため、異なる画面表示にも一つの HTML で対応できるようにします。アプリレイアウトは全部で4つの状態があります。スタイルを編集するときにはこれらのレイアウト全てで表示が崩れないように注意する必要があります。

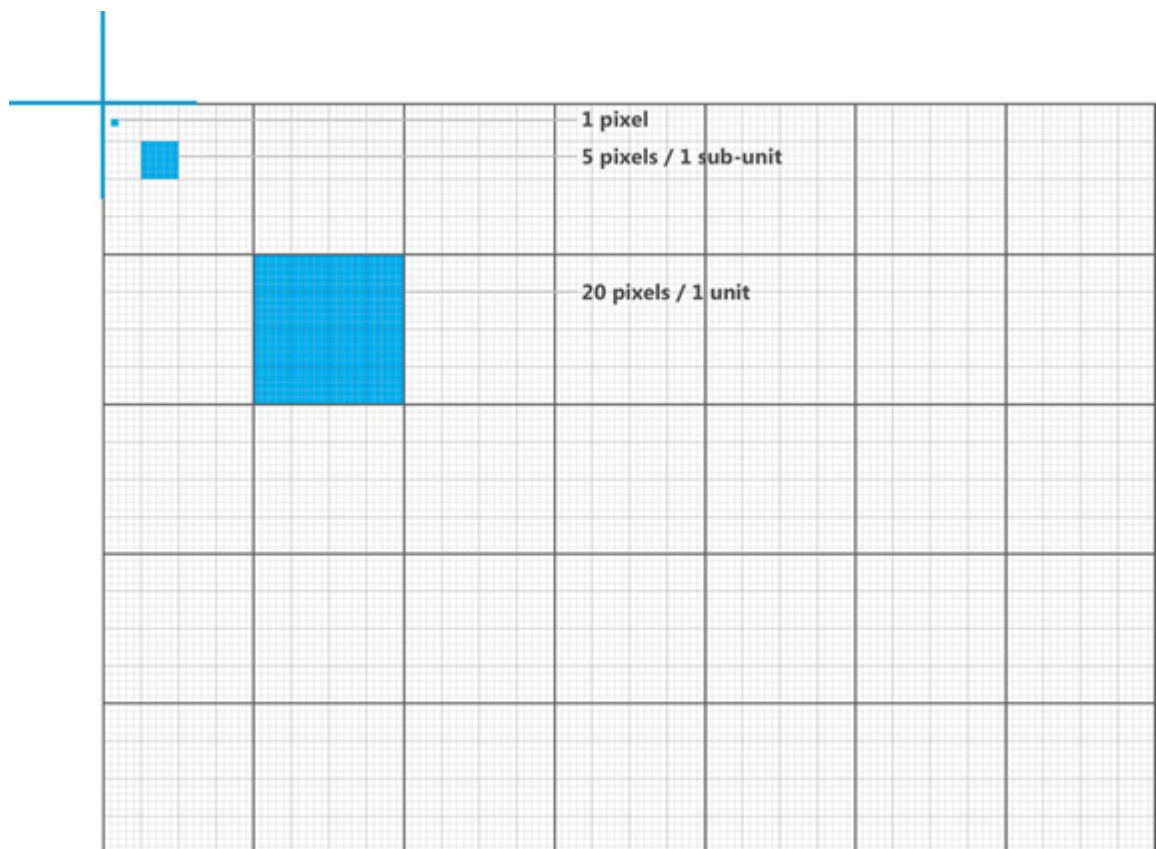
アプリレイアウトについての詳細は microsoft 公式の資料をご確認ください。

<http://msdn.microsoft.com/ja-jp/library/windows/apps/jj150600.aspx>

グリッドシステムへの準拠

ウェブページ開発と windows ストアアプリのもう一つの違いとして、要素を配置する際には microsoft が提供するグリッドシステムに準拠することがあります。

アプリの利用者にとって快適な体験になるように、windows ストアアプリ間で見栄えを揃えるためのものです。ぜひグリッドシステムを守ってレイアウトするようにしてください。



グリッドレイアウトについての詳細は microsoft 公式の資料をご確認ください。

<http://msdn.microsoft.com/ja-JP/library/windows/apps/hh872191>

トップページの編集方法

1. テンプレートを切り替える

3種類あるトップページのテンプレートを切り替えるには default.html を変更します。
例えばテンプレート"top-1"から"top-2"に変更するには以下のようにコードを修正します。

default.html [変更前]

```
<!-- ホームページを変更するには、下の要素の data-win-options 内の uri を変更します。
-->

<div id="contenthost" data-win-control="Application.PageControlNavigator"
data-win-options="{home: '/pages/top-1/top.html'}"></div>
```

default.html [変更後]

```
<!-- ホームページを変更するには、下の要素の data-win-options 内の uri を変更します。
-->

<div id="contenthost" data-win-control="Application.PageControlNavigator"
data-win-options="{home: '/pages/top-2/top.html'}"></div>
```

2. 各テンプレートの表示の違い

/pages/top-1/top.html



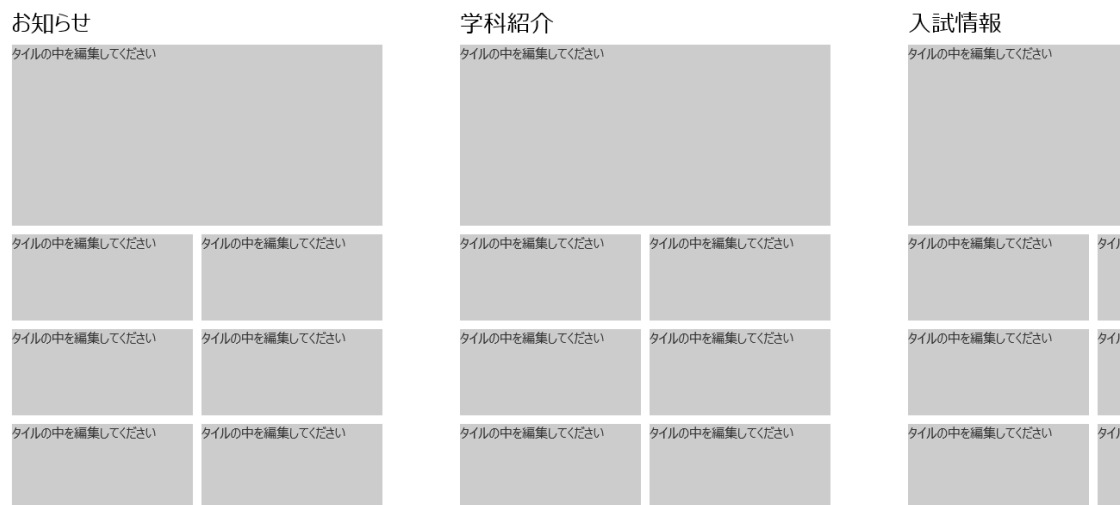
Finance アプリ風のデザインです。左側に画像があり、右側に細長のタイルが並ぶテンプレートです。

/pages/top-2/top.html



小さなタイルを中心に、大小のタイルで構成されるテンプレートです。

/pages/top-3/top.html



四角い大きめのタイルと細長タイルが全面に並ぶテンプレートです。

3. HTML 構造について

トップページの HTML 構造は各テンプレートでほとんど共通しています。

スタイルについては、テンプレート間で共通する部分が/css/top-base.css 内に定義されており、テンプレート固有のスタイルを実現するために各 top.css で top-base.css を上書き・追加する構造になっています。

「タイルの中を編集してください」と書かれている部分を編集するだけで目的が達成できるように設計されています。

また、/pages/top-1/top.html と /pages/top-2/top.html テンプレートは画面の幅や高さに応じてタイルを自動でレイアウトします。タイルを追加するごとに css ファイルを編集する必要はありません。

/pages/top-3/top.html テンプレートは画面幅に応じたタイルのレイアウトを行いません。タイルを追加しすぎた場合は画面からはみだしてしまいますので、10.6 インチの 1366x768 ピクセルの解像度で全てのタイルが表示されるかどうか確認してください。

4. 主なスタイルと機能について

タイルの大きさを変更する

トップページに表示されるタイルの大きさは2種類です。top 変更するには/css/top-base.css を変更します。変更箇所は以下のとおりです。

[スナップ表示以外のタイル]

```
@media screen and (-ms-view-state: fullscreen-landscape), screen and (-ms-view-state: fullscreen-portrait),
screen and (-ms-view-state: filled) {
    .toppage section[role=main] .zoom-in .category ul li {
        /*スナップ表示以外におけるタイルの基本サイズを変更するには以下の width, height プロパティを変更します。*/
        width: 210px;
        height: 100px;
    }
}
```

[スナップ表示以外で pickup クラスが指定されているタイル]

```
.toppage section[role=main] .zoom-in .category ul li.pickup {
    /* pickup クラスがついた場合は通常のタイルの縦横ともに2倍の大きさになります。*/
    /* pickup クラスの要素の横幅 = 通常タイルの横幅*2 + 通常タイルどうしの横の間隔 */
    width: 430px;
    /* pickup クラスの要素の縦幅 = 通常タイルの縦幅*2 + 通常タイルどうしの縦の間隔*/
    height: 210px;
}
```

[スナップ表示のタイル]

```
@media screen and (-ms-view-state: snapped) {
    .toppage section[role=main] .zoom-in .category ul li {
        /*スナップ表示におけるタイルの基本サイズを変更するには以下の width, height プロパティを変更します。*/
        width: 280px;
        height: 100px;
        margin: 10px 0;
    }
}
```

スナップ表示時には pickup クラスのタイルも同じ大きさで表示されます。

また、タイルの横幅をサイズを変更した場合は/pages/top-3/top.css も合わせて変更してください。

```
/* 以下の設定は異なる大きさを一定の順序で並べるための設定です。*/
@media screen and (-ms-view-state: fullscreen-landscape), screen and (-ms-view-state: fullscreen-portrait),
screen and (-ms-view-state: filled) {
    .toppage section[role=main] .zoom-in .category ul {
        -ms-flex-direction: row;
        -ms-flex-align: start;
        -ms-flex-pack: start;
        /* li 要素が2枚入る横幅に設定します */
    }
}
```

```
    max-width: 440px;
    height: auto;
  }
}
```

clickable クラスについて

clickable クラスが設定されている要素の直下にあるアンカー要素はタイルと同じ高さ・幅に設定されます。これはタイルのどの場所をクリックした場合でもアンカー要素のリンク先に遷移させるためです。アンカー要素の挙動については次項をご確認ください。

アンカー要素について

通常のアンカー要素は規定でクリックされた場合に href 属性に指定された URL をブラウザで表示します。

トップページの「リンク集」部分ではこの規定の動作を利用しています。

```
<!-- 以下のアンカー要素はクリック時にブラウザでページを表示します -->
<a href="http://www.microsoft.com/ja-jp/default.aspx">リンク</a>
```

一方、**app-navigation-link** クラスが指定されたアンカー要素は href 属性に指定された URI にアプリケーション内のページを遷移します。

```
<!-- 以下のアンカー要素はブラウザを開かず、アプリケーションに組み込まれた HTML
を表示します -->
<a class="app-navigation-link" href="/pages/detail/example/example.html">
リンク</a>
```

app-navigation-link クラスのアンカー要素に href 属性を設定するは以下の規則を守る必要があります。

1. 必ず`"/`から始めてください。
(例) □ `href="pages/example.html"` ○ `href="/pages/example.html"`
2. `"/`以下のディレクトリは **package.appxmanifest が存在するディレクトリを基準にしてください**。(ディレクトリ構成については「各画面と対応するファイル一覧」を参照してください。)

セマンティックズーム時のカテゴリ移動について

以下はテンプレート内に配置されているセマンティックズーム時に表示されるカテゴリ一覧の HTML です。各テンプレートとも完全に同じ構造になっています。

<!-- ズームアウト時のコンテナです -->

```
<div class="zoomable-view link-categories" data-win-control="App.UI.ZoomOutCategoryList">
  <div class="category-tile clickable" data-jump-to="#notice"><span class="category-name">お知らせ</span></div>
  <div class="category-tile clickable" data-jump-to="#course"><span class="category-name">学科紹介</span></div>
  <div class="category-tile clickable" data-jump-to="#admission"><span class="category-name">入試情報</span></div>
  <div class="category-tile clickable" data-jump-to="#link"><span class="category-name">リンク集</span></div>
</div>
```

data-jump-to 属性に注目してください。category-tile クラスが設定された要素がクリックされた場合、data-jump-to 属性に設定されたセレクトアの要素にジャンプします。例えば

```
<div class="category-tile clickable" data-jump-to="#notice">お知らせ</div>
```

と設定された div 要素は以下のように id 属性に"notice"と設定された要素にジャンプします。

```
<div id="notice" class="category">
  <h2>お知らせ</h2>
```

セマンティックズーム部分の HTML は通常変更する必要はありません。ですがカテゴリを追加したり、削除するときには上記を参考にして正しくセマンティックズームが動作するように変更してください。

詳細ページの編集方法

ディレクトリ構造について

詳細ページは/pages/detail ディレクトリ内に作成することを推奨します。

/pages/detail ディレクトリ内には自由にサブディレクトリを作成しても構いません。

HTML の作成について

詳細ページを新たに作成する場合は/page/detail/template.html をコピーして編集することで windows8 のガイドラインに適合したページを少ない手間で作成することができます。

template.html の編集例は detail/example/example.html をご覧ください。

詳細ページのスタイルを指定するときの注意点

各詳細ページで読み込まれた css ファイルはページ遷移後も他の全てのページに適用されます。

また、default.html とトップページで読み込まれた css ファイルも全ての詳細ページに適用されることにも注意してください。

windows ストアに提出する前に

アプリ画像を規定のものから差し替える必要があります。

アプリ画像は/images ディレクトリ内にあります。

アプリ画像に関する詳細は microsoft 公式の資料をご覧ください。

<http://msdn.microsoft.com/ja-jp/library/windows/apps/hh846296.aspx>