

スマートフォン向けのCSSを読み込む

スマートフォンのように画面サイズの小さい端末で見たときは、専用のCSSを読み込むようにします。

🕒 画面サイズが小さいときはレイアウトを変更する

スマートフォンなど、パソコンに比べて画面の小さい端末で Web サイトを閲覧しようとする、文字や画像が縮小され、ものすごく小さく表示されることがあります。そうならないように、画面サイズが小さいときはページのレイアウトを変更します。

レイアウトの変更は CSS だけで行います。「メディアクエリー」と呼ばれる CSS の新機能を使って、画面サイズが小さいときだけ、追加の CSS ファイルが読み込まれるようにします。

【本節で記述するメディアクエリーの動作概要】

HTML

```
<link rel="stylesheet" href="css/style.css"> .....①
```

```
<link rel="stylesheet" href="css/responsive.css" media="screen and (max-width: 480px)">.....②
```

条件（メディアクエリー）

画面サイズが小さく、条件を満たす場合

①と②を読み込む

画面サイズが大きく、条件を満たさない場合

①だけ読み込む

🕒 メディアクエリーを使って別のCSSファイルを読み込む

画面の横幅が 480 ピクセル以下のスマートフォンで閲覧したときには、style.css、common.css に加え、追加の CSS ファイルとして「css」フォルダーの中にある「responsive.css」を読み込むようにします。

■ <link> タグを追加する

❶ index.html を開きます。<head> 内のすでにある <link> タグの下にもう 1 行、<link> タグを追加します。

【index.html】

```
...
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width,initial-scale=1.0">
<title>HOTEL IMPERIAL RESORT TOKYO</title>
<link rel="stylesheet" href="css/style.css">
<link rel="stylesheet" href="css/responsive.css" media="screen and (max-width: 480px)">
</head>
...
```

2 index.html をブラウザで開きます。ブラウザのウィンドウを狭めていくと、途中でレイアウトが切り替わります。

【通常のウィンドウ幅】



【幅 480 ピクセル以下】



4

高度なリストのデザイン

解説 レスポンシブWebデザイン

スマートフォンやタブレットの登場で、パソコン以外の端末で Web サイトを閲覧する機会が増えました。HTML はいっさい切り替えず、画面サイズなどの条件に応じて適用する CSS だけを変えることで、端末に適したレイアウトを実現する手法があります。それが「レスポンシブ Web デザイン」です。レスポンシブ Web デザインを実現するには、次の 2 つの作業をします。

- HTML に `<meta name="viewport">` タグを追加する
- メディアクエリーを使用して CSS を切り替える

■ `<meta name="viewport">` タグの追加

スマートフォンやタブレットのブラウザが Web ページを表示するとき、そのページがパソコン向けに、横幅 980 ピクセル^{*1} の固定幅で作られていると仮定して一度描画し、その後端末の画面サイズに合わせて縮小表示します。そのため、パソコン向けに作られたページはものすごく小さく表示されることがあります。

【スマートフォンが Web ページを表示するときの通常の処理】



<meta name="viewport"> タグを使用すると、横幅を仮定して描画し、それを縮小表示するという通常の処理手順をキャンセルすることができます。

画面幅が 480 ピクセル以下のときに適用される responsive.css には、ページの横幅を固定せず、常に画面幅の 95% になるような CSS が書かれているため、横幅を仮定して縮小表示する必要がありません。そこで、<meta name="viewport"> タグを使って、通常の表示処理をキャンセルしています。

*1 iOS の場合。Android 端末は機種によって異なる

【表示処理をキャンセルするときの <meta name="viewport"> の書式】

```
<meta name="viewport" content="width=device-width,initial-scale=1.0">
```

■ メディアクエリーの使用

「メディアクエリー」と呼ばれる条件文を用いれば、画面サイズの違いなどに応じて、適用する CSS を切り替えることができます。メディアクエリーを使用するには、HTML に <link> タグを追加するか、もしくは CSS に @media ルールを追加します。

●HTML に <link> タグを追加する

HTML の <link> タグを使って、条件に適合したときの CSS を読み込ませます。条件を指定するには、<link> タグの media 属性を使います。

【<link> タグの media 属性を使用する書式】

```
<link rel="stylesheet" href="CSSファイルのパス.css" media="メディア特性 and (メディアクエリー)">
```

●CSS に @media ルールを追加する

@media ルールを使うと、すでにある CSS ファイルに、条件に応じて切り替える CSS を追加することができます。つまり、responsive.css に書かれた CSS を、style.css に丸ごと移せるわけです。具体的な例を確認したいときは、サンプルの「c04-responsive」フォルダーをご覧ください。

【@media ルールの書式】

```
@media メディア属性 and (メディアクエリー) {  
    /* 条件に適合したときのCSSをここに記述する */  
}
```

【style.css に @media ルールを使って記述する例 (サンプル: c04-responsive /css/style.css)】

```
...  
/* @mediaルールを使ったレスポンシブなCSSの記述例 */  
@media screen and (max-width: 480px) {  
    /* 画面の横幅が480px以下のときに適用されるCSSはすべてここに記述する */  
    header, nav, #graphic, #contents, footer {  
        margin: 0 auto;  
        width: 95%;  
    }  
    ...  
}
```

【「メディア属性」の主な値】

属性値	説明
screen	画面用の CSS を指定
print	プリント用の CSS を指定

【「メディアクエリー」の主な値】

メディアクエリーの書式例	説明
(max-width:480px)	画面幅が 480 ピクセル以下のとき CSS を適用
(min-width:600px)	画面幅が 600 ピクセル以上のとき CSS を適用
(max-height:500px)	画面高が 500 ピクセル以下のとき CSS を適用
(min-height:768px)	画面高が 768 ピクセル以上のとき CSS を適用
(orientation:portrait)	画面幅が高さ以下のとき、つまり端末を縦に持っているとき CSS を適用
(orientation:landscape)	画面幅が高さ以上のとき、つまり端末を横に持っているとき CSS を適用

関連する情報を近くにまとめたり¹⁾、ボーダーラインで囲んだりしてグループ化すると、ページの内容が整理されて読みやすくなります。

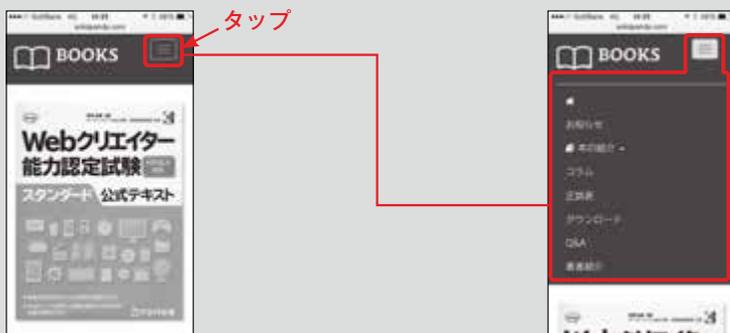
Webデザインならではのグループ化手法の1つに「開閉」があります。掲載する情報が多いときや、常に見えている必要がないものを使用すると効果的です。

¹⁾ 第9章「近接」P.249

■ スマートフォンのナビゲーション

スマートフォンでは、パソコンのようにナビゲーションのリンクを横に並べさせるのは現実的ではありません。だからといってリンクを縦に並べて、狭い画面のかなりの部分を占有してまで見せておく必要もありません。そこで、画面を有効に利用するために、ナビゲーションを開閉できるようにすることがあります。

【開閉するナビゲーションの例】

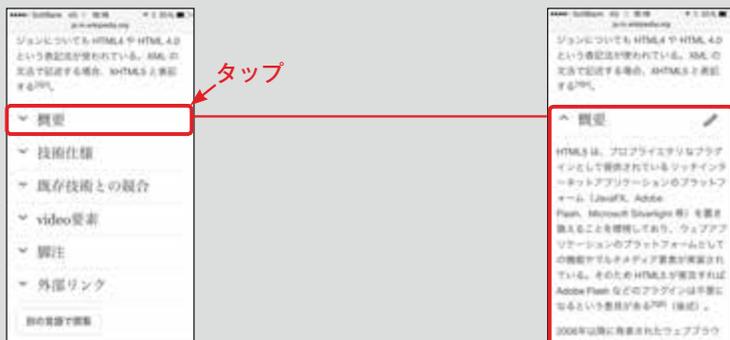


スマートフォンサイトで見かけるナビゲーション。一般的に3本線のアイコンを付けることが多いため、「ハンバーガーマニュー」と呼ばれている。

■ アコーディオン

情報量の多いページをコンパクトに見せる方法としてよく用いられるのは、一度見出しをクリックすると記事が表示され、もう一度クリックすると記事が隠れるというものです。一般的に「アコーディオン」と呼ばれるデザイン手法です。

【アコーディオンの例】



スマートフォン版のウィキペディア。見出しをタップするとテキストが表示されるようになっている。

■ 開閉を実現するには

スマートフォンのナビゲーションやアコーディオンなど、Webページで「開閉」を実現するには、JavaScriptとCSSを組み合わせます。JavaScriptが書けなくても、どういう場面で使うのが効果的か、デザインのアイデアとして知っておいた方がよいでしょう。