

【CSS】基礎

基礎文法編

■用語

- ↓セレクトー
h1 {
font-size: 12px;
}
↑プロパティ ↑値
|—————|
宣言

■基本

- ▶ コメントのしかた
- ▶ HTMLに紐づけ

■こうしたいならこのプロパティ（とこの値）

テキストに関するもの

- ▶ 色
- ▶ フォントサイズ
- ▶ 行揃え
- ▶ 文字間隔
- ▶ 太字
- ▶ 下線、取り消し線
- ▶ フォントの種類
- ▶ 行の高さ、行間
- ▶ 垂直方向の配置
- ▶ 影をつける

リストに関するもの

- ▶ 先頭の記号を変える
- ▶ 記号をリスト内側に
- ▶ 先頭の番号を画像に

【CSS】基礎

基礎文法編

■用語

- ↓セレクトー
h1 {
font-size: 12px;
}
↑プロパティ ↑値
|—————|
宣言

■基本

- ▶ コメントのしかた
- ▶ HTMLに紐づけ

■こうしたいならこのプロパティ（とこの値）

テキストに関するもの

- ▶ 色
- ▶ フォントサイズ
- ▶ 行揃え
- ▶ 文字間隔
- ▶ 太字
- ▶ 下線、取り消し線
- ▶ フォントの種類
- ▶ 行の高さ、行間
- ▶ 垂直方向の配置
- ▶ 影をつける

リストに関するもの

- ▶ 先頭の記号を変える
- ▶ 記号をリスト内側に
- ▶ 先頭の番号を画像に

▶ （上記三つの一括）	
-------------	--

ボックスモデルに関するもの

▶ 幅、高さ	
▶ できれば同じ大きさ	
▶ 文の溢れを制限	
▶ 境界線の太さ	
▶ 境界線の種類	
▶ 境界線の色	
▶ （上記三つの一括）	
▶ 境界線 4 辺個別設定	
▶ 境界線の角を丸める	
▶ 境界内領域を広げる	
▶ （上記一つの一括）	
▶ 境界外領域を広げる	
▶ （上記一つの一括）	
▶ 左右中央揃えにする	
▶ 右揃えにする	
▶ 配置方法を変更	
▶ 要素のサイズを可変に	
▶ なかったことに	
▶ 本来地を基準に配置	
▶ viewportを基準に配置	
▶ 親要素を基準に配置	
▶ 要素の重なり順制御	
▶ ボックスモデル変更	
▶ 影をつける	
▶ 背景色	
▶ 背景画像	
▶ 背画なるべく大きく	

▶ （上記三つの一括）	list-style
-------------	------------

ボックスモデルに関するもの

▶ 幅、高さ	width、height
▶ できれば同じ大きさ	min-width, max-width, min-height, max-height
▶ 文の溢れを制限	overflow
▶ 境界線の太さ	border-width
▶ 境界線の種類	border-style
▶ 境界線の色	border-color
▶ （上記三つの一括）	border
▶ 境界線 4 辺個別設定	border-top, boder-bottom, border-right, border-left
▶ 境界線の角を丸める	border-radius
▶ 境界内領域を広げる	padding-top, padding-bottom, padding-right, padding-left
▶ （上記一つの一括）	padding
▶ 境界外領域を広げる	margin-top, margin-bottom, margin-right, margin-left
▶ （上記一つの一括）	margin ※「マージンの相殺」付
▶ 左右中央揃えにする	margin-right: auto; margin-left: auto; か margin: 0 auto;
▶ 右揃えにする	margin-left: auto;
▶ 配置方法を変更	display ※ 配置に違和感あれば真っ先にこれをいじろう！
▶ 要素のサイズを可変に	display: inline-block;
▶ なかったことに	display: none;
▶ 本来地を基準に配置	position: relative;
▶ viewportを基準に配置	position: fixed; ※スクロールしても動かなくなる
▶ 親要素を基準に配置	position: absolute; ※「ページ全体を基準に」も
▶ 要素の重なり順制御	z-index
▶ ボックスモデル変更	box-sizing
▶ 影をつける	box-shadow
▶ 背景色	background-color
▶ 背景画像	background-image
▶ 背画なるべく大きく	background-size

▶ 背画を中心基に拡張	
▶ （上記四つの一括）	
▶ 端寄せて文囲ませる	
▶ 後続の文囲ませない	
▶ カーソルを変える	
▶ 変形(移動,回転,Resize)	
▶ 変形の起点を変更	
▶ ポインタに無反応に	
▶ 文章の範囲選択不可	

テーブルに関するもの

▶ 隣接セルが罫線共有	
-------------	--

画像に関するもの

▶ 大きさ	
▶ ボックス下枠に沿う	

■高度な技術

- ▶ ☆ 画面のマスク
- ▶ ☆ 三角形を作る

■プロパティの継承に関して

- ▶ ☆ 継承するプロパティ一覧
- ▶ ※ むりやり継承させたいなら、プロパティ名: inherit; とすると親と同じ設定になる。

■色の表現方法

- ▶ ☆ 色キーワードによる指定
- ▶ ☆ RGBAによる指定
- ▶ ☆ HSLAによる指定
- ▶ ☆ 透明度の指定

■ブラウザの初期設定に関する注意

- ▶ ☆ body に関して
- ▶ ☆ h1 などに関して
- ▶ ☆ リストを使うとき

▶ 背画を中心基に拡張	background-position	
▶ （上記四つの一括）	background	！！不正確かも！！
▶ 端寄せて文囲ませる	float	
▶ 後続の文囲ませない	clear	
▶ カーソルを変える	cursor	
▶ 変形(移動,回転,Resize)	transform	※アニメーションで利用！
▶ 変形の起点を変更	transform-origin	※アニメーションで利用！
▶ ポインタに無反応に	pointer-events: none;	
▶ 文章の範囲選択不可	user-select: none;	

テーブルに関するもの

▶ 隣接セルが罫線共有	border-collapse: collapse;	※ table 要素にしか使えない
-------------	----------------------------	-------------------

画像に関するもの

▶ 大きさ	width (, height)	※ height 省略すると縦横比を保ってくれる
▶ ボックス下枠に沿う	vertical-align: bottom;	※しないと余計な余白が

■高度な技術

- ▶ ☆ 画面のマスク
- ▶ ☆ 三角形を作る

■プロパティの継承に関して

- ▶ ☆ 継承するプロパティ一覧
- ▶ ※ むりやり継承させたいなら、プロパティ名: inherit; とすると親と同じ設定になる。

■色の表現方法

- ▶ ☆ 色キーワードによる指定
- ▶ ☆ RGBAによる指定
- ▶ ☆ HSLAによる指定
- ▶ ☆ 透明度の指定

■ブラウザの初期設定に関する注意

- ▶ ☆ body に関して
- ▶ ☆ h1 などに関して
- ▶ ☆ リストを使うとき

- ▶ ☆ 画像を使うとき
- その他の注意
- ▶ ※ 単位が違う計算をしたいとき → calc()を使う
 - ▶ ※ ある**クラスがついてるときだけ**ずらして配置するとき → margin をいじる発想
 - ▶ ※ ボックスのサイズ設定は、**padding と width** を基本にすべし！
 - ▶ ※ 2 者間に余白を設けたいな...って思った時は、**一方の要素だけ**に余白を設定すればよし
 - ▶ ※ できるだけ外側の部品をスタイリングして内側のものの配置・サイズを変更するべき？

セレクトー編

■基本セレクトーの種類

- ▶ 要素型セレクトー
- ▶ classセレクトー
- ▶ idセレクトー
- ▶ 属性セレクトー
- ▶ 全称セレクトー

■セレクトーを組み合わせる

- ▶ aとb
- ▶ aが付加されたb
- ▶ aの直下の階層にあるb
- ▶ aの下 の階層にあるb
- ▶ aの直後にあるb
- ▶ aと兄弟関係にあるb

■高度なセレクト

- 要素の特定の部分をセレクト** **: 擬似要素**
- ▶ 要素の前後
 - ▶ 要素の一行目
 - ▶ 要素の一文字目

- 特定の状態にある要素をセレクト** **: 擬似クラス**

- ▶ ☆ 画像を使うとき
- その他の注意
- ▶ ※ 単位が違う計算をしたいとき → calc()を使う
 - ▶ ※ ある**クラスがついてるときだけ**ずらして配置するとき → margin をいじる発想
 - ▶ ※ ボックスのサイズ設定は、**padding と width** を基本にすべし！
 - ▶ ※ 2 者間に余白を設けたいな...って思った時は、**一方の要素だけ**に余白を設定すればよし
 - ▶ ※ できるだけ外側の部品をスタイリングして内側のものの配置・サイズを変更するべき？

セレクトー編

■基本セレクトーの種類

- ▶ 要素型セレクトー h1 →対象: `<h1>...</h1>`
- ▶ classセレクトー .info →対象: `<p class="info">...</p>`
- ▶ idセレクトー #ok →対象: `<button id="ok">...</button>`
- ▶ 属性セレクトー [href] →対象: `...`
- ▶ 全称セレクトー * →対象: すべての要素

■セレクトーを組み合わせる

- ▶ aとb a, b 例: `h1 p`
- ▶ aが付加されたb ba 例: `h1.info`
- ▶ aの直下の階層にあるb a > b 例: `main > h1`
- ▶ aの下 の階層にあるb a b 例: `main h1`
- ▶ aの直後にあるb a + b 例: `p + p`
- ▶ aと兄弟関係にあるb a ~ b 例: `.like [type="checkbox"]:checked ~ .heart`

■高度なセレクト

- 要素の特定の部分をセレクト** **: 擬似要素**
- ▶ 要素の前後 ::before、::after擬似要素
 - ▶ 要素の一行目 ::first-letter擬似要素
 - ▶ 要素の一文字目 ::first-line擬似要素

- 特定の状態にある要素をセレクト** **: 擬似クラス**

▶ カーソルが乗ってる	
▶ クリックされてる間	
▶ 同階層の～番目	
▶ 同階層同要素～番目	
▶ 中身が空の要素	
▶ 中身が空でない要素	
▶ あるクラスがついてない	
▶ チェックされてる	
▶ リンクが未訪問	
▶ リンクが訪問済み	

■その他の注意

- ▶ ☆ スタイリングの優先順位

フレックスボックス編

■用語

- フレックスコンテナ
- フレックスアイテム
- ▶ ☆ 詳解

■こうしたいならフレックスコンテナにこのプロパティ（とこの値）

▶ フレックスボックスを適用	
▶ 主軸と交差軸の方向◆	
▶ 主軸方向の要素の配置方法	
▶ 交差軸方向の要素の配置方法	
▶ 要素がコンテナを越える際の対応◆	
▶ （上の◆の一括）	
▶ 交差軸方向の複数行の配置方法	

■こうしたいならフレックスアイテムにこのプロパティ

▶ 交差軸方向の行内での要素別の配置方法	
▶ 並び順を変えたい	

▶ カーソルが乗ってる	:hover擬似クラス
▶ クリックされてる間	:active ※クリックされてから離されるまでの状態 (e.g.)
▶ 同階層の～番目	:nth-child、:first-child、:last-child擬似クラス
▶ 同階層同要素～番目	:nth-of-type、:first-of-type、:last-of-type擬似クラス
▶ 中身が空の要素	:empty
▶ 中身が空でない要素	:not(:empty)
▶ あるクラスがついてない	:not(.クラス名)
▶ チェックされてる	:checked ※チェックボックスなどで
▶ リンクが未訪問	:link ※ a要素などで
▶ リンクが訪問済み	:visited ※ a要素などで

■その他の注意

- ▶ ☆ スタイリングの優先順位

フレックスボックス編

■用語

- フレックスコンテナ
- フレックスアイテム
- ▶ ☆ 詳解

■こうしたいならフレックスコンテナにこのプロパティ（とこの値）

▶ フレックスボックスを適用	display: flex;
▶ 主軸と交差軸の方向◆	flex-direction
▶ 主軸方向の要素の配置方法	justify-content
▶ 交差軸方向の要素の配置方法	align-items
▶ 要素がコンテナを越える際の対応◆	flex-wrap
▶ （上の◆の一括）	flex-flow
▶ 交差軸方向の複数行の配置方法	align-content

■こうしたいならフレックスアイテムにこのプロパティ

▶ 交差軸方向の行内での要素別の配置方法	align-self
▶ 並び順を変えたい	order

▶ 主軸方向の要素の長さ	
▶ （上記一つの一括）	

- 注意
- ▶ ※ 画面上縦いっぱい要素を埋め尽くしたいときは注意
 - ▶ ※ フレックスボックスにはマージンの相殺が起きない
 - ▶ ※ 子要素を親要素の中央に配置したいときは注意
 - ▶ ※ 2つの要素を左右隅に切り離して設置したいときは注意
 - ▶ ※ リストを横並びにしたいときflex-direction: row;は不要
 - ▶ ※ メニューをつくるときは注意
 - ▶ ※ 1画像1文章が横並び群の縦連なりで偶数番目のみ左右逆にしたいときは注意

レスポンスウェブデザイン編

- レスポンスウェブデザイン
- ▶ ☆ レスポンスウェブデザインのためのHTMLテンプレ
 - ▶ ☆ メディアクエリという仕組みでviewportの幅に応じてスタイルを切り替えられる
 - ▶ ☆ 簡潔な記述方法（モバイルファースト）
 - ▶ ☆ ブラウザ幅を広げると配置が縦から横に変わりさらに広告も加わるサイト（例）

グリッドレイアウト編

- こうしたいならコンテナにこのプロパティ（とこの値）
- ▶ グリッドレイアウトを適用
 - ▶ ボックス間に隙間を
 - ▶ たくさんの列や行
 - ▶ 何行になるか不明
 - ▶ 画面に入る限りの列
 - ▶ (↑で) 余白作らない
 - ▶ グリッドに名付け
 - ▶ 下へと配置してゆく

▶ 主軸方向の要素の長さ	flex-grow や flex-shrink や flex-basis
▶ （上記一つの一括）	flex

- 注意
- ▶ ※ 画面上縦いっぱい要素を埋め尽くしたいときは注意
 - ▶ ※ フレックスボックスにはマージンの相殺が起きない
 - ▶ ※ 子要素を親要素の中央に配置したいときは注意
 - ▶ ※ 2つの要素を左右隅に切り離して設置したいときは注意
 - ▶ ※ リストを横並びにしたいときflex-direction: row;は不要
 - ▶ ※ メニューをつくるときは注意
 - ▶ ※ 1画像1文章が横並び群の縦連なりで偶数番目のみ左右逆にしたいときは注意

レスポンスウェブデザイン編

- レスポンスウェブデザイン
- ▶ ☆ レスポンスウェブデザインのためのHTMLテンプレ
 - ▶ ☆ メディアクエリという仕組みでviewportの幅に応じてスタイルを切り替えられる
 - ▶ ☆ 簡潔な記述方法（モバイルファースト）
 - ▶ ☆ ブラウザ幅を広げると配置が縦から横に変わりさらに広告も加わるサイト（例）

グリッドレイアウト編

- こうしたいならコンテナにこのプロパティ（とこの値）
- ▶ グリッドレイアウトを適用 display: grid; grid-template-columns grid-template-rows
 - ▶ ボックス間に隙間を gap
 - ▶ たくさんの列や行 grid-template-columns: repeat(〇, △px); や -rows: " ;
 - ▶ 何行になるか不明 grid-auto-rows
 - ▶ 画面に入る限りの列 grid-template-columns: repeat(auto-fill, △px);
 - ▶ (↑で) 余白作らない grid-template-columns: repeat(auto-fill, minmax(〇px, △px));
 - ▶ グリッドに名付け grid-template-columns grid-template-rows
 - ▶ 下へと配置してゆく grid-auto-flow: column;

▶ 空枠を埋める	
▶ 視覚的に要素を配置	
▶ 枠未満の要素の揃え	

■こうしたいならアイテムにこのプロパティ

▶ 好きな位置に配置	
▶ 名付グリッドに配置	
▶ 要素を大きくする	

CSSアニメーション

■transition-somtihig, transitionによる手法

- ▶ ※ この手法では、変化前と変化後の状態(スタイル)を設定して、その途中をtransitionプロパティなどで補完する。

こうしたいならこのプロパティ（とこの値）

▶ どのプロパティを滑らかにするか	
▶ 変化にかかる時間	
▶ アニメーション開始を遅らせる	
▶ 緩急をつける	
▶ （上記の一括）	

注意

- ▶ ☆ 基本の書き方
- ▶ ☆ 移動、回転、リサイズの仕方
- ▶ ☆ Chromeでのバグの回避のために必須のこと

■animation-somtihig, animationによる手法

- ▶ ※ この手法では、変化前、変化後だけではなくてその途中に好きなタイミングでキーとなるフレームを挟んで、そのキーフレームごとにスタイルを設定して、アニメーションさせていくことができる。さらに、transitionによる手法とはちがい、アニメーションを繰り返したり巻き戻せたりする。

こうしたいならこのプロパティ（とこの値）

▶ どのキーフレームを適用するか	
------------------	--

▶ 空枠を埋める	grid-auto-flow: … dense;	※値にdenseを追加すること
▶ 視覚的に要素を配置	grid-template-areas	
▶ 枠未満の要素の揃え	justify-content align-content justify-items align-items	

■こうしたいならアイテムにこのプロパティ

▶ 好きな位置に配置	grid-row grid-column
▶ 名付グリッドに配置	grid-row grid-column
▶ 要素を大きくする	grid-row: span O; や grid-column: span O;

CSSアニメーション

■transition-somtihig, transitionによる手法

- ▶ ※ この手法では、変化前と変化後の状態(スタイル)を設定して、その途中をtransitionプロパティなどで補完する。

こうしたいならこのプロパティ（とこの値）

▶ どのプロパティを滑らかにするか	transition-propaty	※ コンマ忘れない
▶ 変化にかかる時間	transition-duration	
▶ アニメーション開始を遅らせる	transition-delay: Os;	
▶ 緩急をつける	transition-timing-function	
▶ （上記の一括）	transition	

注意

- ▶ ☆ 基本の書き方
- ▶ ☆ 移動、回転、リサイズの仕方
- ▶ ☆ Chromeでのバグの回避のために必須のこと

■animation-somtihig, animationによる手法

- ▶ ※ この手法では、変化前、変化後だけではなくてその途中に好きなタイミングでキーとなるフレームを挟んで、そのキーフレームごとにスタイルを設定して、アニメーションさせていくことができる。さらに、transitionによる手法とはちがい、アニメーションを繰り返したり巻き戻せたりする。

こうしたいならこのプロパティ（とこの値）

▶ どのキーフレームを適用するか	animation-name
------------------	----------------

▶ アニメーションにかかる時間	
▶ アニメーション開始を遅らせる	
▶ 初期状態に戻らず最終形のままに	
▶ 繰り返す回数	
▶ 再生順（逆再生？ループ？）	
▶ 緩急をつける	
▶ （上記の一括）	

注意

- ▶ ☆ 基本の書き方
 - ▶ ☆ 0%, 100%のみの記述ならfrom, toに代替可能
 - ▶ ☆ キーフレームで同じスタイルが続くなら `◯%, △% {` と書ける
- 実例集（新しいことは書いてないので見なくても構わない）
- ▶ ☆ ボタンをふわっと変色する
 - ▶ ☆ ローディングアイコンを作る
 - ▶ ☆ ポップアップを作る
 - ▶ ☆ ハートアイコンを作る

CSS変数

- CSS変数
- ▶ ☆ 定義の仕方
- 注意
- ▶ ☆ 数字を値に持つ変数に単位をあとからつける工夫
 - ▶ ☆ テーマカラーを一気に変える

Webデザイン用語

- 未分類
- ヒーローイメージ
 - ナビゲーションペイン

▶ アニメーションにかかる時間	animation-duration
▶ アニメーション開始を遅らせる	animation-delay: 0s;
▶ 初期状態に戻らず最終形のままに	animation-fill-mode: forwards;
▶ 繰り返す回数	animation-iteratoin-count
▶ 再生順（逆再生？ループ？）	animation-direction
▶ 緩急をつける	animation-timing-function
▶ （上記の一括）	animation

注意

- ▶ ☆ 基本の書き方
 - ▶ ☆ 0%, 100%のみの記述ならfrom, toに代替可能
 - ▶ ☆ キーフレームで同じスタイルが続くなら `◯%, △% {` と書ける
- 実例集（新しいことは書いてないので見なくても構わない）
- ▶ ☆ ボタンをふわっと変色する
 - ▶ ☆ ローディングアイコンを作る
 - ▶ ☆ ポップアップを作る
 - ▶ ☆ ハートアイコンを作る

CSS変数

- CSS変数
- ▶ ☆ 定義の仕方
- 注意
- ▶ ☆ 数字を値に持つ変数に単位をあとからつける工夫
 - ▶ ☆ テーマカラーを一気に変える

Webデザイン用語

- 未分類
- ヒーローイメージ
 - ナビゲーションペイン

