レスポンシブWebデザイン（RWD）の基本設計　　　　　　2017.12.28

スマホとパソコンの画面ではサイズが大きく異なることから、どちらでも快適に閲覧できるようにするにはページのレイアウトを変える必要があります。

このレイアウトを切り替える方法に

①スマホとパソコンで別々のサイトを作る

②共通のHTMLを使用しつつ、CSSだけを一部切り替えて、どんな端末で閲覧しても最適の

 レイアウトにして表示する

という2つの考え方があり、②での構築方法がレスポンシブWebデザイン（RWD）といいます。

◇ RWDに必要な主たる機能

(1)メディアクエリを使って、画面サイズに合わせてレイアウトを切り替える

　① スマホ･ブラウザ機能の縮小表示機能をキャンセル

　② 画面サイズに合わせてCSSを切り替える

(2)ウィンドウ幅に合わせて伸縮するレイアウトにする

(3)ページ内で使用する画像の一部も、固定ではなく、ウィンドウ幅に合わせて伸縮表示が

 できるようにする

(4) Responsive Web Design Testing Tool【 http://mattkersley.com/responsive/ 】

 ウェブサイトのURLを入力すれば、異なるウィンドウ幅の画面を同時に表示でき、

 RWDの検証が出来る

以下に、枚方南支部のRWD対応の基本となるhtml構造および関連ファイルについて

説明する。

**１）**RWDの基本html構造（テンプレート）

外部CSSもhtml内CSSもメディアクエリーを使い、パソコン、スマホの画面サイズに合わせて

レイアウトを切り替える。画面サイズは768pxをブレークポイントとして設定する。

CSSの構成はパソコンおよびスマホの両方を記述して、対比して理解しやすい方式にする。

<meta charset="utf-8">

<title>枚方南支部RWD</title>

<link href="../../css/**hira-r.css**" rel="stylesheet" media="print,screen and (min-width: 768px)">

<link href="../../css/**hira-s.css**" rel="stylesheet" media="only screen and (max-width: 767px)">

<link href="../../colorbox/**colorbox.css**" rel="stylesheet">

<link href="../../jquery/to-top.css" rel="stylesheet">

・・・・・・・・・・・・・・・・・・

<script src="../../jquery/**viewport.js**"></script>

<script src="../../colorbox/jquery.colorbox-min.js"></script>

<script src="../../jquery/to-top.js"></script>

<script src="../../jquery/analytics.js"></script>

<style>

<script>

$(function(){

 $(".group1").colorbox({rel:'group1',maxWidth:'100%',maxHeight:'100%'});

 $(".group2").colorbox({rel:'group2',maxWidth:'100%',maxHeight:'100%'});

　　　・・・・・・・・・・・・・・・・・・

 $(".youtube").colorbox({iframe:true,innerWidth:560,innerHeight:315,maxWidth:'100%',maxHeight:'100%'});

});

</script>

<style>

<!—

/\* 共通css \*/

・・・・・・・・・・・・・・・・・・

/\* 768px以上css \*/

@media print,screen and (min-width: 768px) {

・・・・・・・・・・・・・・・・・・

}

/\* 767px以下 css\*/

@media only screenand (max-width: 767px) {

・・・・・・・・・・・・・・・・・・

｝

-->

</style>

</head>

<body><!--トップメニュー-->

<div id="upper\_navi\_g" ><iframe src="../**navi\_menu3.html**" class="navisize"

title="トップメニュー"></iframe></div>

<!--ページ本体-->

<div id="container">

・・・・・・・・・・・・・・・・・・

 <ul class="img-yoko2 fnt80">

 <li><a class="group1" href="s17092611.jpg" title="乗務員"><img src="s17092611.jpg" alt=" "></a><br></li>

 <li><a class="group1" href="s17092612.jpg" title="大橋"><img src="s17092612.jpg" alt=" "></a><br></li>

・・・・・・・・・・・・・・・・・・

**２）**hira-r.cssについて（rはレスポンシブの意）

RWD:768PX以上の標準CSS外部ファイルとなり、従来のhira-mに該当して、内容はほぼ変更なし。

変更対応しやすくするためルートフォルダーCSSにおき、各サイトが共通で使う。

/\* new 文章の行端揃え 2017/10 \*/

.justify{

 padding: 0; /\* 2017/10新規追加 \*/　　　・従来のreport機能で左右の余白をとらず

 text-align: justify;

 text-justify: inter-ideograph; }

ナビメニューのプリントをしない対応は今回はhira-r.cssから外して、ナビメニューnavi\_menu3.htmlの中に

@media print{.insatsu{display:none;}}を入れて、メニューhtmlの中のclass="insatsu"でメニューは

印刷しないことにしてます。これで各ページのclass="insatsu"は不要です。

**３）**hira-s.cssについて（sはスマホの意）

RWD:767PX以下の標準CSS外部ファイルとなり、ルートフォルダーCSSにおき、共通で使う。

body{

 background-color: #ffffff; /\* hira-sで追加　\*/　　・画面は白バック

 margin: 0px;

 padding: 0px;

 text-align: center;

 color: black; /\* hira- sで変更 \*/　　・文字色は黒

 font-family: "メイリオ", "Meiryo";

 line-height: 1.6; /\* hira-sで変更 \*/　　・行間は広めにline-height: 1.6

}

html{

 font-size: 16px; /\* hira-sで追加 \*/　　・文字の標準大きさは16px

 }

iframe{

 background-color: transparent;

 border : none;

}

#container {

 width: 98%; /\* hira-sで変更 \*/　　・画面に左右に2%の余白

 margin: 0 auto;

 padding-top: 44px; /\* hira-sで変更 \*/　 ・トップメニューの幅44pxとる

}

/\* new 文章の行端揃え 2017/10 \*/

.justify{

 padding: 0; /\* 2017/10新規追加 \*/　　　・従来のreport機能で左右の余白をとらず

 text-align: justify;

 text-justify: inter-ideograph;

}

/\* トップメニューのサイズ \*/ ・トップメニューの画面上下幅の変更

/\* 479px以下 \*/

/\* 319px以下 \*/

（注）従来からの.reportもそのままあります。

**４）**viewport.jsについて

スマホ、タブレットの画面表示領域（viewport）はuserAgentで判別して、

スマホについてはRDW対応した画面幅をそのままで表示し、

タブレットについてはパソコン画面を縮小した表示をして、これに沿ってページ構築する。

// userAgentによるviewportの設定（枚方南支部）2017.9.10

$(function () {

var ua = navigator.userAgent;

if((ua.indexOf('iPhone') > 0) || (ua.indexOf('Android') > 0 && ua.indexOf('Mobile') > 0) || (ua.indexOf('Windows Phone') > 0)){

 $('head').prepend('<meta name="viewport" content="width=device-width,initial-scale=1">');

}else{

 $('head').prepend('<meta name="viewport" content="width=900">');

}

});

参考）デスクトップ環境（PCなど）のブラウザでは、特にviewportは意識しないですが、

　　　ブラウザウインドウの表示領域がviewportになります。

　　　ページがviewportからはみ出た場合は、スクロールをしてviewportを変えてます。

**５）**トップメニューnavi\_menu3.htmlについて

画面幅に合わせて伸縮するレイアウトにする。

display: flex; の伸縮機能を使用（逆に固定フォーマットは難しい）

/\* 768px以上 \*/

@media print,screen and (min-width: 768px) {

nav{

 width: 820px;

 margin: 0px auto; }

nav ul{

 list-style: none;

 display: flex;}

nav li {

 flex: 1 1 auto;}

nav li a{

 display: block;}

}

/\* 767px以下 \*/

@media only screen and (max-width: 767px) {

nav{

 width: 100%;

 margin: 0px auto; }

nav ul{

　　　list-style: none;

　　　display: flex;}

nav li {

 flex: 1 1 auto;}

nav li a{

 display: block;}

}

/\* 　　パソコン画面用css　　 \*/

@media print,screen and (min-width: 768px) {

nav{

 width: 820px;

 margin: 0px auto; }

nav ul{

 list-style: none;

 display: flex;}

nav li {

 flex: 1 1 auto;}

nav li a{

 display: block;}

}

**6）**colorbox の説明文（タイトル）について

colorboxでは説明文の記入欄はtitle表示として扱っていて、長いサイズの文は考慮されていなかったが、

今回colorbox.cssをカスタマイズして、以下の制限の中で多少長めの文は入るように運用する。

なお、スマホ画面の説明文の文字は13pxに固定してますので、

パソコン画面の文字サイズを変更しても、字数制限は変わりません。

①colorbox.cssを使用（**パソコン画面およびスマホ画面：1行表示**）➡　標準とする

・説明文は最大**19文字以内**で記載する。

その他のオプションとして

②colorbox1.cssを使用（**パソコン画面：1行表示、スマホ画面：2行表示**）

・パソコン画面は1行表示で最大**30文字以内**で記載する。

③colorbox2.cssを使用（**パソコン画面：2行表示、スマホ画面：2行表示**）

・パソコン画面は2行表示で最大**44文字以内**で記載する。

**7）**単位について（px、em、rem、%）

Webページの代表的なレイアウトは3種類あり、width,margin,paddingを全てpxで指定する

「固定幅レイアウト」、全てを%で指定する「フルードデザイン」、全てを相対単位であるemで指定する

「Elasticレイアウト」です。

「%」、「em」単位はhtml要素のfont-sizeを変更するだけで、子要素の指定値が連動して、変更されるメリットはあるが、直感的にそのサイズが判断しにくく必ずPX単位での置き換えが必要なことや、親要素との関係で分かりにくいこともあり、スマホhtmlで文字16pxを基準とするなら、伸縮を必要とするレイアウト、画像については「%」指定をするが、他の単位は判断しやすい「px」とする。

html {

  font-size: 100%} 　　　　　/\* =16px \*/

body {

  font-size: 2em; } 　　　　　/\* =32px \*/

p {

  font-size: 1em; }　　　　　　/\* =32px \*//\* font-size: 0.5em; =16px \*/

bodyの「2em」は、親要素であるhtmlを継承するので「32px」になります。

pの「1em」は、親要素であるbodyの値を継承するので「32px」になります。

**8）**今までにRWD対応をしたページ

①トップページ ⑩行事案内（トップページ） ⑲松市会　（最新一式）

②支部長の挨拶（トップページ） ⑪2017年 65歳節目懇談会 ⑳どんぐり会（最新一式）

③2017年 行事予定（トップページ） ⑫2017年 歴代役員懇談会 ㉑enjoy shoai

④行事報告（3）（トップページ） ⑬2017年 支部総会 ㉒マジック同好会 （最新一式）

⑤地区委員会報告9月度（トップページ） ⑭2017年 社会見学会

⑥枚方南支部だより（24）（トップページ） ⑮2017年 新春懇親会

⑦お願い事項（トップページ） ⑯2017年 秋レク

⑧ふるさと枚方発見表紙（トップページ） ⑰2017年 女性会員懇談会

⑨同好会表紙（トップページ） ⑱茶道同好会

**9）**新たにRWDに切り替えるときの手順

①RWDテンプレートから今回使用する<link ><script>の外部ファイルを入れ替えする。

②html内CSSをつくる。

<style>

<!--

**/\* 共通 \*/**

両者に共通のcssを記入

**/\* 768px以上の画面 \*/**

**@media print,screen and (min-width: 768px) {**

現在、使用しているcssを記入

**}**

**/\* 767px以下の画面\*/**

**@media only screen and (max-width: 767px) {**

スマホ用のcssを対比させて書いていく

**}**

-->

</style>

②-1．現在、設定しているhtml内CSSを/\* 768px以上の画面 \*/のcss部分に移す。

②-2.　このタイミングで、パソコン画面は正常なことを確認しておく。

②-3.　スマホ画面での表示形態を決める。

 ・写真画像の表示（横1枚、2枚、3枚）と配置（回り込み）

 ・ロゴ、表題文字、区切り画像の大きさ

 画像、ロゴはスマホ画面で表示する大きさを検討して基本は伸縮可能な％設定（一部はpx設定）

 で/\* 768px以上の画面 \*/cssと対比しながら記載していく。

②-4.　html内cssで/\* 768px以上の画面 \*/cssと1対1で対応させて/\* 767px以下の画面\*/ cssを

　　　 記載する。両者共通でいけるcssは/\* 共通 \*/に記載する。

 最初は、両者の対比表をつくり進めると漏れなく、わかりやすくなる。

②-3.　このタイミングでも、パソコン側の画面は正常なことを確認しておく。

②-4.　以後、ページが同様のフォーマットで標準化できる場合はこれらのhtml内CSSをまとめて、

　　　 CSS外部ファイルに切り替える。

③htmlのRWD対応変更

③-1．トップメニューの切り替え

<div id="upper\_navi\_g"><iframe src="../../navi\_menu3.html" class="navisize" title="トップメニュー"></iframe></div>

③-2．画面幅をオーバしているものをチェックする。

　　　基本は横画面でスクロールしない定着する画面にする。

③-3．html内で直接style でpxで指定しているwidth,margin,paddingについて

　　　スマホ画面幅（目安350px以下）をオーバする場合は、classタグでCSSに切り替え、

　　　適当なサイズに設定する。

　　　今後は出来るだけstyleでのwidth,margin,padding指定は避けること。

　　　テーブルの各セルのサイズ指定も％を使うとテーブル幅の伸縮が出来、RWDの設定が楽になる。

③-4．写真はcolorboxへの切り替えをする。

④スマホ画面の実機での確認をする

⑤スマホでキャッシュして確認する場合が必要

　html以外のファイル（css､js）は変更してもスマホではすぐ反映しない場合があり、この場合は

　スマホのキャッシュ（iPhoneではSafariでのキャッシュクリア）が必要。

**10）** RWD対応のCSSテクニック

(1) 画像の％指定（画面横1枚表示、画面横2枚表示）

ページ内で使用する画像も、固定サイズではなく、画面サイズに合わせて伸縮して表示できるようにする。

＜img＞タグに「width:100%;」を指定すると、画像を親要素の幅に合わせて、縦横比を保ったまま伸縮

する。通常は#container「width:98%;」(パソコンは850px)が親要素になるので＜img＞「width:100%;」

としても画面98%の画像表示となる。

画質の劣化を防ぐには画像の親要素にmax-width: 600px;をいれておくのがよい。

⑫歴代役員懇談会　⑪65歳節目懇談会を参照。

/\* 767px以下 \*/

@media only screen and (max-width: 767px) {

ul img{

 width: 100%;　　　　　　 1枚表示

 height: auto;

　　　　 max-width: 600px;　　　　最大横600px

}

ul.img-yoko li{

 display: block ;　　　　　　縦に並べる

 margin: 0 3px ;

}

ul.img-yoko2 li{

 display:inline-block;　　　floatで横に並べる

 float: left;

 width: 48%;　　　　　　 48%にして**2枚並べる**

 height: auto;

 margin: 0 1%;　　　　　 余白は左右１％

}

/\* 768px以上 \*/

@media print,screen and (min-width: 768px) {

ul img{

 width: 200px;

 height: 150px;

}

ul.img-yoko li{

 display: inline-block;

 margin: 0 3px;

}

ul.img-yoko2 li{

 display: inline-block;

 margin: 0 3px;

}

}

(2) スマホ画面表示だけを改行して見やすくする (display:block;)

・パソコンで<br>で文を修飾している場合、スマホ画面では画面幅による文の改行が頻繁に発生する

　ので、<p></p>に切り替えるか、後述の<span> disply：block；、<br> disply：none；を使い、

　見え方の工夫をする。

・１．とか１）とか①を使った短文でもスマホではすぐ2行、3行になるので、2行目以降を

　1文字ずらしをすれば、見やすくなる。

例：

【html】

<p>・今も仕事をしている 12名　<span>・働く予定なし 10名　</span><span>・未定 1名</span></p>

【CSS】

/\* 767px以下 \*/

@media only screen and (max-width: 767px) {

span{

 display: block;}

}

・今も仕事をしている 12名

・働く予定なし 10名

・未定 1名

/\* 768px以上 \*/

@media print,screen and (min-width: 768px) {

span{

 display: inline;}

}

・今も仕事をしている 12名　・働く予定なし 10名　・未定 1名



(3) 2行目以降の1文字ずらしで、最初の1文字を前に出す

例：

〇参加者の皆さまから本日、自己紹介で

　お聞きしたこと（23名の内訳）

【html】<p class="ind1-">〇参加者の皆さまから本日、自己紹介でお聞きしたこと（23名の内訳）</p>

【CSS】

/\* 共通 \*/

.ind1-{

 text-indent: -1em;

 padding-left: 1em;

}

(4) htmlソースを削除する　(display：none；を使う)

パソコン画面のみhtmlソースを削除する、あるいは逆にスマホ画面のみhtmlソースを削除する場合に使用するが、頻繁に使うと複雑になるので、考慮して使用すること。

【html】 <h1>2017年度<br class="none">枚方南支部</h1>

【CSS】

/\* 768px以上 \*/

@media screen and (min-width: 768px) {

br.none{ display: none;}

/\* 767px以下 \*/

@media only screen and (max-width: 767px) {

br.none{ display: block;}

【html】<div class="none" style="clear:both;"></div>

【CSS】

/\* 768px以上 \*/

@media print,screen and (min-width: 768px) {

.none{ display: none;}

(5) head内cssは外部ファイルにして簡略化する

　　RWD作成以後は同じフォーマットで使うように標準化して、head内cssを全て外部ファイルに

　　しておく。

**11）**ページを標準化すればhead内cssは外部ファイルに出来る　　enjoy shoai の例（html全文）

<!DOCTYPE html>

<html lang="ja">

<head>

<meta charset="UTF-8">

<title>enjoy shoai地区委員会員　徳田さん</title>

<link href="../../css/hira-r.css" rel="stylesheet" media="print,screen and (min-width: 768px)">

<link href="../../css/hira-s.css" rel="stylesheet" media="only screen and (max-width: 767px)">

<link href="../../slick/slick.css" rel="stylesheet">

<link href="../../slick/slick-theme.css" rel="stylesheet">

<link href="../../image/css/font-awesome.min.css" rel="stylesheet">

<link href="../enjoy-rwd.css" rel="stylesheet">

<link href="../../jquery/to-top.css" rel="stylesheet">

<script src="../../jquery/jquery-3.1.1.min.js"></script>

<script src="../../slick/slider2rwd.js"></script>

<script src="../../slick/slick.min.js"></script>

<script src="../../jquery/viewport.js"></script>

 <script src="../../jquery/to-top.js"></script>

<script src="../../jquery/analytics.js"></script>

<style>

<!--

/\* 共通 \*/

/\* 768px 以上 \*/

@media print,screen and ( min-width: 768px ){

}

/\* 767px 以下 \*/

@media only screen and ( max-width: 767px ){

}

-->

</style>

</head>

<body>

<!--トップメニュー-->

<div id="upper\_navi\_g"><iframe src="../../navi\_menu3.html" class="navisize"

 title="トップメニュー"></iframe></div>

<!--本文-->

<div id="container">

<!-- メイン -->

 <div><img src="../../image/enjoyshoai8.gif" class="emo2" alt="enjoy shoai">

 <img src="../../image/enjoygroup.png" alt="enjoy shoai" class="emo1"></div>

 <h1><b>徳田 博</b></h1>

 <p class="right"><i class="fa fa-calendar" aria-hidden="true"></i>2017.10.1</p>

 <h3 class="right">支部担当 7班地区委員</h3>

 <div class="float-r img1"><img src="htokuda00.jpg" alt="enjoy shoai" class="img1" ></div>

 <div class="fnt110 justify" >

 <h2>お世話役活動の効用！</h2>

 <p>愛媛で生まれ育った田舎者がAVC社の経理部門一筋でお世話になった後、2013年1月に定年退職を迎えました。

 体調面（2012年8月に右目の緑内障手術）のこともあり･････････････････宜しくご指導、ご協力をお願い致します。</p>

 </div>

 <p class="fnt80">≪自動スライドショーです。写真にマウスを当てると停止して、外すと動き出します≫</p>

 <ul class="slider iti"><!--スライド2列写真-->

 <li><img src="htokuda06.jpg" alt="2017.9秋レク車両長少し緊張？">2017.9　秋レク１号車車両長</li>

 <li><img src="htokuda07.jpg" alt="2017.9秋レク車両長として挨拶">2017.9秋レク１号車車両長</li>

 <li><img src="htokuda09.jpg" alt="2017.9　秋レク中野うどん学校">2017.9 "中野うどん学校</li>

 </ul>

 <!--戻るボタン--><br>

 <img src="../image/roseline1.gif" class="roseline" alt="roseline">

 <a class="back" href="javascript:history.back(-1)">&laquo;&nbsp;Back</a>

 </div><!--container-->

 <!-- ここからpage-topボタン -->

 <div style="margin-top:3em;"></div>

 <div id="page-top" class="page-top">

 <p style="font-size: 60%; text-indent: 0em;"><a id="move-page-top" class="move-page-top">

 <span style="font-size: 18px;">▲</span><br>ページ<br>トップ</a></p>

 </div><br><!-- ここまでpage-topボタン -->

</body>

</html>

**head内cssは外部ファイルで固定**

**head内cssは都度記載必要なし**

**顔写真**

**紹介文**

**スライド写真**