

CHAPTER

5

1. 今日の作業に使用するフォルダやファイルを準備しよう！
2. レスポンシブ WEB デザインとは
- 3.viewport 定義を head 要素内に指定する
4. 横幅変動時のコンテンツ幅を設定する
- 5.Media Querie(メディアクエリ) を使用してデザインを整える
6. 画面サイズに合わせた表示 (ハンバーガーメニューなど) を導入する。

今日の作業に使用するフォルダやファイルを準備しよう！

使用するファイルの確認

- + chapter5.zip
 - + chapter5.pdf…手順を記した教科書です
 - + design…デザインに関する情報
 - + デザインカンパの画像ファイルなど
 - + content…WEB ページに使用する素材
 - + サイト制作で使用する画像・原稿ファイルなど
- + flowerdays-04…chapter4 で作成したサイトファイル
 - + images
 - + 画像ファイル
 - + index.html
 - + css
 - + reset.css
 - + style.css
- + flowerdays-05…完成サイトファイル (解答編)

今日の作業用のファイルを準備する

chapter.4 の内容が最後までできている人は前回の続きに記述しても構いませんし、もし不安がある場合は chapter5 > flowerdays-04 に chapter.4 の作業が終わったファイルがありますので、そのファイルを使用します。

前回の続きに記述する方

デスクトップにある flowerdays-04 フォルダをコピーして、フォルダ名を flowerdays-05 に変更します。

ダウンロードしたファイルを使用する方

chapter4.zip の中の flowerdays-04 フォルダをデスクトップにコピーし、フォルダ名を flowerdays-05 に変更します。

flowerdays -05 は完成させたファイルです。最後に見比べてみましょう。

レスポンス WEB デザインとは

スマートフォンやタブレットからのインターネットアクセスが増加し、WEB サイトの制作側もユーザーの閲覧環境の変化に合わせてどんなデバイスでもストレスなく閲覧できることが求められるようになりました。ユーザーが閲覧する端末 (PC、スマホ、タブレットなど) の画面サイズに合わせてページレイアウトを最適化する方法は二つあり、それぞれにメリットとデメリットがあります。

PC 版、タブレット版、スマホ版などそれぞれに別のページを用意する方法

- それぞれの環境ごとに表示する内容を大きく変えられる。
- デバイスごとの作成になるため、混乱しにくい。
- × デバイスごとに HTML 文書を作成するため、文言の修正などがあった場合それぞれに修正しなければならず、チェックの手間がかかる。
- × それぞれの HTML 文書ごとに URL が異なるため、異なるデバイス間で共有しにくい。

複数のデバイスで一つのページを共有する方法…レスポンス

- HTML 文書が一つなので修正や更新が一度で済み、チェックも一度で済む。
- URL も共通なので、異なるデバイス間でも共有しやすい。
- Google などの検索エンジンが推奨している「モバイルフレンドリー」に対応し、SEO の効果がある。
- × 同じセレクトに対して条件分けの記述をする必要があるため、CSS が複雑になり、そのぶん重くなる。
- × 文言や内容そのものを大きく変更することは出来ない。

レスポンスデザインを実装することによって PC 版とモバイル版など複数ファイルを管理する必要がなくなる一方で、「表示するデザインが元のページに依存する」といったデメリットもありますが、この章では2つめの方法を解説していきます。

Point!

スマートフォンなどでページを閲覧した時にユーザーにストレスを与えないような配慮を『モバイルファースト』と言います。

スマートフォンは画面が小さく、屋外や移動中に閲覧することが多いため、フォントが小さすぎると読みにくかったり、ボタンなどのクリックできる領域が小さすぎると指ではタップできなかったりとも見づらいサイトになってしまう可能性があります。

「スマートフォン向けサイトを優先的に制作・公開する」という意味で、モバイルファーストという言葉が使われることもありますが、必ずしもスマートフォン向けサイトを先に公開したり PC ユーザーをないがしろにするという意味ではありません。

ユーザーの利用状況などの情報を元に Web サイトを設計し、スマートフォンでもストレスなく利用できるデザインにするということです。

レスポンス WEB デザインの手順

文書そのものは1つのHTMLに記述しておき、CSSの指定で「画面幅〇〇ピクセル以上の場合はA、それ以下の場合はB」というふうに条件分けして指定します。

最初に大きな画面(PC)を作成し画面の幅が小さくなるにしたがってデザインを変更する方式と、最初に小さい画面(スマホなど)を作成し画面の幅が大きくなるに従ってデザインを変更する方式がありますが、ここではPC→(タブレット)→スマートフォンの順に作っていくことにします。

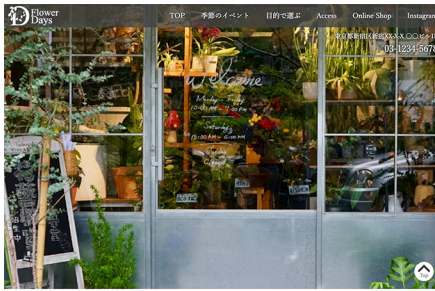
1. PC版サイトの作成 ←完了
2. viewport 定義を head 要素内に指定する
3. 横幅変動時のコンテンツ幅を設定する
4. Media Querie(メディアクエリ)を使用してデザインを整える
5. 画面サイズに合わせた表示(ハンバーガーメニューなど)を導入する。

1. PC 版サイトの作成 ←完了

メインコンテンツの横幅は 960px(width: 960px;)

デベロッパーツールを開いた状態で表示領域右上に表示されるウィンドウサイズを確認しながら幅を変化させて、どの時点で横スクロールバーが表示されるか確認してください。

- PC 画面でブラウザの横幅を小さくした時、横スクロールしないと全体を読めない。
- スマホまたはデベロッパーツールのデバイスモードで、スマホ画面での表示を確認
→そのまま縮小されたような画面になり、字が小さすぎて読めない。



2024
5/12
Saturday

Mother's Day

他にはない種類・カラーのお花からメインのお花をお選びいただき、イメージ・ご予算に合わせてスタッフが一つ一つ丁寧に作りします。
お花を贈るのがはじめての方でも安心して当店にお任せください。



2024
5/12
Saturday

Mother's Day

他にはない種類・カラーのお花からメインのお花をお選びいただき、イメージ・ご予算に合わせてスタッフが一つ一つ丁寧に作りします。
お花を贈るのがはじめての方でも安心して当店にお任せください。



目的で選ぶ



Gift

フローリストがセレクトした特のお花をたっぷり使って作り出します。お部屋に飾って頂くだけで季節感を演出し、贈られた方が手入れに困らないよう、一つ一つのアレンジメントにこだわった贈り方をご用意しております。



Daily

「日々の暮らしに彩りが欲しい」「そう思っているもの。」「どう組み合わせればいいのか分からない」「花屋を物々しいから苦手だなぁ」「どこに飾りたいのかわからない」そんなお客様の声から生まれたミニブーケ。手頃なサイズと置き場所などそのままで飾っていただけます。
日々の生活に小さな彩りを取り入れてみませんか？

目的で選ぶ



Gift

フローリストがセレクトした特のお花をたっぷり使って作り出します。お部屋に飾って頂くだけで季節感を演出し、贈られた方が手入れに困らないよう、一つ一つのアレンジメントにこだわった贈り方をご用意しております。



Daily

「日々の暮らしに彩りが欲しい」「そう思っているもの。」「どう組み合わせればいいのか分からない」「花屋を物々しいから苦手だなぁ」「どこに飾りたいのかわからない」そんなお客様の声から生まれたミニブーケ。手頃なサイズと置き場所などそのままで飾っていただけます。
日々の生活に小さな彩りを取り入れてみませんか？



Wedding

花嫁様のブーケはもちろん、バージンロードを開くチャペル花、大切なゲストをお迎えするためのウェルカムブーケ。季節のお花やこだわりのアイテムと共に二人らしい結婚式を演出するお手伝いをさせていただきます。



Wedding

花嫁様のブーケはもちろん、バージンロードを開くチャペル花、大切なゲストをお迎えするためのウェルカムブーケ。季節のお花やこだわりのアイテムと共に二人らしい結婚式を演出するお手伝いをさせていただきます。

Access

主要駅からの所要時間

新宿駅 東口より徒歩10分
西武新宿駅 西口より徒歩10分
東武東上線 有楽町線 池袋駅より徒歩10分
池袋駅 西口より徒歩10分
池袋駅 西口より徒歩10分



Google Map→

Access

主要駅からの所要時間

新宿駅 東口より徒歩10分
西武新宿駅 西口より徒歩10分
東武東上線 有楽町線 池袋駅より徒歩10分
池袋駅 西口より徒歩10分
池袋駅 西口より徒歩10分



Google Map→

TOP 季節のイベント 目的で選ぶ Access Online Shop

特定商取引法に基づく表記 プライバシーポリシー



東京都新宿区新宿XX-X ○ビル1F
03-1234-5678

© 2024 Flower Days

TOP 季節のイベント 目的で選ぶ Access Online Shop

特定商取引法に基づく表記 プライバシーポリシー



東京都新宿区新宿XX-X ○ビル1F
03-1234-5678

© 2024 Flower Days

| 5 |

viewport 定義を head 要素内に指定する

『viewport』はブラウザの表示領域 (ユーザーが Web サイトを訪れた際に見ているページの範囲) で、レスポンシブデザインには必須の設定です。

表示領域を指定すると、PC・タブレット・スマホなどユーザーのデバイスの画面サイズに合わせて表示を変えることができます。

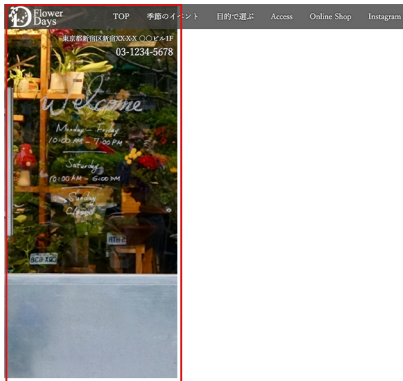
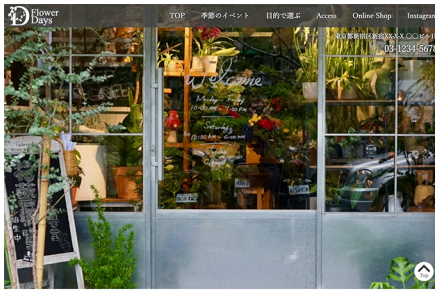
これがないと PC 版の画面を単純に縮小したものが表示されるため、文字が小さすぎて読めず拡大して横移動して縮小してまた拡大して…と閲覧するのに大変な手間がかかり、ページを離脱されてしまいます。

そればかりか Google から評価されず、検索順位が上がりにくくなります。

★印は改行せず、前の行に続けて書く

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

- PC 画面は変化なし
- スマホまたはデベロッパーツールのデバイスモードで、スマホ画面での表示を確認
→ 文字などは大きくなるが、ページが見切れていて横スクロールしないと全体を読めない。
※ PC の画面幅を狭くしただけでは有効になりません。
表示確認はスマホまたはデベロッパーツールのデバイスモードで行ってください。



2024 5/12

Mother's Day

他にはない種類・カラーのお花からメインのお花をお選びいただき、イメージ・ご予算に合わせてスタッフが一つ一つ丁寧に作り上げます。
お花を贈るのとはじめての方でも安心して当店にお任せください。

2024 5/12

Mother's Day

他にはない種類・カラーのお花からメインのお花をお選びいただき、イメージ・ご予算に合わせてスタッフが一つ一つ丁寧に作り上げます。
お花を贈るのとはじめての方でも安心して当店にお任せください。

目的で選ぶ

Daily

「日々の暮らしに彩りが欲しい」そう思っているもの。『どう組み合わせればいいのか分からない』『花瓶を飾っていいかわからない』『どこに飾ったらいいかわからない』そんなお客様の声から生まれたミニブーケ。手持ちのガラスや空き瓶などご自身のまま飾っていただけます。
日々の生活に小さな彩りを取り入れてみませんか？

Gift

フロッグストがモテた！と知ったお花をたっぷり盛り込んで飾り立てますので、お部屋に飾って頂くだけで季節感を演出できる贈り物です。
贈られた方が手入れに困らないよう、一つ一つのアレンジメントに合わせた管理方法カードをお付けしております。

Wedding

花嫁様のブーケはもちろん、バージンロードを飾るチャペル装花、大切なゲストをお迎えするためのウェルカムアラック。季節のお花やこだわりのアイテムと共に二人らしい結婚式を演出するお手伝いをさせていただきます。

目的で選ぶ

Daily

「日々の暮らしに彩りが欲しい」そう思っているもの。『どう組み合わせればいいのか分からない』『花瓶を飾っていいかわからない』『どこに飾ったらいいかわからない』そんなお客様の声から生まれたミニブーケ。手持ちのガラスや空き瓶などご自身のまま飾っていただけます。
日々の生活に小さな彩りを取り入れてみませんか？

Gift

フロッグストがモテた！と知ったお花をたっぷり盛り込んで飾り立てますので、お部屋に飾って頂くだけで季節感を演出できる贈り物です。
贈られた方が手入れに困らないよう、一つ一つのアレンジメントに合わせた管理方法カードをお付けしております。

Wedding

花嫁様のブーケはもちろん、バージンロードを飾るチャペル装花、大切なゲストをお迎えするためのウェルカムアラック。季節のお花やこだわりのアイテムと共に二人らしい結婚式を演出するお手伝いをさせていただきます。

Access

主要駅からの所要時間

駅名	所要時間
新宿駅	東口より徒歩10分
西武新宿駅	正面口徒歩10分
東武東上線	池袋より徒歩10分
池袋駅	西口より徒歩10分
池袋三丁目駅	西口より徒歩10分

Google Map →

Access

主要駅からの所要時間

駅名	所要時間
新宿駅	東口より徒歩10分
西武新宿駅	正面口徒歩10分
東武東上線	池袋より徒歩10分
池袋駅	西口より徒歩10分
池袋三丁目駅	西口より徒歩10分

Google Map →

TOP 季節のイベント 目的で選ぶ Access Online Shop

特定商取引法に基づく表記 プライバシーポリシー

東京都新宿区新宿XX-X-X ○ビル1F
03-1234-5678

© 2024 Flower Days

横幅変動時のコンテンツ幅を設定する

表示領域の横幅を狭くしても横スクロールが発生しないように、コンテンツの幅を設定します。

コンテンツ全体幅の設定には、ボックスモデルの理解が不可欠です。

コンテンツそのものの幅 + padding + border + margin の合計値が画面幅を超えないようにします。

- width には px(固定値) より %(親要素の width を基準として何パーセントか) を指定する
- max-width プロパティ、min-width プロパティを使用する

などすると調整しやすくなります。

今回設定するのは以下の通りです。

※前回記述分

```
main {
  width: 960px;
  ...
}
...

main section.access dl {
  width: 26em;
  ...
}
...

main section.access div {
  width: 500px;
  ...
}
...

footer nav ul.primaryLinks {
  width: 960px;
  ...
}
...

footer nav ul.secondaryLinks{
  width: 960px;
  ...
}
```

```
main {
  max-width: 960px;
  ...
}
...

main section.access dl {
  max-width: 26em;
  ...
}
...

main section.access div {
  max-width: 500px;
  ...
}
...

footer nav ul.primaryLinks {
  max-width: 960px;
  ...
}
...

footer nav ul.secondaryLinks{
  max-width: 960px;
  ...
}
```

※前回記述分

ここにも固定値がはいっていますが、親要素に
display:flex;
を指定していますので変更しなくてもおさまりま
す。

```
header .globalNavi ul{  
  width: 700px;  
  ...  
}
```

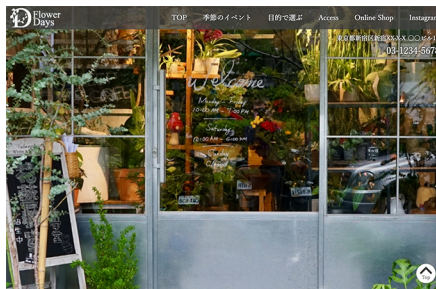
● PC 画面

→ブラウザ画面の横幅を小さくした時、文字の領域と画面幅のバランスが悪く文字が読みにくい。

● スマホまたはデベロッパーツールのデバイスモードで、スマホ画面での表示を確認

→グローバルナビゲーションの文字が改行してしまって読みにくい。

→画面内にはおさまったものの、画面の幅と画像や文字の領域のバランスが悪く文章が読みにくい。



2024

5/12

Sunday

Mother's Day

他にはない種類・カラーのお花からメインのお花をお選びいただき、イメージ・ご予算に合わせてスタッフが一つ一つ丁寧にアレンジいたします。
お花を贈るものがはじめての方でも安心して当店にお任せください。

2024

5/12

Sunday

Mother's Day

他にはない種類・カラーのお花からメインのお花をお選びいただき、イメージ・ご予算に合わせてスタッフが一つ一つ丁寧にアレンジいたします。
お花を贈るものがはじめての方でも安心して当店にお任せください。

目的で選ぶ

Daily

「日々の暮らしに彩りが欲しい」そう思っているものの、「どう組み合わせていいかわからない」「花を飾っていいかわからない」「どこに飾ったらいいかわからない」そんなお客様の声から生まれたミニブーケ。手頃なガラスや空き瓶などからそのままとっていただけます。

日々の生活に小さな彩りを取り入れてみませんか？

Gift

フーリストがセレクトした旬のお花をたっぷり使ってお作り致しますので、お部屋に飾って頂くだけで季節を感じられる贈り物です。贈られた方がお人様に困らないよう、一つ一つのアレンジメントに合わせた管理方法カードをお付けしております。

Wedding

花嫁様のブーケはもちろん、バージンロードを飾るチャペル装花、大目玉装花をお迎えするためのウェルカムフラワー。季節のお花やこだわりのアイテムと共に二人らしい結婚式を演出するお手伝いをさせていただきます。

Access

主要駅からの所要時間

新宿駅 徒歩10分
西武新宿駅 徒歩10分
東武東上線 徒歩10分
有明駅 徒歩10分
有明駅 徒歩10分

Google Map →

目的で選ぶ

Daily

「日々の暮らしに彩りが欲しい」そう思っているものの、「どう組み合わせていいかわからない」「花を飾っていいかわからない」「どこに飾ったらいいかわからない」そんなお客様の声から生まれたミニブーケ。手頃なガラスや空き瓶などからそのままとっていただけます。

日々の生活に小さな彩りを取り入れてみませんか？

Gift

フーリストがセレクトした旬のお花をたっぷり使ってお作り致しますので、お部屋に飾って頂くだけで季節を感じられる贈り物です。贈られた方がお人様に困らないよう、一つ一つのアレンジメントに合わせた管理方法カードをお付けしております。

[TOP](#)
[季節のイベント](#)
[目的で選ぶ](#)
[Access](#)
[Online Shop](#)

[特定商取引法に基づく表記](#)
[プライバシーポリシー](#)

東京都新宿区新宿XX-X-X ○○ビル1F
03-1234-5678

© 2024 Flower Days

Media Querie(メディアクエリ) を 使用してデザインを整える

ブレイクポイントとは

レスポンシブデザインの表示を切り替えるポイントのことです。

新しい端末が発売されるたびに最適と言われる数値が変わります。

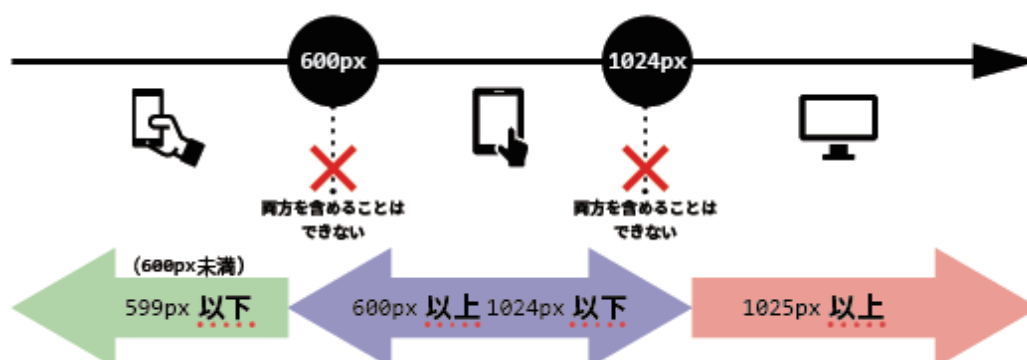
そのほか運営するサイトのターゲット層によってどの端末での表示を優先させるかが違うため、決まったサイズはありません。

2024 年 10 月時点でよく使われるサイズは以下の通りです。

- ・ PC 1025px 〜
- ・ タブレット 600px 〜 1024px
- ・ スマートフォン 〜 599px

ブレイクポイントの考え方

上記のサイズを参考にした場合、図のようになります。『以下』と『未満』に気をつけましょう



場合によって、タブレット (横画面)、スマホ (横画面) などさらに細かく設定する場合があります。もちろんその分 CSS が複雑になっていきます。

今回はブレイクポイントは一つとして、『760px 以下』でデザインが切り替わるようにします。

メディアクエリとは

ユーザーは PC だけではなくスマホやタブレットなどさまざまなデバイスで Web ページを閲覧します。

それぞれのデバイスによって画面サイズや解像度が異なるため、同じスタイルで表示しようとするとうデバイスによっては表示が崩れたり読みにくくなったりすることがあります。

たとえば、PC の画面では画像と文章を横並びにする、スマホなどの小さい画面では横並びレイアウトをやめて縦にする、などデバイスの条件 (画面幅や解像度など) に応じてスタイルシートを切り替えて適用するための技術です。

CSS3 から使えるようになりました。

※※※これは説明用コードです。style.css には記述しないでください。※※※

```
header nav {  
  width: 800px;  
  margin: 0 100px;  
}  
@media screen and (max-width: 760px) {  
  header nav {  
    width: 100%;  
  }  
}
```

このように記述した場合、

「header > nav の幅は 800px、左右のマージンは 100px。ただし画面幅が 760px 以下になったら幅を 100% にしてください。」という意味になります。

改めてすべての指定をやり直しているのではなく、『変更点を上書きする』ものになります。

この場合 760px 以下になったときに幅 100% + 左右のマージンそれぞれ 100px で、200px ぶんはみ出してしまうため、初期値に戻す「margin: 0;」を書かなくてははいけません。

書き方の注意点は、

- {} の数 (緑色の {})。特に、メディアクエリの {} の数。
→条件分けのための {とスタイル指定用の {、それぞれを閉じる}
- 変更したいものはすべて書く。
→初期値に戻したいものの記述を忘れずに書く。
- セレクトアの書き方、点数。(ピンク色の文字)
→同じセレクトアをそのまま書く。

※※※これは説明用コードです。style.css には記述しないでください。※※※

```
header nav {  
  width: 800px;  
  margin: 0 100px;  
}  
@media screen and (max-width: 760px) {  
  nav {  
    width: 100%;  
    margin: 0;  
  }  
}
```

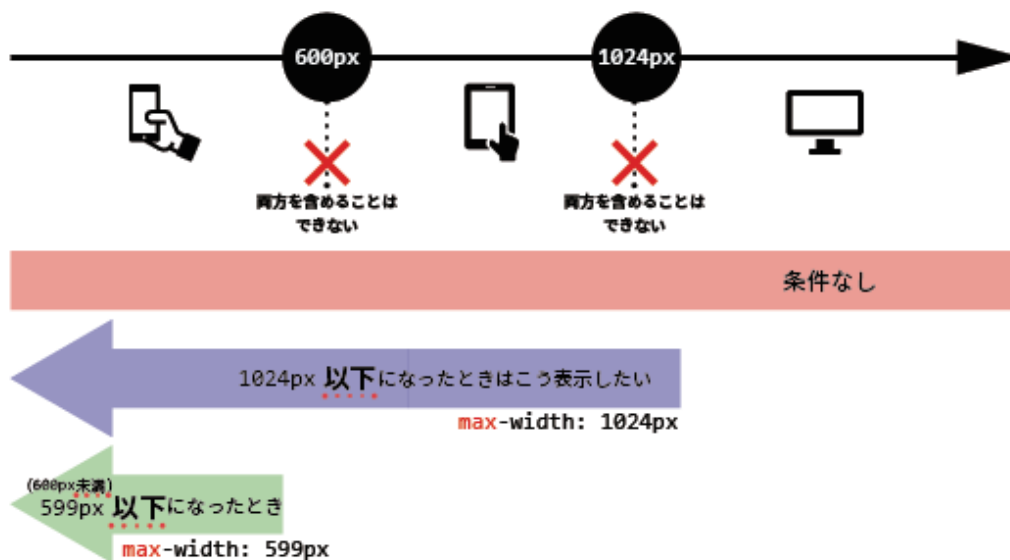


メディアクエリだけセレクトアを省略してしまうと点数で負けてしまい、760px 以下の時のスタイルが適用されませんので気をつけましょう。

そのほか、書く順番も重要です。

レスポンシブデザインに対応する場合、PC用のCSSを書いてからタブレット・スマホ用を追記するやり方と、逆にスマホ用のCSSを書いてからタブレット・PC用を追記するやり方の2通りあります。

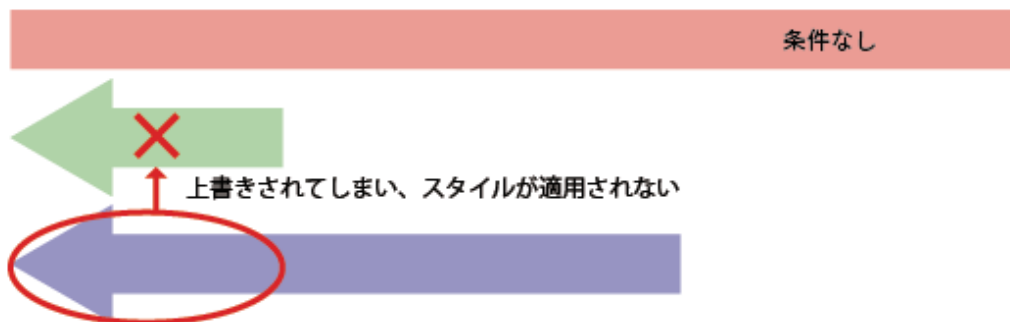
今回はPC用を先に書くやり方で解説しています。



横幅の広い順に、必ず順序を守って記述します。

CSS には、同じセクタ・同じ点数であれば後ろに書いたものが上書きする、というルールがあります。

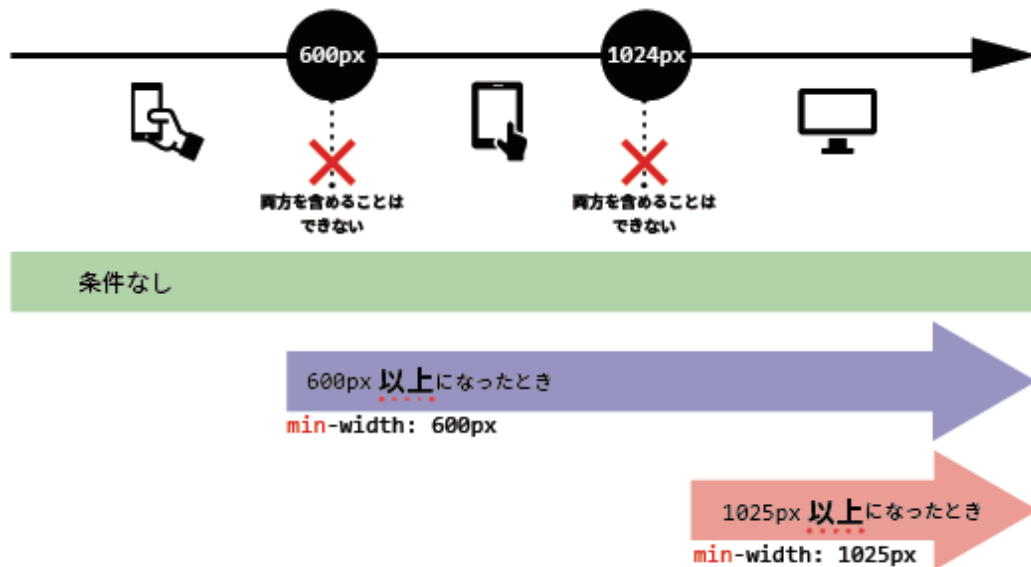
そのため、順番を守らないと思ったようにレイアウトすることはできません。



※※※これは説明用コードです。style.css には記述しないでください。※※※

```
header nav {  
  background-color: pink;  
}  
@media screen and (max-width: 1024px) {  
  header nav {  
    background-color: lightskyblue;  
  }  
}  
@media screen and (max-width: 599px) {  
  header nav {  
    background-color: darkolivegreen;  
  }  
}
```

幅の狭いほうから記述する場合は、このようになります。



※※※これは説明用コードです。style.css には記述しないでください。※※※

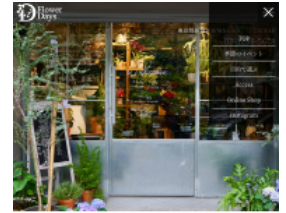
```
header nav {  
  background-color: darkolivegreen;  
}  
@media screen and (min-width: 600px) {  
  header nav {  
    background-color: lightskyblue;  
  }  
}  
@media screen and (min-width: 1025px) {  
  header nav {  
    background-color: pink;  
  }  
}
```

記述する順番と、『max』から『min(最低でも〇〇px 以上)』に変わりますので注意しましょう。

それでは、実際に追記してみましょう。

header

- 背景画像は画面の半分の大きさ
- ナビゲーションはそのままでは見にくいので、ハンバーガーメニューを導入する
※これは後ほど解説します。



```
header {  
  width: 100%;  
  height: 100vh;  
  margin: 0 0 100px;  
  background-image: url("../images/hero.jpg");  
  background-repeat: no-repeat;  
  background-position: center;  
  background-size: cover;  
  color: #fff;  
  text-shadow: 0 0 5px #333, 0 0 5px #333;  
}  
@media screen and (max-width: 760px) {  
  header {  
    height: 50vh;  
  }  
}
```



12 05 2024

main

- コンテンツ幅を 90% に
(左右 5 % の余白を撮った
中央配置ですが、PC 用の
指定で既に左右の auto は
指定済み。下線部分)

★

★印は改行せず、前の行に続けて書く

```
main {
  max-width: 960px;
  padding: 110px 0 0;
  margin: -60px auto 0;
}
@media screen and (max-width:
760px) {
  main {
    width: 90%;
  }
}
```

main > section.event

- ボーダーの内側の余白 :
左右 20px
- 画像と画像の隙間 20px

★

★

★

★

```
main section.event {
  padding: 100px 30px 0;
  border: #333 1px solid;
  margin: 0 0 150px;
}
@media screen and (max-width:
760px) {
  main section.event {
    padding: 100px 20px 0;
  }
  ...
  main section.event div.eventFlex
img {
  width: calc((100% - 60px)/3);
  margin: 0 0 30px;
}
  @media screen and (max-width:
760px) {
    main section.event div.
eventFlex img {
      width: calc((100% - 20px)/2);
    }
  }
}
```

目的で選ぶ

Daily

「日々の暮らしに彩りが欲しい」そう思っているものの、「どう組み合わせればいいのかわからない」「花瓶を持っていないから生けられない」「どこに飾ったらいいのかわからない」そんなお客様の声から生まれたミニブーケ。手持ちのグラスや空き瓶などにそのまま飾っていただけます。日々の生活に小さな彩りを取り入れてみませんか？



Gift

フローリストがセレクトした旬のお花をたっぷり使ってお作り致しますので、お部屋に飾って頂くだけで季節を感じられる贈り物です。贈られた方がお手入れに困らないよう、一つ一つのアレンジメントに合わせた管理方法カードをお付けしております。



Wedding

花嫁様のブーケはもちろん、バージンロードを飾るチャペル装花、大切なゲストをお迎えするためのウェルカムフラワー。季節のお花やこだわりのアイテムと共に二人らしい結婚式を演出するお手伝いをさせていただきます。



main

```
> section.purpose
> div.purposeWrapper
> div.purposeInner
```

- 内容は main の幅のまま (画面の 90%)
- 内側の余白なし
- 文字と写真の間 10px

main

```
> section.purpose
> div.purposeWrapper
> div.purposeImg
```

- 画像は main の幅のまま (画面の 90%)

★印は改行せず、前の行に続けて書く

```
main section.purpose div.
purposeWrapper div.purposeInner {
  width: 50%;
  padding: 20px 30px;
}
@media screen and (max-width:
760px) {
  main section.purpose div.
  ★ purposeWrapper div.purposeInner {
    width: 100%;
    padding: 0;
    margin: 0 0 10px;
  }
}
...
main section.purpose div.
★ purposeWrapper div.purposeImg {
  width: 50%;
}
@media screen and (max-width:
★ 760px) {
  main section.purpose div.
★ purposeWrapper div.purposeImg {
    width: 100%;
  }
}
```

★印は改行せず、前の行に続けて書く

Access

主要駅からの所要時間

新宿駅
東口より徒歩10分
西武新宿駅
正面口歌舞伎町方面より徒歩10分
東新宿駅
A1出口より徒歩6分
新宿三丁目駅
E1出口より徒歩3分



Google Map→

main > section.access
> dl > dt

- main の幅 (画面の 90%)
- 余白なし
- テキストは中央寄せ
- dd と横並びにしない

main > section.access
> dl > dd

- main の幅 (画面の 90%)
- 次の dd との隙間は 20px
- テキストは中央寄せ

```
main section.access dl dt {
  width: 8em;
  margin: 0 1em 0 0;
  text-align: right;
  float: left;
}

★ @media screen and (max-width:
760px) {
  main section.access dl dt {
    width: 100%;
    margin: 0;
    text-align: center;
    float: none;
  }
}

main section.access dl dd {
  width: 17em;
  margin: 0 0 0 9em;
  text-align: left;
}

★ @media screen and (max-width:
760px) {
  main section.access dl dd {
    width: 100%;
    margin: 0 0 20px;
    text-align: center;
  }
}
```

★印は改行せず、前の行に続けて書く



```
footer > nav >  
  ul.primaryLinks > li
```

- ・各項目を親要素の幅いっぱいにして、縦に並べる
- ・詰まっているとタップしにくいので、間隔は1行★分あける

```
footer > nav >  
  ul.secondaryLinks  
    > li
```

- ・各項目を親要素の幅いっぱいにして、縦に並べる★
- ・詰まっているとタップしにくいので、間隔は1行分あける

```
footer nav ul.primaryLinks li {  
  width: 20%;  
  text-align: center;  
}  
  
@media screen and (max-width:  
760px) {  
  footer nav ul.primaryLinks li {  
    width: 100%;  
    line-height: 2;  
  }  
}  
...  
  
footer nav ul.secondaryLinks li {  
  width: 50%;  
  text-align: center;  
}  
  
@media screen and (max-width:  
760px) {  
  footer nav ul.secondaryLinks li {  
    width: 100%;  
    line-height: 2;  
  }  
}
```

Point!

viewport がスマホで効かない原因と対策

● HTML ファイル内の viewport の書き忘れや記述ミス

→表示領域が最適化されずスマホで閲覧しても PC 向けの横幅で表示され、スマホの画面からはみ出してしまいます。

● CSS の記述ミスや未指定のため

→幅に固定値 (px) を指定している、またはコンテンツそのものの幅+マージンの合計値 (box-sizing: border-box; の場合) が画面幅を超えているなどした場合はみだしてしまいます。

px ではなく%に置き換える、max-width プロパティを併記するなどして合計値を 100%以内におさめるようにしましょう。

画面サイズに合わせた表示(ハンバーガーメニューなど)を導入する。

画面サイズに合わせて表示方法をガラッと変えることで操作性を向上させることができます。

例えば、ナビゲーションメニューが横並びのままだと、ひとつひとつのメニューのボタンの領域が限られてしまい、スマートフォンの小さい画面では押しにくくなってしまふなどの不都合があります。

メニュー表示を横並びからハンバーガーメニューなどにする事で狭い画面を有効に使えるようになります。

ハンバーガーメニューとは主にスマートフォンなど横幅の狭い端末でよく使われるナビゲーションメニューの表示形式の一つで、画面の左上隅か右上隅に配置されている『三(または二)』のようなアイコンをタップすると普段は隠れているナビゲーションメニューが表示されます。



ハンバーガーメニューを実装するには様々な方法がありますが、今回は HTML+CSS で実現します。

まずは、HTML に、760px 以下になったら表示するナビゲーションを追記します。

chapter5 > content > hamburger.txt を開き、2 ~ 16 行目を以下の部分にコピー & ペーストします。

同時に、PC 用の nav > ul に、区別するためのクラス名を追記します。


```

<header id="pageTop">
  <nav class="globalNavi">
    <h1><a href="#pageTop"></a></h1>
    <ul class="pcMenu">
      <li><a href="#pageTop">TOP</a></li>
      <li><a href="#mainTop"> 季節のイベント </a></li>
      <li><a href="#purposeTop"> 目的で選ぶ </a></li>
      <li><a href="#accessTop">Access</a></li>
      <li><a href="https://www.rakuten.co.jp/" target="_
★ blank">Online Shop</a></li>
      <li><a href="https://www.instagram.com/" target="_
★ blank">Instagram</a></li>
    </ul>
    <!-- ↓↓↓スマホ用メニュー↓↓↓ -->
    <div class="hamburgerMenu">
      <input type="checkbox" id="menuBtnCheck" value="">
      <label for="menuBtnCheck" class="menuBtn"><span></span></
★ label>
      <div class="menuContent">
        <ul>
          <li><a href="#pageTop">TOP</a></li>
          <li><a href="#mainTop"> 季節のイベント </a></li>
          <li><a href="#purposeTop"> 目的で選ぶ </a></li>
          <li><a href="#accessTop">Access</a></li>
          <li><a href="https://www.rakuten.co.jp/" target="_
★ blank">Online Shop</a></li>
          <li><a href="https://www.instagram.com/" target="_
★ blank">Instagram</a></li>
        </ul>
      </div>
    </div>
  </nav>
  <address> 東京都新宿区新宿 XX-X-X ○○ビル1F<br><em>03-1234-5678</em></
address>
</header>

```

正しい位置にペーストしたら、HTML での記述は終了です。保存しましょう。

この時点では新しいメニューが表示されたぶん、表示が崩れていますが、これから CSS で調整します。

『三』も CSS で表現しますので、まだなくて大丈夫です。



PC ではハンバーガーメニューを隠しスマホでは表示する + 三のボタンを作って表示する

chapter5 > content > hamburger.txtを開き、20～133行目を style.css の header address {} の前にコピー & ペーストします。

```
/*== スマホ用メニュー (ハンバーガーメニュー) ==*/
/* PC 版では非表示、ブレイクポイント (今回は 760px) 以下で表示 */
div.hamburgerMenu {
    display: none;
}
@media screen and (max-width: 760px) {
    div.hamburgerMenu {
        display: block;
    }
}

/* 三ボタンの領域を右上に作成、背景色設定、top/right でボタンの位置を調整、z-index で最前面に */
div.hamburgerMenu .menuBtn {
    width: 60px;
    height: 60px;
    background-color: rgba(0,0,0,0.3);
    display: flex;
    justify-content: center;
    align-items: center;
    position: fixed;
    top: 0;
    right: 0;
    z-index: 90;
    cursor: pointer;
}
```

```

/* ボタンの3本線を作成、線の幅、線の太さ、線の色を設定 */
div.hamburgerMenu .menuBtn span,
div.hamburgerMenu .menuBtn span:before,
div.hamburgerMenu .menuBtn span:after {
    content: '';
    width: 35px;
    height: 3px;
    border-radius: 3px;
    background-color: #fff;
    display: block;
    position: absolute;
}

/* 上の線は上方向に10pxずらす */
div.hamburgerMenu .menuBtn span:before {
    bottom: 10px;
}

/* 下の線は下方向に10pxずらす */
div.hamburgerMenu .menuBtn span:after {
    top: 10px;
}

/* ボタンクリック時：真ん中の線は透明に */
div.hamburgerMenu #menuBtnCheck:checked ~ .menuBtn span {
    background-color: rgba(255, 255, 255, 0);
}

/* ボタンクリック時：上の線を斜めに(／) */
div.hamburgerMenu #menuBtnCheck:checked ~ .menuBtn span::before {
    bottom: 0;
    transform: rotate(45deg);
}

/* ボタンクリック時：下の線を斜めに(＼) */
div.hamburgerMenu #menuBtnCheck:checked ~ .menuBtn span::after {
    top: 0;
    transform: rotate(-45deg);
}

```

```

/* メニューの状態を保持するチェックボックスは非表示 */
div.hamburgerMenu #menuBtnCheck {
    display: none;
}
/* メニューの領域を作成、背景の大きさ、背景色、表示位置、アニメーション（スピードの調整） */
div.hamburgerMenu .menuContent {
    width: 100%;
    height: auto; /* ★100% にするとコンテンツ領域をウィンドウの下までに */
    padding: 0 10px; /* ★メニュー領域の左右の隙間 */
    background-color: rgba(0,0,0,0.7);
    position: fixed;
    top: 0;
    left: 100%;
    z-index: 80;
    transition: all 0.5s;
}

/* メニューの領域を作成、横幅を設定。メニューリストの表示位置を×ボタンの大きさぶん下げる */
div.hamburgerMenu .menuContent ul {
    width: 180px; /* ★100% にすると横幅いっぱい */
    padding: 80px 0 0;
    display: block;
}

/* メニュー項目間の線を追加 */
div.hamburgerMenu .menuContent ul li {
    border-top: solid 1px #fff;
}

/* ☆最初のメニュー項目の上線なくす */
div.hamburgerMenu .menuContent ul li:first-child {
    border-top: none;
}

/* メニューのリンク実体。クリックできる余白を作る、文字色、表示位置 */
div.hamburgerMenu .menuContent ul li a {
    padding: 10px 0px;
    display: block;
    font-size: 16px;
    color: #fff;
}

```

```

    text-align: center;
}

/* ボタンクリック時：メニュー領域の引き出し位置 */
div.hamburgerMenu #menuBtnCheck:checked ~ .menuContent {
    left: calc(100% - 200px);/* ★0 にすると左端まで */
}

```

このままだと PC 用のナビゲーションとハンバーガーメニュー両方が表示されているので、760px 以下になったら「header>nav.globalNavi>ul.pcMenu」を非表示にする CSS を追記します。
この時、クラス名を追記するのも忘れないようにしましょう。

※前回記述分

```

header nav.globalNavi ul {
    ...
    ...
}

```

	← 760px →	
.pc Menu	非表示	表示
.hamburgerMenu	表示	非表示

★

★印は改行せず、前の行に続けて書く

```

header nav.globalNavi ul.pcMenu {
    ...
    ...
}
@media screen and (max-width:
760px) {
    header nav.globalNavi
ul.pcMenu{
        display: none;
    }
}

```

横並びのナビゲーションメニュー⇄ハンバーガーメニューの切り替え表示ができました。
CSS で実装するハンバーガーメニューの場合、メニューをクリックして画面遷移した後もハンバーガーメニューの表示が残ってしまいます。閉じる処理だけは JavaScript や JQuery が必要です。

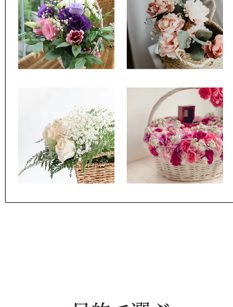
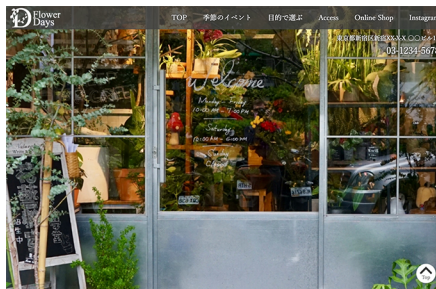
chapter5>content>hamburger.txt を開き、136～141 行目を index.html の</body>の直前にコピー＆ペーストして保存します。

★印は改行せず、前の行に続けて書く

```

<script>
    let a = document.querySelectorAll( 'div.hamburgerMenu a')
    for (let i = 0; i < a.length; i++) {
        a[i].addEventListener( 'click', () => { document.
★ getElementById( 'menuBtnCheck').checked = false; })
    }
</script>

```



これでサンプルサイトは完成です。

ここまでお疲れ様でした。

次は皆さんがデザインしたサイトのコーディングもがんばりましょう♪