

Layout parts

課題説明

まずクローンしてください。

方法については以前の資料を参照してください。

URLは下記のURLからアクセスしてください。

[GitHub](#)

クローンできたら来たフォルダーを開いてください。

中に「q6～q10」の名前のついたフォルダーがあります。

さらにその中を開くと、開いたフォルダーと同じフォルダーと「answer」の名前のついたフォルダーがあります。

どちらもHTMLファイルとCSSファイルが用意されています。

「answer」の中にあるHTMLファイルをブラウザで開き、それを参考にしながら「q6～q10」内にあるHTMLファイルとCSSファイルをエディターで開いて同じものを作ってください。

問題ごとに説明と画像に背景の色なども明記してますのでしっかり目を通してください。

CSSファイルは3つありますが、「style.css」にCSSを書いてください。

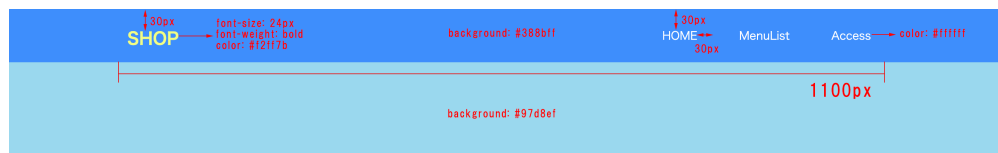
ベースは書いているのでどこから自分が書いていくのかも観察しながら進めてください。

ただディベロッパーツールなどを使うと答えが分かってしまいます。なので**極力見ないで**取り組んでください。

ネットで調べる、質問をするなどしたけどどうしてもできない時は見ても良いですが、HTMLがどう言う構造、タグを使っているか、CSSは何を使っているかをしっかりと観察と調査してください。

レビューの時に抜き打ちで質問させていただきます。

Q6



よくあるナビバーです。

ロゴ（SHOP）とナビリンク（HOMEなど）はaタグを使ってください。こういったものは何かしらのリンクになっていることが多いからです。

href属性ですが今回は、`href="#"`と書いてください。こう書くことによって同じindex.htmlをリンクとして扱えます。

開発途中でリンク先が決まってない時になどによく使います。

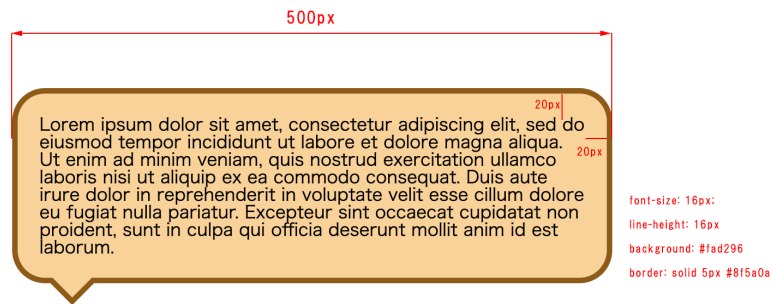
右側のナビリストですが以下の構造で作成お願いします。

```
nav > ul > li > a
```

ナビリンクのフォントサイズはとくに指定してないのでそのまま大丈夫です。

横並びの方法はお任せします。

Q7



これもよく見かける吹き出しです。

吹き出し自体はそこまで難しくありませんが、下の出っ張りを作るのが難しいです。

ネットで調べながら作って見てください。

位置は特指定しませんが、できるだけサンプルと同じ位置に配置してください。

配置する基準を決めた時に0よりもしたの位置に配置したい場合は`-○○px`でできます。

重要なCSS


```
border-radius
```

```
:before
```

```
:after
```

Q8

この課題はとくに余白などを使ってないので実際にブラウザで表示されているものを参考しながら読んでください。

今までは画像をタグを使用して表示していましたが、画像を背景として使用することもあるのでその練習をします。

画像はimageフォルダーの中に用意してますのでそちらを使用してください。

また背景画像は表示するにはコンテンツの高さが必要です。ですが今回はとくに要素を作る作業はないので、`height: 500px;`で無理矢理ではありますが高さを作ってください。

Q8-1

まずは背景画像としてimageフォルダーの画像を表示させてください。

Q8-2

Q8-1の背景を見ると、画像がリピートしてますね。

連続した模様のようなものであれば良いのですが、こう言った写真は連続させたくないのでリピートしないようにしてください。

Q8-3

Q8-2でリピートは解除できましたが画像が中心に寄っていないのと、画像を画面いっぱいに表示させたい時もあります。

なのでこの問題では画像を中央に寄せ、画面いっぱいの幅にしてください。

その時に注意なのがリピートを解除することを忘れないでください。

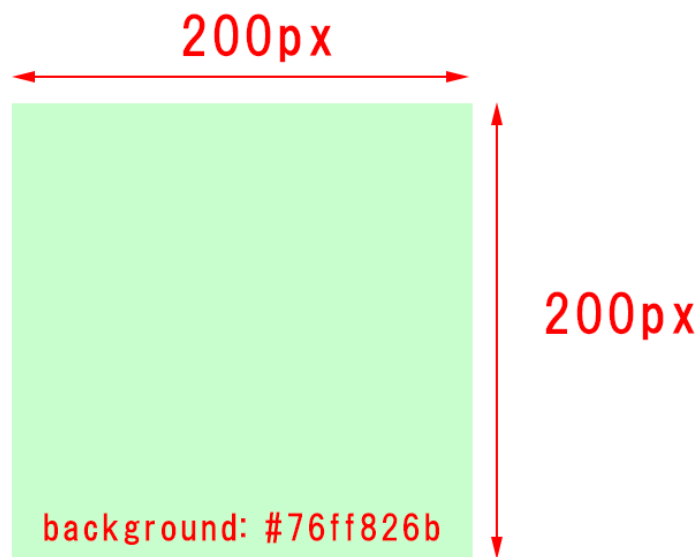
Q8-4

こちらはQ8-3のリピートの解除と画像を中央に寄せるところまで是一緒ですが、今までは拡大などがされていましたがすべて収まるようにして表示させてください。

重要なCSS

```
background
```

Q9



Q9-1~Q9-3までは上記と同じ箱を使ってください。

今回はアニメーションについて練習します。

アニメーションのついたWebサイトは数多く存在し、複雑な動きがあるものはjQueryなどを使用しますが、簡単な動きはCSSだけで実装できますのでその練習をします。

Q9-1

まずはページが読み込まれたタイミングでフェードインするようにしてください。

どのくらいの時間をかけてフェードインする設定ができるのですがそれは**4秒**をお願いします。

Q9-2

この問題ではページが読み込まれて**4秒経過**したらフェードアウトさせてください。

この時も何秒かけてフェードアウトさせるのか設定できますが**4秒**をお願いします。

Q9-3

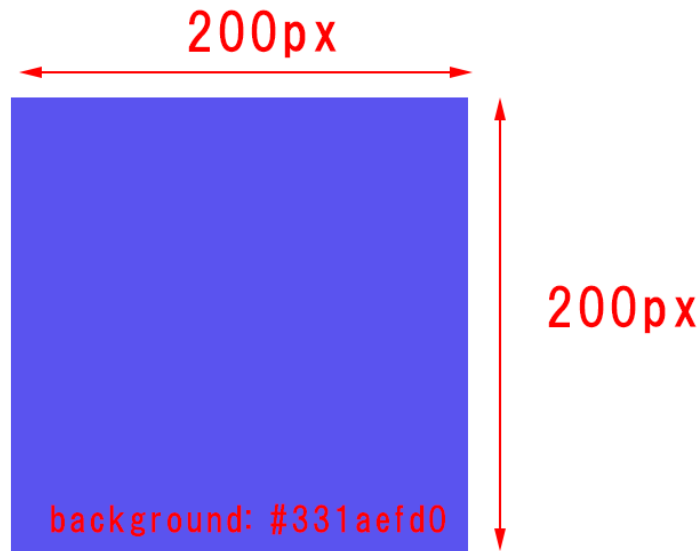
この問題ではページが読み込まれてから**永遠**と箱が回るようにしてください。

こちらは何秒かけて1回転するのか設定できますが、**4秒**をお願いします

重要なCSS

```
animation
```

```
@keyframes
```



Q9-4~Q9-11までは上記と同じ箱を使ってください。

前回まではアニメーションのタイミングを自分で決めていましたが、今回はマウスカーソルが要素の上に来た時に動くよう実装してください。

Q9-4

マウスカーソルが要素の上に来た時、箱の色を青からピンクにしてください。

変更後のCSSは`background: #ff27e2c9;`でお願いします。

Q9-5

マウスカーソルが要素の上に来た時、箱が`360°`回転するようにしてください。

Q9-6

マウスカーソルが要素の上に来た時、箱が上に`20%`ほど動くようにしてください。

Q9-7

マウスカーソルが要素の上に来た時、箱が右に`20%`ほど動くようにしてください。

Q9-8

マウスカーソルが要素の上に来た時、箱の高さが`300px`になるようにしてください。

Q9-9

マウスカーソルが要素の上に来た時、箱の横幅が500pxになるようにしてください。

Q9-10

マウスカーソルが要素の上に来た時、箱の周りに枠線を表示させてください。その時のスタイルを以下を使用してください。

```
border: 10px #00ffe5 solid;
```

またこの時にただ枠線を付ける書き方をすると箱が大きくなったように見えるのでCSS`box-sizing`を使ってください。

使用するプロパティに関しては調べてみてください。

Q9-11

マウスカーソルが要素の上に来た時、箱の周りに影を表示させてください。その時のプロパティは以下を使用して下さい。

```
8px 11px 10px 3px #06031ed0;
```

使うCSSについては調べてみてください。

注意点

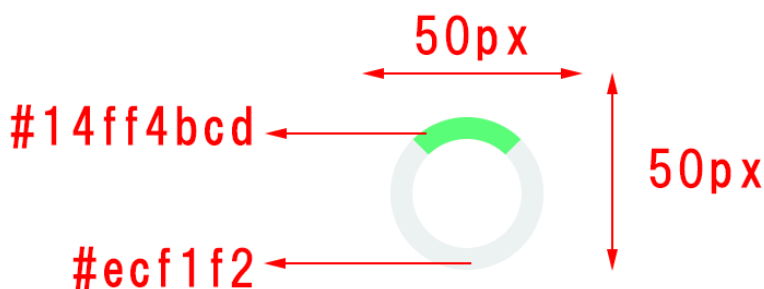
ホバー時にどのくらいの時間をかけて変化させるのか時間を決めないと、カーソルを当てた際いきなり箱が変化してアニメーション とも言えないのでその設定もしてください。

そのQ9-4~Q9-11は共通で0.5秒で設定してください。

重要なCSS

```
transition  
  
:hover
```

Q9-12



よくロード中や処理待の時に表示されるサークルです。

これは**Q9-3**でやったことを活かせば簡単です。

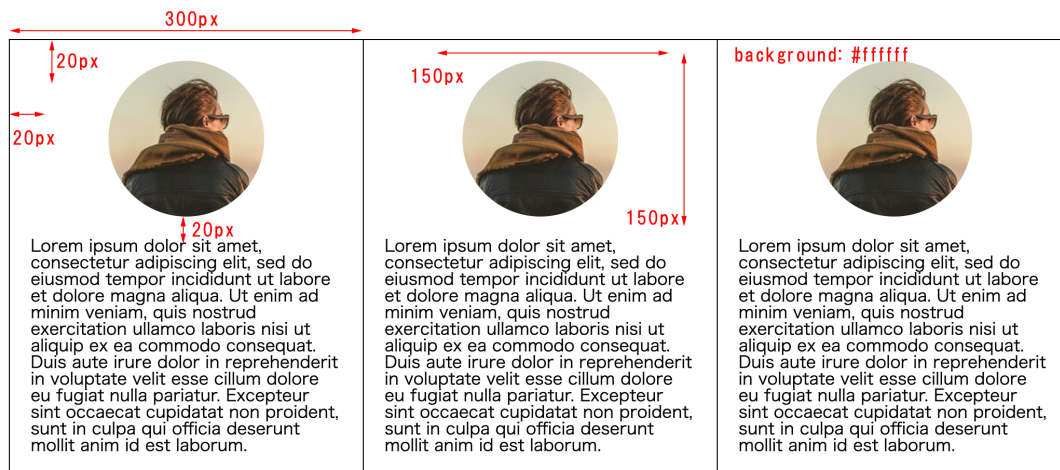
問題は丸の作り方だと思いますが、箱を作って枠線の色を設定を行い、丸くするだけです。

枠線の太さは**10px**にしてください。

アニメーションは**1.1秒**かけて1周させ、永遠に回るように設定してください。

今回はとくにヒントを出しません。過去に作成したものを参考にすればできるので自力でやって見てください。

Q9-13



こちらもスタッフ一覧でよく見るサイトです。

今どのスタッフを選択しようとしているのか分かり易くする為にカーソルをあてている部分は影が付くようになっています。

写真は背景画像として表示させてください。

設定はリピートなし、中央寄せ、要素を覆う設定でお願いします。

枠線のプロパティは以下を使用してください。

1px solid #000000;

影のプロパティは以下を使用してください。

0px 0px 10px 3px #424242d0;

重要なCSS

z-index

他にもありますが、過去に作成したものを参考にすればできるので自力でやって見てください。

Q10

今回はボタンです。

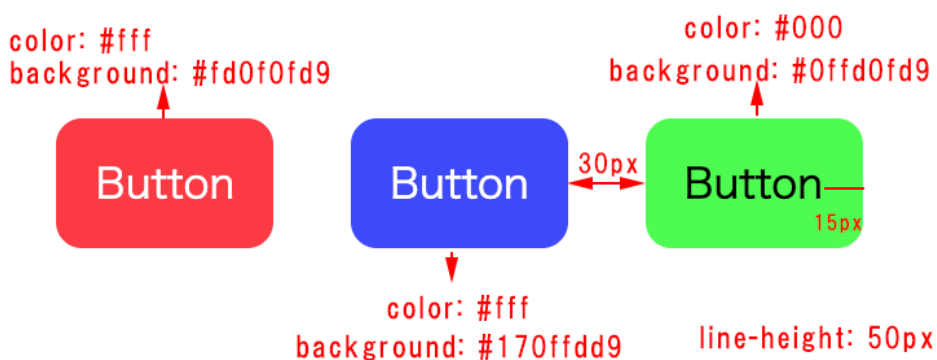
ボタンも色々な種類のデザイン、動きのボタンがあります。

覚えることはできないのでデザインや、要件定義を見てイメージできるくらいになっていきましょう！

本来ボタンは[a](#)タグや、[button](#)タグを使って作りますが、他にも練習していただきたいことがあるので今回は[div](#)タグで作成してください。

もし[a](#)タグや、[button](#)タグで作成していた場合、問題文を読んでないとみなし再レビューとするので注意してください。

Q10-1



まずはフラットなボタンを作ってみましょう。

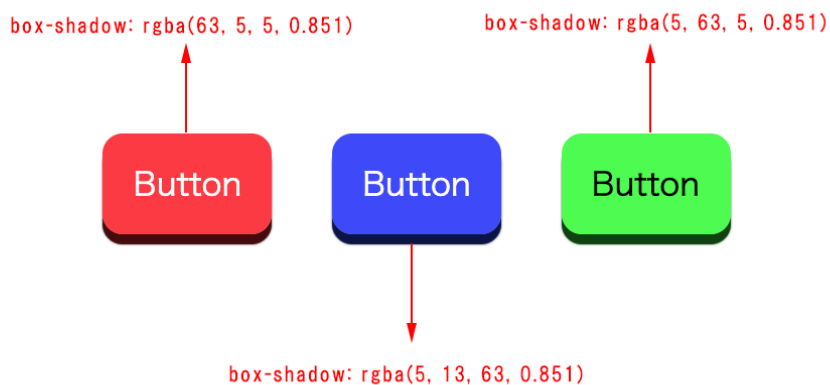
角丸は10pxで設定してください。

完成したら、マウスカーソルがポインターに切り替わるとこともCSSで実装してください。

重要なCSS

cursor

Q10-2



次にボタンに影を付けてください。

ボタンのベースはQ10-1と同じものを使用してください。

影を付けるだけで同じボタンにもかかわらず立体感が生まれます。

影のスタイルを自分で付けてみてください。色は画像を参考にしてください。

Q10-3

次は影のついたボタンを押した時に影がなくなるようにしてください。

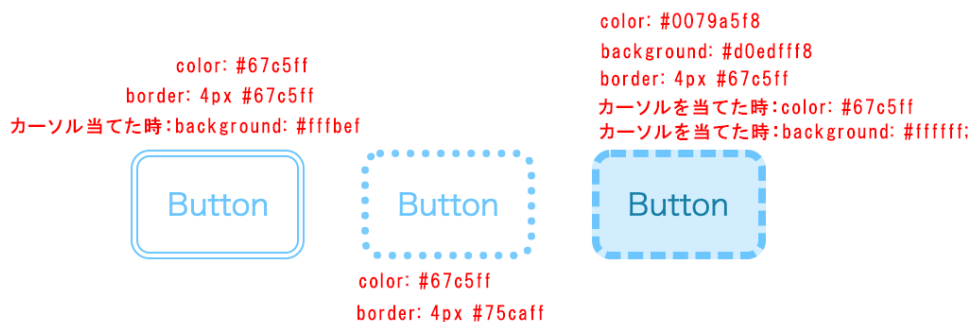
ボタンのベースはQ10-1と同じものを使用してください。

影を無くすだけでは押した感が生まれないのもうひと工夫加えてください。

重要なCSS

```
:active  
  
transform
```

Q10-4



この問題では、マウスカーソルが当たった時にボタンのデザインが変わる実装をしてもらいます。

ボタンのベースはQ10-1と同じものを使用してください。

色、背景、カーソル時の色、枠線の太さ、色は画像を参照してください。

枠線の種類に関してはあえて記載していないので自分で調べてみてください。

あとは過去に作成したものを参考にすればできるので自力でやってみてください。

課題提出とレビュー依頼について

課題提出はまず作業が終えてからGitHubの自分のリポジトリにプッシュしてください。

GitHubのURLをClassroomにて提出してください。

レビュー依頼がありましたら、メンターがクローンして実際に確認します。

コードが正しく書けているかを確認した後、こちらからいくつか質問をするのでその答えが正しければ課題クリアとなります。

課題途中でも気になったことなどがあれば遠慮なく質問してください。