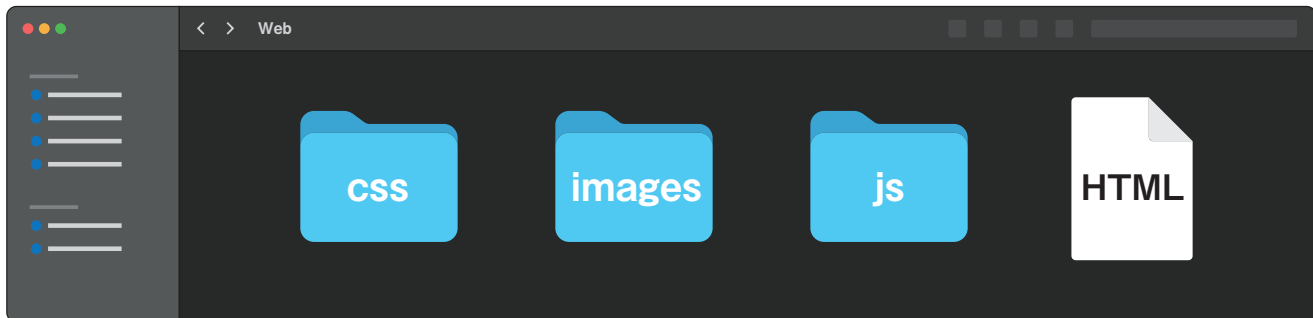


01

フォルダの構成と画像

基本的なフォルダの構成と拡張子
画像の書き出し

● フォルダの作成



web サイトをつくる時は基本的に css フォルダ、images フォルダ、js フォルダを作り

- web ページそのものとなる html ファイル (拡張子 .html)
- web ページのデザインを整えるための css ファイル (拡張子 .css)
- ページ内の画像を表示するための画像ファイル (拡張子 .jpg / .png / .gif)
- 高度なアニメーションを実装できる (拡張子 .js)

をそれぞれのフォルダに入れます。

index.html・・・サイトのトップページとして自動で認識してくれるファイル

style.css・・・主に使われるスタイルシートの名称

common.css・・・ページの共通のコードをかくスタイルシート

画像・・・web ページでは JPEG、PNG、GIF などの画像ファイル形式がよく使われます

形式	拡張子	色数	透過	アニメーション	その他
JPEG	.jpg	フルカラー	×	×	圧縮率を変更できる
PNG	.png	フルカラー	○	×	JPEG よりファイルサイズが大
GIF	.gif	256 色	○	○	

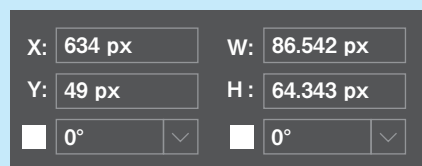
● 画像の書き出し



画像やオブジェクトなど書き出した画像を【アセットの書き出し】ウィンドウのパネルにドラッグ & ドロップ



【変形】ウィンドウの width と height の値を整数にしてから書き出さないと 1px 大きく書き出されるので注意



02

HTML の基礎知識

タグ
HTML の仕組み

● タグの書式と名称



- タグ・・・<で始まり>で終わります。多くのタグには開始タグと終了タグがあります。
- 開始タグ・・・開始タグにはタグ名と属性が含まれることがあります。
- 終了タグ・・・終了タグは<に /、タグ名が続き >で終わります。終了タグに属性は含まれません。
- 属性・・・タグに情報を与えるのが属性です。id、class では css で要素を指名するのに使われます。

- id・・・id 名は 1 つの HTML ドキュメントにつき一度しか使用できません。
- class・・・class 名は同じドキュメント内で何度も使うことができます。1 つの要素に対して複数の class 名をつけることもできます。
- href・・・ファイルの場所や URL を指定することができます。

一つの開始タグから終了タグの間に別のタグが入ってはいけません。タグは開始タグから終了タグまでを挟むように記述しましょう。

- <div><h1> お知らせ </h1></div>
- ✕ <div><h1> お知らせ </div></h1>

コメント文・・・記述してもブラウザに表示されないタグです。メモや目印などに使用すると便利です。コメント文の開始タグは「<!--」で、終了タグは「-->」です。

```
<h1> お知らせ </h1>
<!-- コメント部分はブラウザに表示されません -->
<p> 変更のお知らせ </p>
```

● HTML

HTML では大きさがある要素は左上に寄って表示されます。イメージとしてはブラウザの左側に強力な磁石、上側に弱い磁石があり要素がそれに引っ張られる感じです。

```
<h1> タイトル </h1>
<p> ここに本文ここに本文 <br> ここに本文ここに本文 </p>
```



03

よく使うタグ

タグ

終了タグがないタグ

よく使うタグ

h1~h6 タグ・・・見出しを作成するときに使います。h1 が大見出しで h2~h6 は小さな見出しに使います。

見出し

見出し

見出し

見出し

見出し

見出し

p タグ・・・テキストの段落を作成するときに使います。

br タグ・・・文章を改行するときに使います。

p タグで文章を書いています。

br タグで改行しました。

a タグ・・・リンクを埋め込むときに使います。href の中にリンク先を指定します。

[リンク先 URL](#)

ul、ol、li タグ・・・リストを作成するときに使います。

・項目

・項目

1. 項目

2. 項目

img タグ・・・画像を表示するときに使います。

画像

table タグ・・・表を作成するときに使います。

職業 デザイナー

年齢 25 歳

div タグ・・・要素をグループ化するときに使います。

あいうえお

かきくけこ

span タグ・・・要素をグループ化するときに使います。span は下に積まれずに横並びになります。

あいうえおかきくけこ

section タグ・・・見出しや章の区切りをまとめるときに使います。

header タグ・・・サイトのヘッダー部分を作るときに使います。

nav タグ・・・サイトのメニュー部分を作るときに使います。

footer タグ・・・サイトのフッター部分を作るときに使います。

04

CSS の基礎知識

書式と名称
CSS の仕組み

● CSS の書式と名称

```
h1 {  
  color : red ;  
}
```

セレクター
プロパティ プロパティ値

```
h1 {  
  color : red ;  
}
```

改行
Tab
半角スペース

セレクター・・・HTML ドキュメントから特定の要素を選択します。

プロパティ・・・フォントを指定する、テキストの色を変更するなど css で操作できるスタイルにプロパティが定義されています。

プロパティ値・・・プロパティに設定する値です。例えばテキストの色を変更するプロパティであればプロパティ値には色を指定します。

CSS コメント文・・・CSS として解釈されません。メモや目印などに使用すると便利です。

CSS コメント文は「/*」と「*/」の間に書きます。

```
h1 {  
  color : red ;  
} /* この部分はメモです */
```

● CSS の仕組み

CSS のコードを HTML のコードに反映させるには HTML に css をリンクしなければいけません。リンクするには head タグの中に link タグ書きます。

```
<head>  
  <link rel="stylesheet" href=" ファイル名 ">  
</head>
```

HTML

```
<h1> 見出しテキスト </h1>  
<p> 本文テキスト </p>  
<p> 本文テキスト </p>
```

選択

CSS

```
h1 {  
  color : red ;  
}
```

適用

ブラウザ

```
見出しテキスト  
本文テキスト  
本文テキスト
```

05

テキストに関するCSSプロパティ

CSSプロパティ
プロパティ値

よく使う CSSプロパティ

color・・・テキストの文字色を指定することができます。

プロパティ値

```
color: red;  
color: rgba(255,0,0,0);  
color: #FF0000;
```

テキストの文字色を指定する

font-size・・・フォントの大きさを指定します。

プロパティ値

```
font-size: 50px;
```

フォントの大きさを指定する

プロパティ値

```
font-size: 100px;
```

フォントの大きさを指定する

font-family・・・フォントの種類を指定することができます。

小塚ゴシック

フォントの種類を指定する

游明朝

フォントの種類を指定する

font-weight・・・フォントの太さを指定することができます。

プロパティ値

```
font-weight: bold;
```

フォントの太さを指定する

line-height・・・行間を指定することができます。

プロパティ値

```
line-height: 2;
```

行間を指定する

行間を指定する

letter-spacing・・・文字同士の間隔を指定することができます。

プロパティ値

```
letter-spacing: 2em;
```

文字同士の間隔を指定する

H H H H H H H H H H

text-align・・・文字や画像に対して水平方向の揃え方を指定します。

プロパティ値

```
text-align: left;  
text-align: center;  
text-align: right;
```

左揃え

中央揃え

右揃え

06

要素に関するCSSプロパティ

CSSプロパティ
プロパティ値

よく使う CSSプロパティ

width・・・要素の横幅を指定します。

プロパティ値

```
Abox {width: 100px;}
```

```
Bbox {width: 50%;}
```

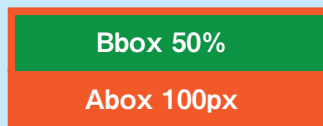


height・・・要素の高さを指定します。

プロパティ値

```
Abox {height: 100px;}
```

```
Bbox {height: 50%;}
```

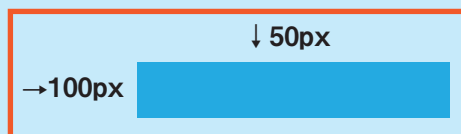


margin・・・要素の「外側」の余白を指定することができます。

プロパティ値

```
margin-top: 50px;
```

```
margin-left: 100px;
```

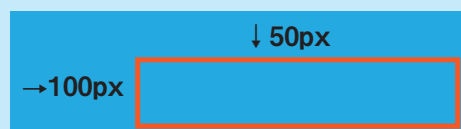


padding・・・要素の「内側」の余白を指定できます。

プロパティ値

```
padding-top: 50px;
```

```
padding-left: 100px;
```



border・・・要素の周りに線をつけることができます。

プロパティ値

```
border-width: 2px;
```

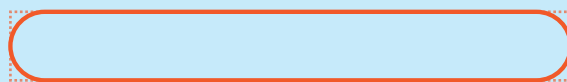
```
border-style: dotted;
```



border-radius・・・要素の4つ角に丸みをつけることができます。

プロパティ値

```
border-radius: 50%;
```



background・・・背景に関するデザインを指定することができます。

プロパティ値

background-color・・・要素の背景色を指定する

background-image・・・要素の背景画像を指定する

background-repeat・・・背景画像の繰り返しについて指定する

background-position・・・背景画像を表示する位置を指定する

07

Dreamweaverの便利機能

ライブプレビュー
値の変更

● ライブプレビュー

Dreamweaver ウィンドウの上中央の「コード 分割 ライブ / デザイン」タブで画面の表示方法を切り替えられます。

コード・・・コードのみの表示でデザイン画面が表示されません
分割・・・デザイン画面とコードが分割で表示されます。分割画面でライブに切り替えるとブラウザにいちいち切り替えなくてもリアルタイムでコードが反映されます

ライブ画面が更新されない時は一度デザイン画面にすると反映されます。



● 値の変更

CSS デザイナーウィンドウのプロパティパネルで数値を変更するとき、数値上で左右にドラッグすると簡単に変更できます。ライブ画面を見ながらドラッグすると調整しやすいです。

width : 50%

● コメント文

コメント文を書きたい時は Dreamweaver ウィンドウ左の「コメントの適用」をクリックするとコメント文を適用できます。



HMTL コメントの適用・・・HTML コメント
/** コメントの適用・・・css コメント
// コメントの適用・・・JavaScript コメント

● ソースの整理

ソースを Dreamweaver ウィンドウ左の「ソースコードのフォーマット」をクリックするとソースコードがインデントや改行されみやすいコードになります。



ソースフォーマットの適用

● リアルタイムプレビュー

Dreamweaver ウィンドウ右下の「プレビュー」でブラウザでのプレビューを見ることができます。ファイル→リアルタイムプレビューよりもクリック数を短縮できます。



▼ブラウザでプレビュー

[一覧を編集](#)

Google Chrome

Safari

● ソースの整理

タグをタグで囲みたい時は囲みたいタグをドラッグで選択した状態で、command+T を押すと囲みたいタグを選択できます。

タグで囲む : <a>

08

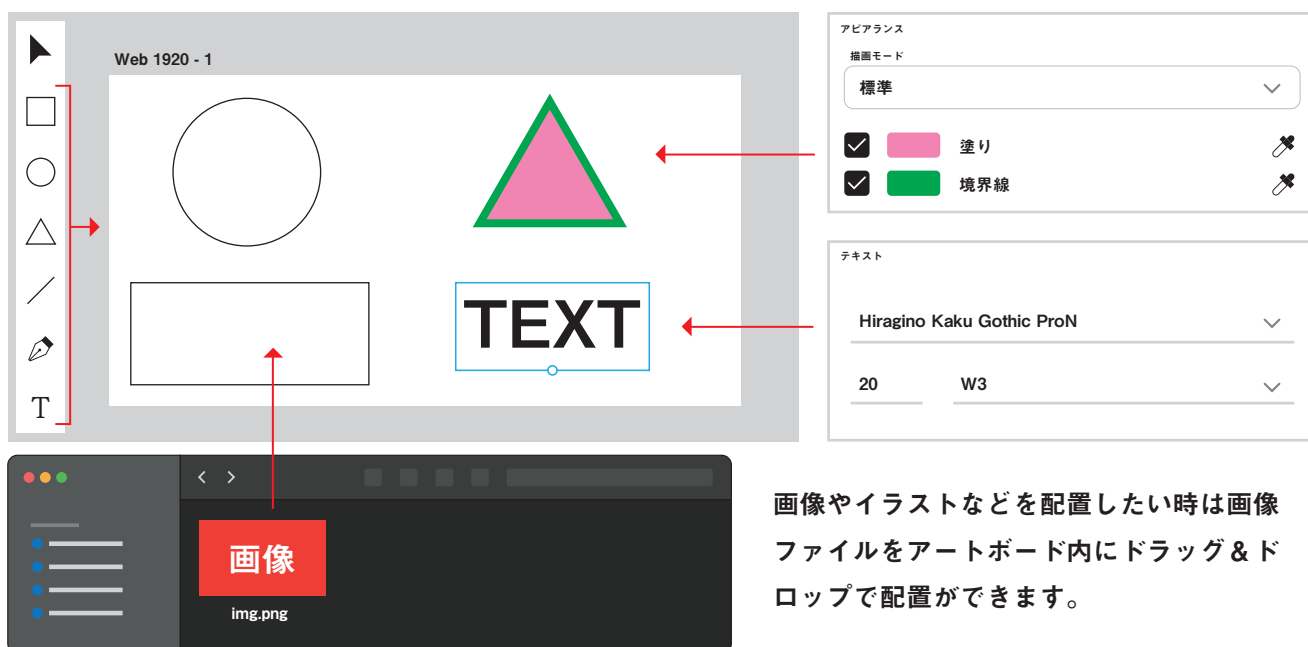
XD の使い方

基本的な XD の使い方
プロトタイプ

● 基本的な XD の使い方

Web サイトやモバイルアプリなどの UI/UX デザイン、プロトタイプ作成、共有までを行える XD の使い方です。Illustrator の基本的な使い方がわかっていることが前提です。

まずは、新規作成でアートボードを作成し画面左側のツールバーから図形を選んでドラッグで作成しましょう。配置した図形の色や形、線の太さなどは、画面右側にあるプロパティインスペクターから変更できます。



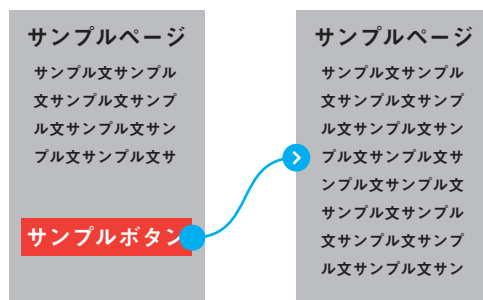
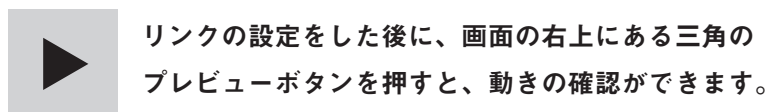
同じ形のオブジェクトを等間隔に配置したい場合にはリピートグリッドを使います。

まず、オブジェクトを選択し、プロパティインスペクターにある[リピートグリッド]をクリックします。オブジェクトを囲む緑の枠線を、ドラッグすると等間隔に複製されます。オブジェクト間の赤い長方形をドラッグすると間隔を調整することができます。



● XD 独自の機能

XD ではボタンを押したときにどのページに移動するかを直感的に設定することができます。画面左上からプロトタイプモードを選択し、リンクしたいボタンを選択します。青い丸ボタンをドラッグしてリンク先のアートボードと繋げます。



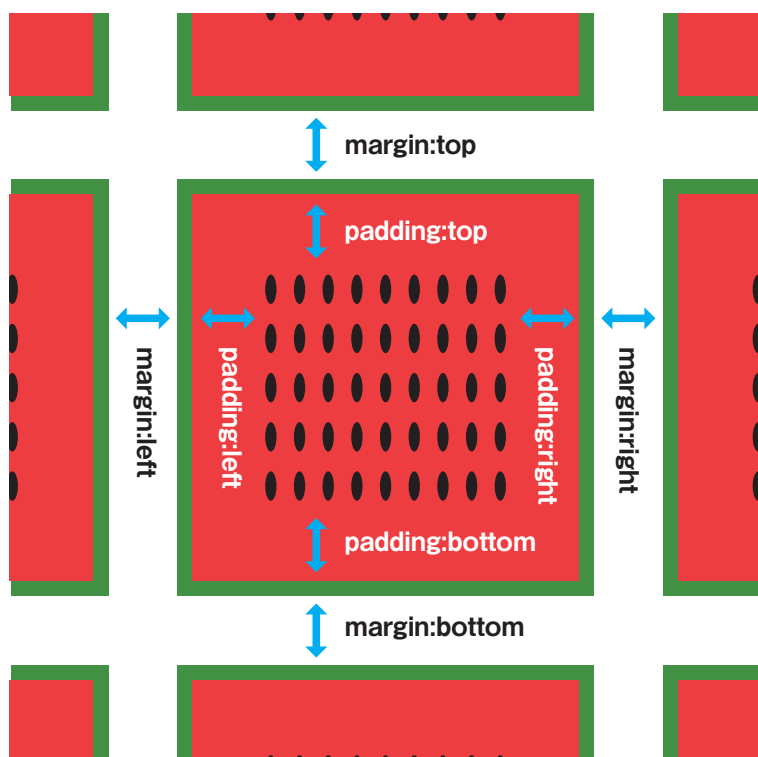
09

margin と padding

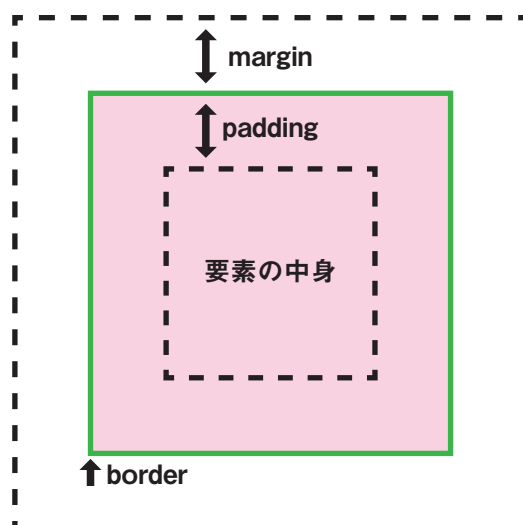
余白の考え方
padding の使い方

margin と padding の余白

marginは要素の外側の余白、paddingは要素の内側の余白です。



例えば半分に切られた四角いスイカが並んでいて周りのスイカと距離をあげたい時はmarginを指定します。スイカの中のタネを内側に向かってよせたい時はpaddingを指定します。



margin と padding のショートハンド

上下左右の余白に別々の値を指定したいときに「margin-top:〇〇px; margin-right:〇〇px…」と書くともみにくいコードになってしまいます。そんなときは、以下のように上下左右の値をまとめて指定します。

CSS

```
{margin: 12px 3px 6px 9px}
```

上のmargin:12px、右のmargin:3px、下のmargin:6px、左のmargin:9pxと時計回りに値を指定します。

p タグの padding

pタグにpaddingと背景色、ボーダーをつけてみます。

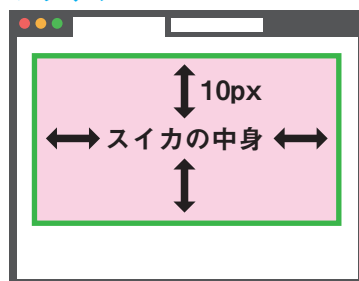
HTML

```
<p>  
スイカの中身  
</p>
```

CSS

```
p {  
padding: 10px;  
background: pink;  
border: sokid 2px green;  
}
```

ブラウザ



文字や画像にpaddingをつけるとpaddingで作られる余白にも背景色がつきその外側にボーダーが付きまます。

10

float プロパティ

横並びにする
回り込み

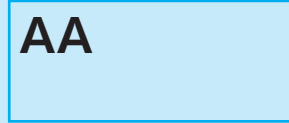
● float プロパティ

css プロパティ「float」は要素を横並びにすることができます。

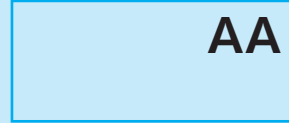
プロパティ値

float: left;
float: right;
float: none;

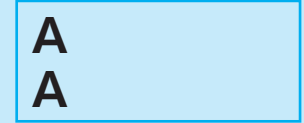
left



right



none

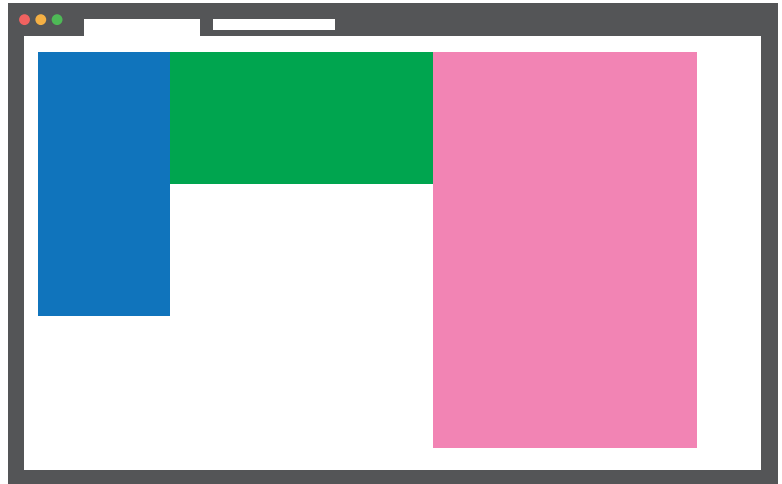


実際に3つの box を左に寄せるコードを書いてみます。

HTML

```
<div class="blue"></div>
<div class="green"></div>
<div class="pink"></div>
```

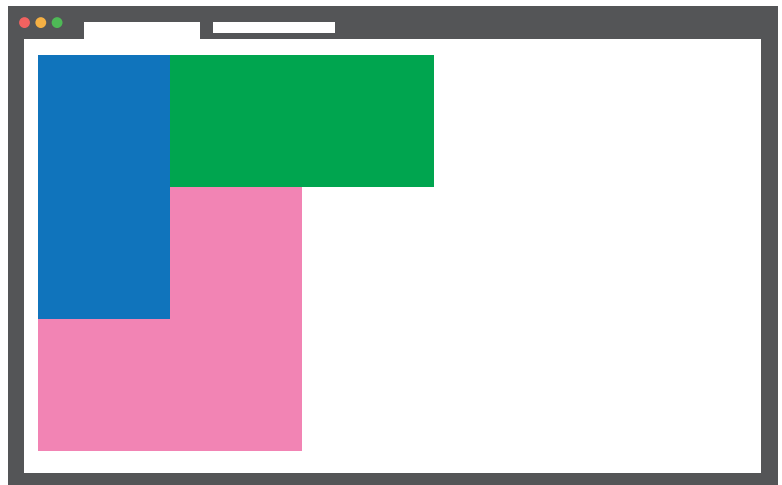
ブラウザ



CSS

```
.blue{
  background-color: blue;
  width: 100px;
  height: 200px;
  float: left;
}
.green{
  background-color: green;
  width: 200px;
  height: 100px;
  float: left;
}
.pink{
  background-color: pink;
  width: 200px;
  height: 300px;
  float: left;
}
```

では pink の float を取るとどうなるでしょうか。



ピンクの box は青の box の下にくるかと思ったらおかしな位置にきてしまいました。なぜかという float プロパティでは要素が浮いている状態になります。なので青と緑の box の下にピンクの box が入り込んでしまいます。

こうらないようにするには pink に clear プロパティをつけます。値は both です。

```
clear: both;
```

11

overflow プロパティ

overflow プロパティ
回り込みの解除

● overflow プロパティ

css プロパティ「overflow」は要素のボックスからはみ出た部分をどう扱うかを指定します。

HTML

```
<p class="overf">  
あいうえおかきくけこさしすせそ  
</p>
```

CSS

```
.overf {  
  width: 100px;  
  height: 100px;  
  overflow: visible;  
  background: pink;  
}
```

プロパティ値

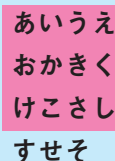
visible・・・はみ出た部分が、はみ出たままの状態が表示される

hidden・・・はみ出た部分が隠れる

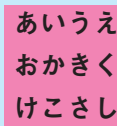
scroll・・・はみ出た部分が隠れてスクロールできる状態に

auto・・・ブラウザにより表示が変わる。ほとんどの場合 scroll を指定した時と同じようにスクロールバーが表示される

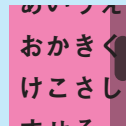
visible

A pink rectangular box containing the text "あいうえおかきくけこさしすせそ". The text is wrapped onto multiple lines, and the bottom portion of the text is visible outside the bottom edge of the box.

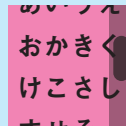
hidden

A pink rectangular box containing the text "あいうえおかきくけこさしすせそ". The text is wrapped onto multiple lines, but the bottom portion of the text is completely hidden behind the bottom edge of the box.

scroll

A pink rectangular box containing the text "あいうえおかきくけこさしすせそ". The text is wrapped onto multiple lines, and the bottom portion is hidden. A vertical scrollbar is visible on the right side of the box, indicating that the content can be scrolled.

auto

A pink rectangular box containing the text "あいうえおかきくけこさしすせそ". The text is wrapped onto multiple lines, and the bottom portion is hidden. A vertical scrollbar is visible on the right side of the box, indicating that the content can be scrolled.

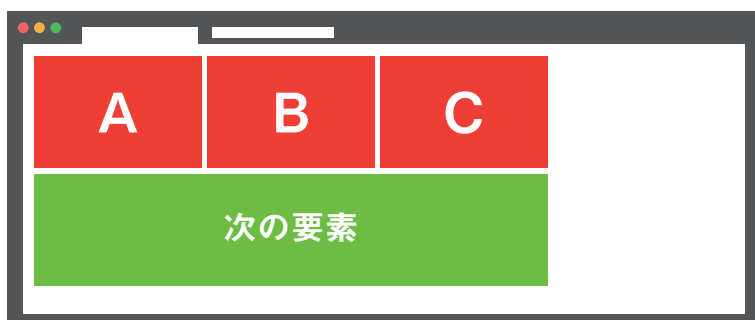
● 回り込みの解除

float の回り込みを解除するには overflow: hidden を親要素に追記するだけです。理由としては overflow プロパティでは float で浮いている子要素の高さを認識してくれるからです。

HTML

```
<div class="outbox">  
  <div class="inbox">A</div>  
  <div class="inbox">B</div>  
  <div class="inbox">C</div>  
</div>  
<div class="nextbox"> 次の要素 </div>
```

ブラウザ



CSS

```
.outbox {  
  overflow: hidden;  
}  
.inbox {  
  background: #FF0000;  
  width: 150px;  
  height: 100px;  
  float: left;  
}  
.nextbox {  
  background: #00FF00;  
  width: 450px;  
  height: 100px;  
}
```

12

position プロパティ

position の基本
position の使い方

● position プロパティ

positionとは要素の位置を決めるためのプロパティです。基本的にtop、bottom、left、rightプロパティとセットで使います。

プロパティ値

static・・・初期値

relative・・・現在の位置を基準に相対的な位置を指定する

absolute・・・親要素を基準に絶対的な位置を指定する

fixed・・・要素を画面のきまった位置に固定する

プロパティ値

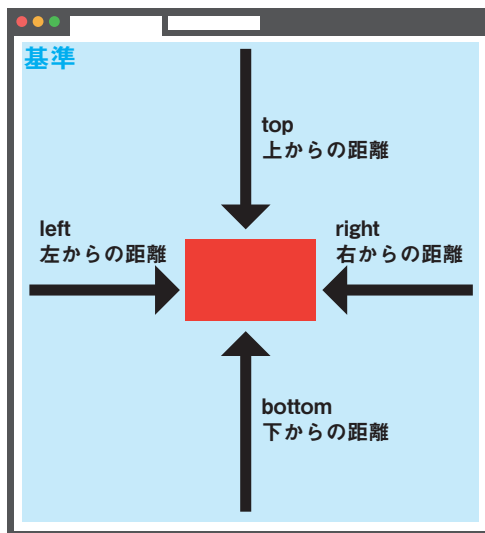
top・・・基準の上からの距離

bottom・・・基準の下からの距離

left・・・基準の左からの距離

right・・・基準の右からの距離

ブラウザ



● static

relativeは初期値です。z-index: 0の状態になっており、変えることができません。

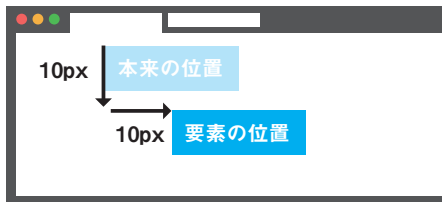
● relative

relativeは現在の表示位置から相対的に要素の位置を動かしたいときに使います。要素をposition: relativeにしたうえでtopとleftをそれぞれに10pxに指定しました。

position:relativeとしても、topやbottomを指定しなければstaticと同じ位置に表示されます。

relativeではstaticと違いz-indexを指定できます。

ブラウザ



● absolute

absoluteを使うと親要素を基準に、絶対的な位置を指定できます。「絶対的な位置」というのは、要素がどんな形であろうと、間に他の要素や余白が入っていようと、その位置に配置してくれるということです。

absoluteを使って位置調整するときは、親要素にrelativeを指定して使います。

例えば、親要素(画像)の左上から下に10px、右に10pxの位置に要素を配置してみます。

ブラウザ



● fixed

fixedは画面の決まった位置に要素を固定させたいときに使います。スクロールしても位置を変えたくないナビゲーションやハンバーガーメニューを作るときに便利です。

位置の基準は親要素ではなく「ウィンドウ全体」です。

ブラウザ



13

display プロパティ

display プロパティ

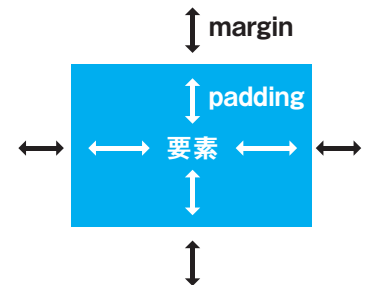
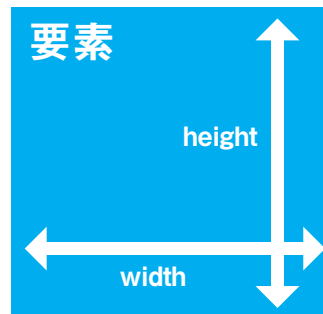
block、inline、inline-block、none

● block

div タグや p タグなどの要素は、はじめから値が block になっています。block では要素の前後に改行が入り縦並びになります。幅と高さの指定ができ margin と padding の設定ができます。text-align と vertical-align の設定はできません。



要素に改行が入り縦並びになる



● inline

span タグや a タグ、img タグなどの要素は、はじめから値が inline になっています。inline では要素が横並びになります。幅と高さの指定ができず、左右の margin と padding の設定はできますが、上下の margin と padding の設定はできません。text-align と vertical-align の設定はできます。



要素が横並びになる



幅と高さは指定できない
要素の容量によってきまる



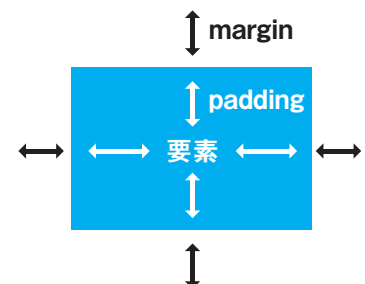
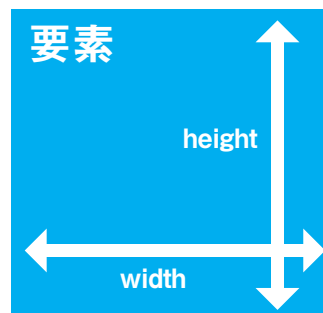
横方向のみ margin と padding
が設定できる

● inline-block

inline-block は inline と block の間を取ったようなものです。inline-block では要素が横並びになります。幅と高さの指定ができ margin と padding の設定ができます。text-align と vertical-align の設定もできます。



要素が横並びになる



● none

none はブラウザ上で非表示になります。none はレスポンシブデザインを作るときによく使われます。「パソコンで見ると表示されるけどスマホで見たときには非表示になる」といったときに使います。

14

Flexbox 使い方

flex の基本
flex の使い方

● display: flex

Flexbox は細かい所を自動で調整してくれたり、簡単にレイアウトを作成することができます。
Flexbox を使用する場合には、親要素となるコンテナを作成し「display:flex」を指定し、その中に子要素となるアイテムを作ります。コンテナに使用できるプロパティは下記のとおりです。

flex-direction・・・アイテムの並び順を指定する
flex-wrap・・・アイテムの折り返しを指定する
justify-content・・・アイテムの水平方向の位置を指定する
align-items・・・アイテムの垂直方向の位置を指定する
align-content・・・アイテムの行の垂直方向の位置を指定する



● flex-direction

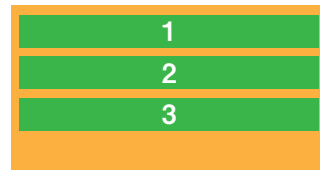
row(デフォルト)



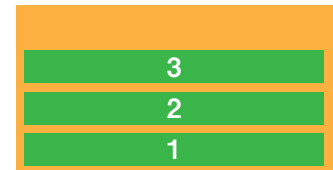
row-reverse



column



column-reverse



● flex-wrap

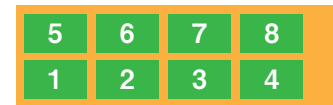
nowrap(デフォルト)



wrap



wrap-reverse



● justify-content

flex-start(デフォルト)



flex-end



center



space-between



space-around



● align-items

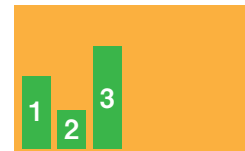
stretch(デフォルト)



flex-start



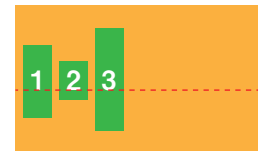
flex-end



center

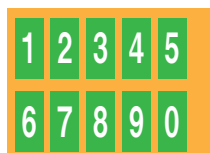


baseline

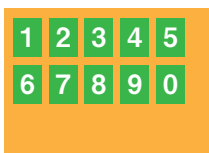


● align-content

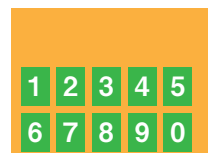
stretch(デフォルト)



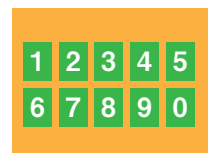
flex-start



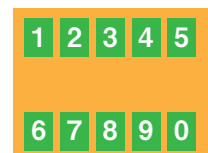
flex-end



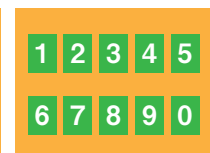
center



space-between



space-around



15

before と after

before とは
after とは

● 疑似要素

CSS の `::after` と `::before` は、疑似要素と呼ばれます。疑似要素を使うと HTML には書かれていない要素を CSS で作ることができます。

`::before`・・・要素の直前に内容を追加できる。

`::after`・・・要素の直後に内容を追加できる。

```
h1::before{ content:" 内容 "}
```

```
h1::after{ content:" 内容 "}
```

↓ ↓
<h1> タイトルテキスト </h1>

● スタイル

HTML

```
<h1 class="text"> 日本語のタイトル </h1>
```

CSS

```
.text {  
  text-align: center;  
}  
.text::before{  
  content: 'Title English';  
  color: red;  
  font-size: 2em;  
  display: block;  
}
```

ブラウザ

Title English

日本語のタイトル

`content:"`の中に追加したい文字列を入れます。文字を赤色にし、フォントサイズを「日本語のタイトル」の2倍にして、`display: block`で改行しています。疑似要素は普通の要素と同じようにスタイルの指定ができます。また、`before/after`を `position: absolute`で絶対配置にすることで、自分の好きな位置に配置できるようになります。

注意点としてコロン (:) の数です。疑似要素には、コロンが1つ `::before` の書き方と、コロンが2つ `::before` の書き方があります。コロン2つを使えるのは、CSS3に対応しているブラウザだけです。コロン1つならCSS3に対応していないブラウザでも表示できます。

16

要素の中央揃え

左右中央揃え
上下左右中央揃え

● 左右中央揃え

文字や画像を左右中央揃えにします。

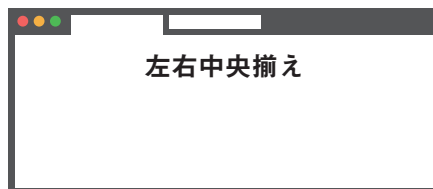
HTML

```
<div>
  <p> 左右中央揃え </p>
</div>
```

CSS

```
div {
  text-align: center;
}
```

ブラウザ



要素自体を左右中央揃えにします。このほうほうでは親要素がインライン(aタグ、spanタグ)要素には効きません。その場合インライン要素にdisplay:blockを指定し幅を持つようにします。

HTML

```
<div>
  左右中央揃え
</div>
```

CSS

```
div {
  margin: 0 auto;
  width: 100px;
}
```

ブラウザ



● 上下左右中央揃え

要素を上下左右中央揃えにします。positionで要素を上下左右中央揃えにします。この方法では親要素に高さ、子要素に幅を指定します。

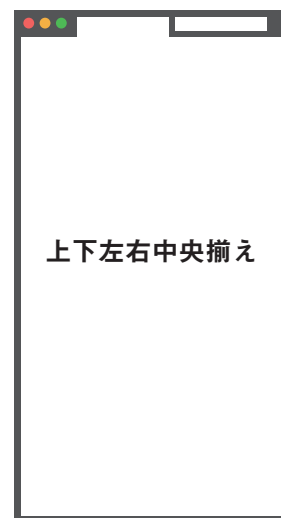
HTML

```
<div>
  <p>BOX</p>
</div>
```

CSS

```
div {
  position: relative;
}
p {
  position: absolute;
  top: 50%;
  left: 50%;
  -webkit-transform : translate(-50%,-50%);
  transform : translate(-50%,-50%);
}
```

ブラウザ



● 注意点

中央揃えをする時はその要素に応じた方法で書かなければいけません。

例えば、text-align: centerが使えない時は要素がブロック要素になっていないかを確認し、margin: 0 autoが使えない時はインライン要素になっていないかを確認します。

position: absoluteができないときには、親要素にrelativeが指定されているかを確認します。

幅や高さを指定しないとイケない時もあるので注意が必要です。

17

ナビゲーション

ナビゲーションを作る
ホバーアニメーション

● ナビゲーションを作る

まずは HTML と CSS で簡単なナビゲーションを作っていきます。

HTML の `` の「#」のところにはクリックしたときにリンクするページを記述します。

CSS では `text-align` で中央に寄せ、`li` タグを `inline-block` で横並びにします。`a` タグを `block` にして `padding` を上下左右設定できるようにします。

HTML

```
<nav>
  <ul>
    <li><a href="#"> トップ </a></li>
    <li><a href="#"> 紹介 </a></li>
    <li><a href="#"> ニュース </a></li>
    <li><a href="#"> SNS </a></li>
    <li><a href="#"> お問い合わせ </a></li>
  </ul>
</nav>
```

CSS

```
ul{
  background: red;
  text-align: center;
}
li{
  display: inline-block;
}
a{
  display: block;
  padding: 10px;
  color: white;
  text-decoration: none;
}
```

ブラウザ



● ホバーアニメーション

次にマウスカーソルをボタンと重なるとボタンの色が変わる CSS を記述します。

セレクター `:hover` { プロパティ: プロパティ値; } で設定できます。今回は背景色が緑になるように設定します。

CSS

```
li:hover{
  background: green;
}
```

ブラウザ



18

ライトボックス

ライトボックスの準備
ライトボックスの使い方

● ライトボックスの準備

ブラウザ



ブラウザ



画像をポップアップで拡大表示させることができるライトボックスという仕組みを作っていきます。まず、URL「lokeshdhakar.com/projects/lightbox2/」にアクセスして Lightbox をダウンロードします。いろいろなファイルが入っていますが、使用するのは dist フォルダだけです。dist 内の「lightbox.css」と「lightbox.js」、「images フォルダ」をサイトフォルダに移動します。



● ライトボックスの使い方

ライトボックスの準備ができたら CSS ファイルを head タグ内にリンクし、JavaScript ファイルを body タグ内の一番最後にリンクしてください。

次に、ポップアップで表示させたい画像を a タグで囲み、href に画像へのパスと、data-lightbox 属性になんでもいいので値を付けてください。data-title 属性には画像をポップアップ表示したときのタイトルが設定できます。タイトルを設定しない場合は属性を削除して大丈夫です。

HTML

```
<a href=" ポップアップ画像のリンク " data-lightbox="name" data-title=" 画像のタイトル ">

</a>
```

ポップアップした状態でスライドさせながら見られるようにするには共通の data-lightbox の値を設定します。

HTML

```
<a href=" 画像のリンク " data-lightbox="some"></a>
<a href=" 画像のリンク " data-lightbox="some"></a>
<a href=" 画像のリンク " data-lightbox="some"></a>
```

19

ハンバーガーメニュー①

ハンバーガーボタンを作る
ハンバーガーボタンのカスタマイズ

● ハンバーガーボタンを作る

まず、ハンバーガーボタンを作ります。赤字部分は自分好みにカスタマイズしてください。

赤文字部分以外を変えるとデザインが崩れることがあります。

メニュー部分はハンバーガーメニュー②で作ります。

HTML

```
<div class="ham">
  <span class="ham_line ham_line1"></span>
  <span class="ham_line ham_line2"></span>
  <span class="ham_line ham_line3"></span>
</div>
```

ブラウザ



CSS

```
.ham {
  position: relative;
  top: 0px;
  left: 0px;
  width: 50px;
  height: 50px;
  background-color: black;
  z-index: 10;
}
.ham_line {
  position: absolute;
  left: 10px;
  width: 30px;
  height: 3px;
  background-color: white;
  transition: all 0.3s;
}
.ham_line1 {
  top: 12px;
}
.ham_line2 {
  top: 24px;
}
```

```
.ham_line3 {
  bottom: 12px;
}
.close .ham_line1 {
  transform: rotate(45deg);
  top: 24px;
}
.close .ham_line2 {
  width: 0px;
}
.close .ham_line3 {
  transform: rotate(-45deg);
  top: 24px;
}
```

ハンバーガーボタンの幅や高さを変えたら中の線の幅と高さ、太さ、空きを調整してデザインを整えてください。

ハンバーガーボタンの場所を固定したい場合は `position: relative;` を `position: fixed;` に変更してください。また、ボタンの場所を変えたい場合は `.ham` の `top` と `left` で指定してください。

ハンバーガーメニューを作る
ハンバーガーメニューのカスタマイズ

● ハンバーガーメニュー

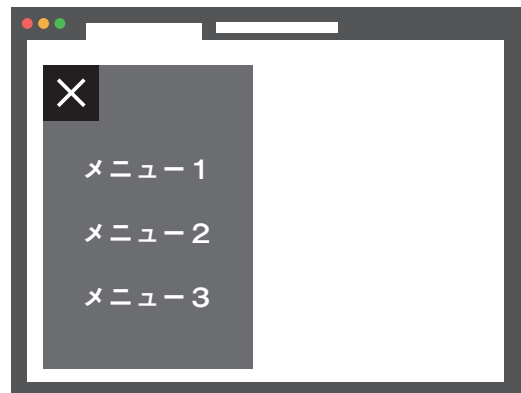
ハンバーガーメニューのメニュー部分を作ります。赤字部分は自分好みにカスタマイズしてください。
赤文字部分以外を変えるとデザインが崩れることがあります。

JQuery ファイルをダウンロードもしくは CDN を、リンクしてください。

HTML

```
<div class="menu">
  <ul>
    <li><a href="#">メニュー1</a></li>
    <li><a href="#">メニュー2</a></li>
    <li><a href="#">メニュー3</a></li>
  </ul>
</div>
```

ブラウザ



CSS

```
.menu{
  position: fixed;
  width: 30%;      メニュー幅
  height: 100vh;  メニュー高さ
  top: 0px;       上の空き
  left: -400px;   左の空き
  background-color: grey;  メニュー色
  transition: all 0.3s;
}

.noscroll{
  overflow: hidden;
}

.menu.show{
  top: 0px;       上の空き
  left: 0px;      左空き
}

.menu ul li a{
  display: block;
  color: white;   文字色
  margin-top: 100px;  上の空き
  margin-left: 30px;  左の空き
}
```

JavaScript

```
$(function () {
  $('ham').click(function () {
    $('ham').toggleClass('close');
    $('menu').toggleClass('show');
    $('body').toggleClass('noscroll');
  })
});
```

.menu.show は半角スペースを入れずに入力してください。

メニューを上からアニメーションで出てくるようにするには .menu の top を負の数、left を 0 もしくは正の数にしてください。

メニューを右からアニメーションで出てくるようにするには .menu と .menu.show の left を right に変更してください。

JavaScript にエラーが出た場合はコードの確認と、ソースフォーマットを適用してコードを整えてください。

● slick の準備

slick というプラグインを使ってスライドを作ります。URL 「kenwheeler.github.io/slick/」からファイルをダウンロードします。「.js」ファイルを作成し「slick.css」「jquery.min.js」「slick.min.js」と作った JavaScript ファイルを head タグの中にリンクします。

赤文字部分は自分好みにカスタマイズしてください。赤文字部分以外を変えるとデザインが崩れることがあります。

HTML

```
<div class="slider">
  <ul id="js-slider">
    <li></li>
    <li></li>
    <li></li>
  </ul>
  <div class="dots"></div>
</div>
```

ブラウザ



JavaScript

```
$(function () {
  $('#js-slider').slick({
    arrows: true, 〰️ 矢印を表示するか (表示しない場合は false)
    dots: true, 〰️ ドットを表示するか (表示しない場合は false)
    appendDots: $('.dots'),
    autoplay: true, 〰️ スライドを自動で切り替えるか (切り替えない場合は false)
    autoplaySpeed: 1000, 〰️ スライドが自動で切り替わる速度 (ミリ秒)
    slidesToShow: 1, 〰️ 何枚の画像を同時に表示するか
    slidesToScroll: 1, 〰️ 何枚の画像を同時にスクロールするか
    responsive: [{ 〰️ レスポンシブでの設定
      breakpoint: 600, 〰️ 何 px で設定を適用するか
      settings: {
        slidesToShow: 1, 〰️ ここにレスポンシブの設定を記述する
      }
    }
  ]
});
});
```

● スライドの装飾

CSS

```

.slider{
  width: 50%; ———— スライドの幅
}
.slick-slide {
  height: 400px; ———— スライドの高さ
  position: relative;
}
.slick-slide img {
  display: block;
  width: 100%;
  height: 100%;
  object-fit: cover;
}
.slick-arrow {
  width: 60px; ———— 矢印背景の幅
  height: 60px; ———— 矢印背景の高さ
  border-radius: 50%; - 角を丸める
  background: white; - 背景の色
  border: none;
  outline: none;
  font-size: 0;
  position: absolute;
  top: 170px; ———— 矢印背景の位置
  z-index: 1;
}
.slick-arrow::before {
  content: "";
  display: block;
  width: 12px; ———— 矢印の幅
  height: 12px; ———— 矢印の高さ
  border: solid black; - 矢印の色
  position: absolute;
  top: 24px; ———— 矢印上の空き
  transform: rotate(45deg);
}

.slick-next {
  right: 20px; ———— 右矢印背景右の空き
}
.slick-prev {
  left: 20px; ———— 左矢印背景左の空き
}
.slick-next::before {
  border-width: 3px 3px 0 0; - 右矢印の太さ
  left: 20px; ———— 右矢印の左の空き
}
.slick-prev::before {
  border-width: 0 0 3px 3px; - 左矢印の太さ
  right: 20px; ———— 左矢印右の空き
}
.dots .slick-dots {
  text-align: center;
}
.dots .slick-dots li {
  display: inline-block;
}
.dots .slick-dots button {
  display: block;
  width: 10px; ———— ドットの幅
  height: 10px; ———— ドットの高さ
  margin: 6px; ———— ドットの空き
  font-size: 0;
  padding: 0;
  border: 0;
  outline: none;
  background: gray; ———— ドットの色
  border-radius: 50%; ———— 角を丸くする
}
.dots .slick-dots .slick-active button {
  background: red; ———— 表示中のドットの色
}

```

● アコーディオンメニュー

HTML

```
<p class="acc_menu"> メニュー </p>
<div class="acc_con"> 内容 </div>
```

CSS

```
.acc_con {
  display: none;
}
.acc_menu {
  padding: 15px;
  color: white;
  font-size: 20px;
  background: gray;
  position: relative;
}
.acc_menu::before {
  content: "+";
  position: absolute;
  right: 20px;
}
.acc_menu.active::before {
  content: "-";
}
```

JavaScript

```
$(function () {
  $(' .acc_menu').click(function () {
    $(' .acc_con').slideUp();
    $(' .acc_menu').removeClass('active');
    $(this).toggleClass('active');
    $(this).next(' .acc_con').slideToggle();
  });
});
```

メニューをそれぞれ別々に開きたい時は
この部分を削除

ブラウザ



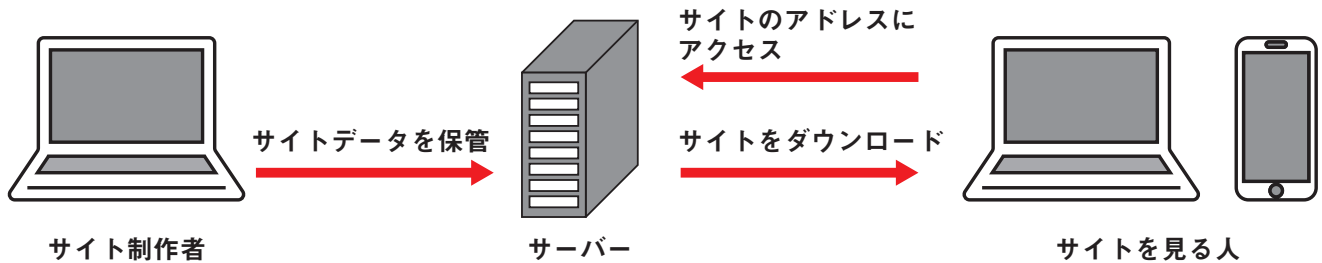
メニューの数を増やしたい時は
HTML 部分を必要な分だけコピー
& ペーストしてください。

下記の JavaScript では一つのメ
ニューしか表示されない(別のメ
ニューは閉じるようになってい
る)のでメニューを別々に開きた
い時は 3 行目と 4 行目を削除して
ください。

JQuery ファイルをダウンロード
もしくは CDN を、リンクしてく
ださい。

サーバー
アップロード

● サーバーの仕組み

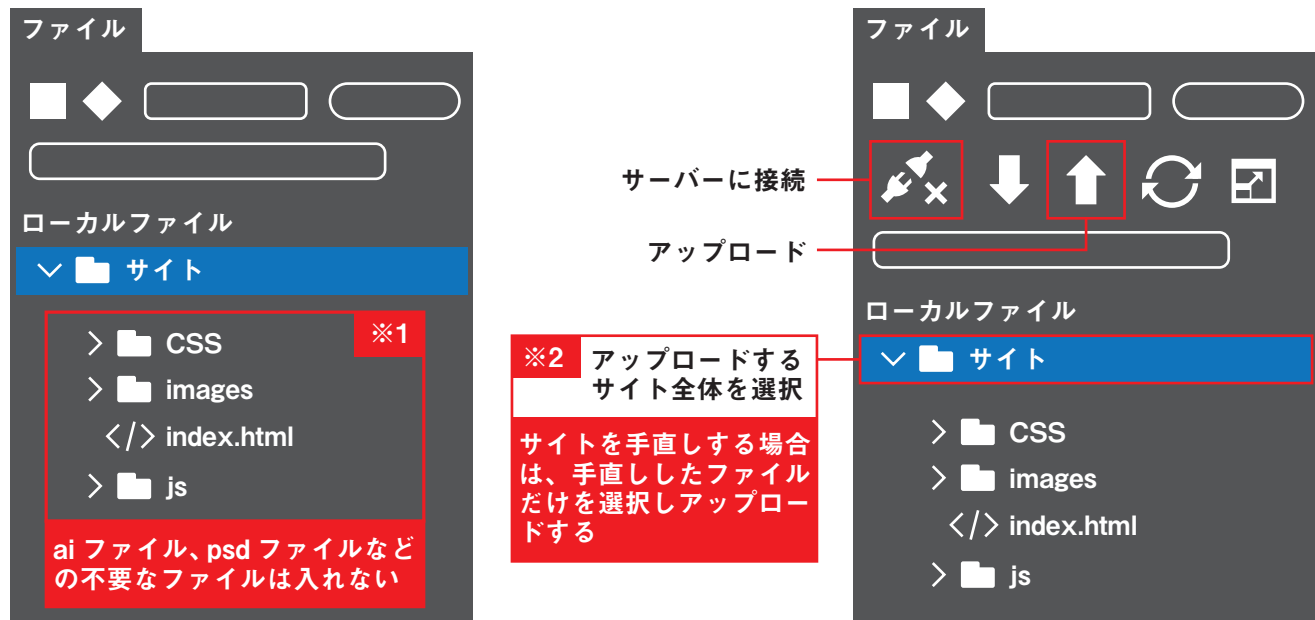


● サイトのアップロード

サイトをアップロードするソフトはいろいろありますが今回は Dreamweaver でアップロードします。

ファイルウィンドウのローカルファイル(※1)に不要なファイルがないことを確認します。次に「サイトの管理」 「サイトの編集」 「サーバー」 「+」 を選択し、「サーバー名」 「FTP アドレス」 「ユーザー名」 「パスワード」 「ルートディレクトリ」 を入力し、「テスト」 を選択、成功したら「保存」 します。

次に「リモートサーバーに接続」 「サイト(※2)」 を選択し、「ファイルをリモートサーバーに PUT する」 を選択するとアップロードが始まります。



※1 の不要なファイルをアップロードしたくないけど削除もしたくない場合は、不要なファイルの上で右クリック 「クローク」 「クローク実行」 を選択すると赤い斜線が引かれアップロードされません。

「ファイルをリモートサーバーに PUT する」 ボタンの 2 つ右の 「展開してローカルおよびリモートサイトを表示」 を選択するとファイルウィンドウが開かれ、左側にリモートサーバーのデータ、右側にローカルファイルのデータが表示される。このウィンドウでも上記の操作と同様にファイルのアップロードができます。