

# Web プログラミング

第九回 : CSS(6)

## CSS(6) - 「作り方」コンテンツの実装

1

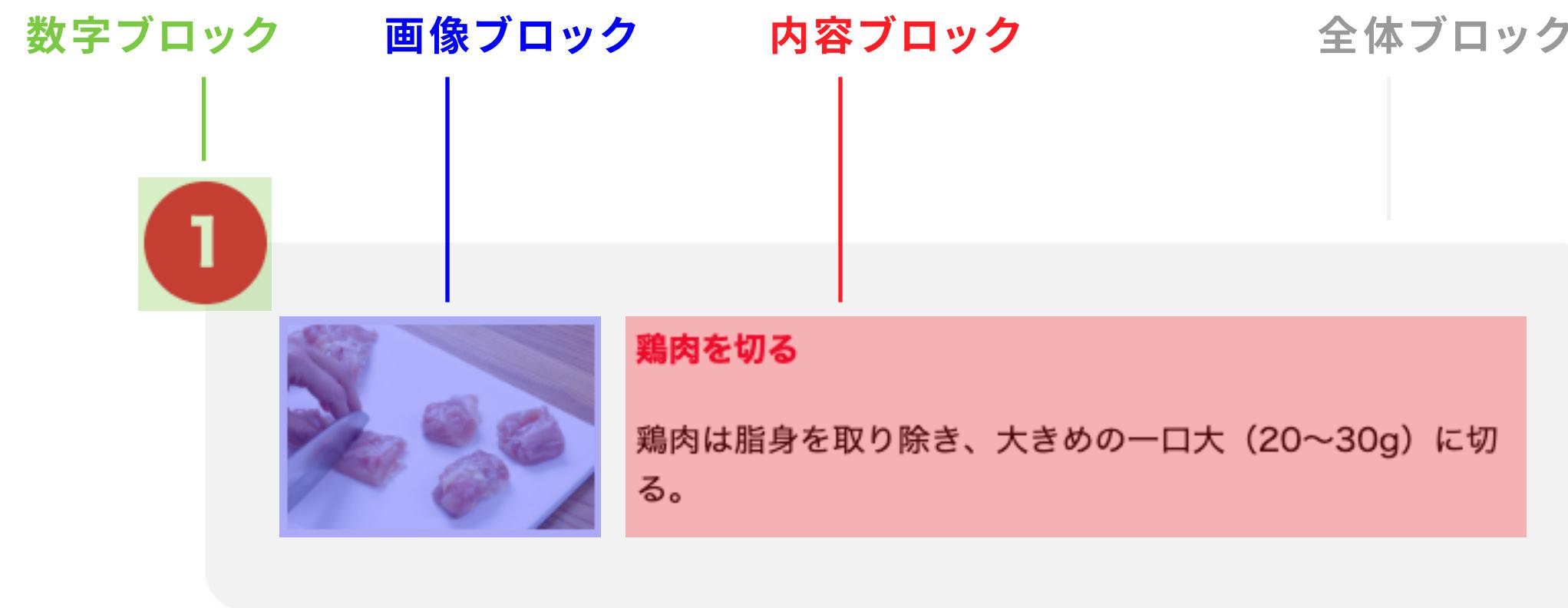


鶏肉を切る

鶏肉は脂身を取り除き、大きめの一口大（20～30g）に切る。

「作り方」のコンテンツは複雑そうに見えますが、構造を把握し、実装方法を整理すれば簡単に実装できます。まずはこのレイアウトを実現するための HTML 構造を把握しましょう。

## CSS(6) - 「作り方」コンテンツの実装



各ステップは、全体を囲むブロックの中に

- ・数字ブロック
- ・画像ブロック
- ・内容ブロック

の 3 つのブロックが内包される形になります。

## CSS(6) - 「作り方」コンテンツの実装



配置位置の指定や横並びなど  
レイアウトを定義する `css` をオフにした時、  
このように並びます。  
まずはこの形を作成し、レイアウトの作成を始めましょう。

## CSS(6) - 「作り方」コンテンツの実装

1

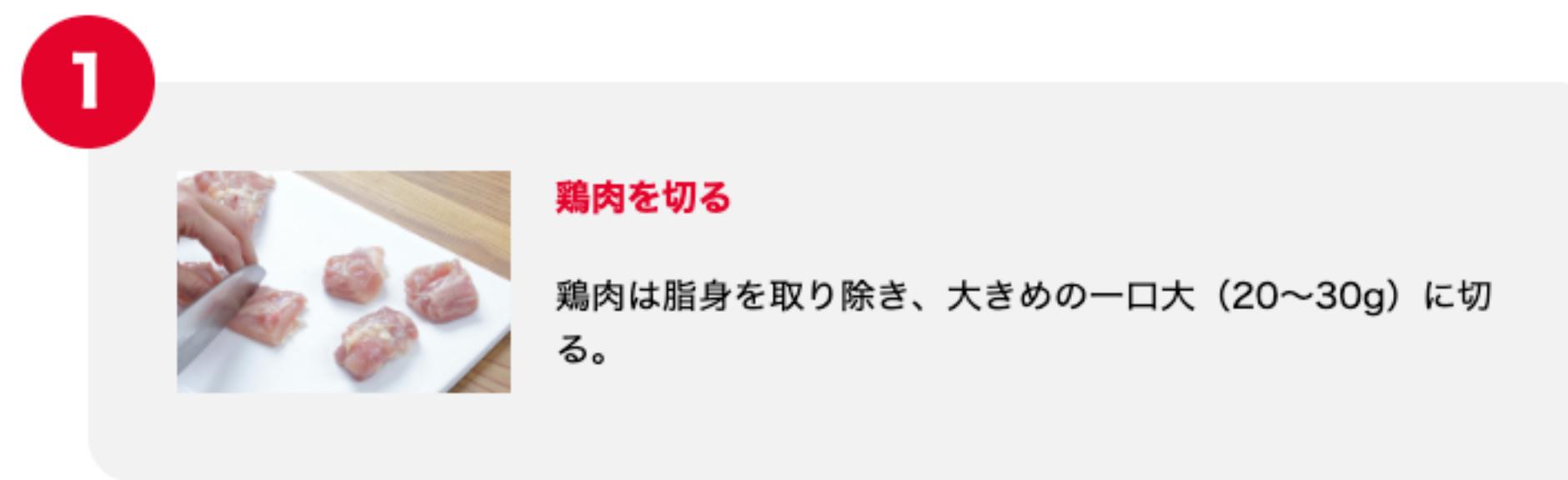


鶏肉を切る

鶏肉は脂身を取り除き、大きめの一口大（20～30g）に切る。

まずは数字ブロックを **position: absolute** に設定します。  
全体ブロックを基準に、**left:-30px**、**top:-30px** にしていすると、  
全体ブロック左上すこしあみ出した位置にレイアウトできます。

## CSS(6) - 「作り方」コンテンツの実装



その後、全体ブロックを `display:flex` に指定し、内包する要素を横並びにすることでレイアウト完了です。  
数字ブロックは `position:absolute` なので、レイアウトルールに則らず、左上にいつづけます。

## CSS(6) - **iframe** について

**<iframe>** タグは、外部のサイトを自分のサイトに埋め込むことができるタグです。

**src** 属性に書いた URL で指定されたリソースを、**iframe** の内部に表示します。

```
<iframe src="https://www.nagoya-bunri.ac.jp/"></iframe>
```

iframe タグ

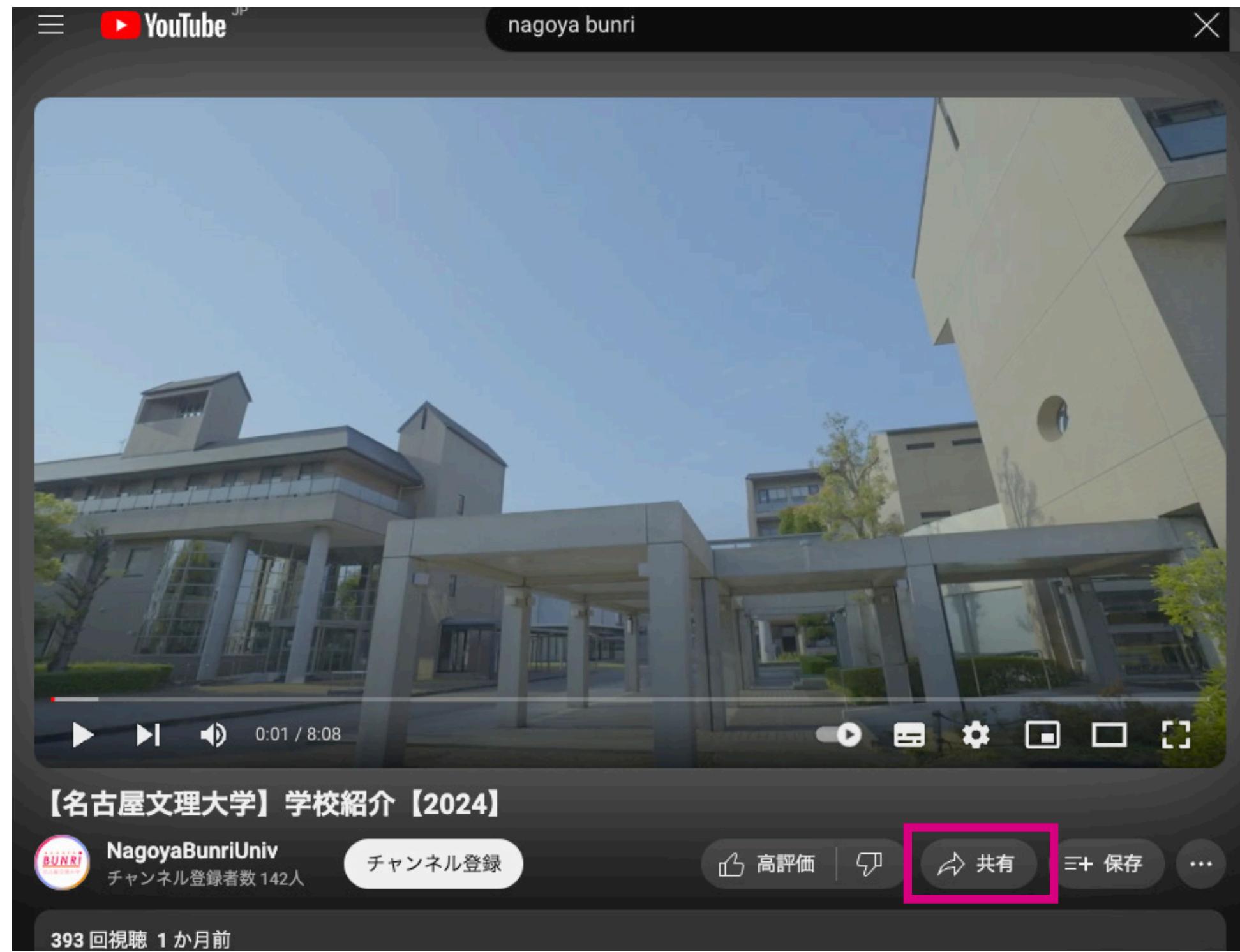


## CSS(6) - iframeについて

YouTube や GoogleMap といったサービスは、  
コンテンツを自分のページに埋め込めるよう、**iframe** タグを発行しています。  
これらを利用して、地図や動画を自分のページに表示することができます。

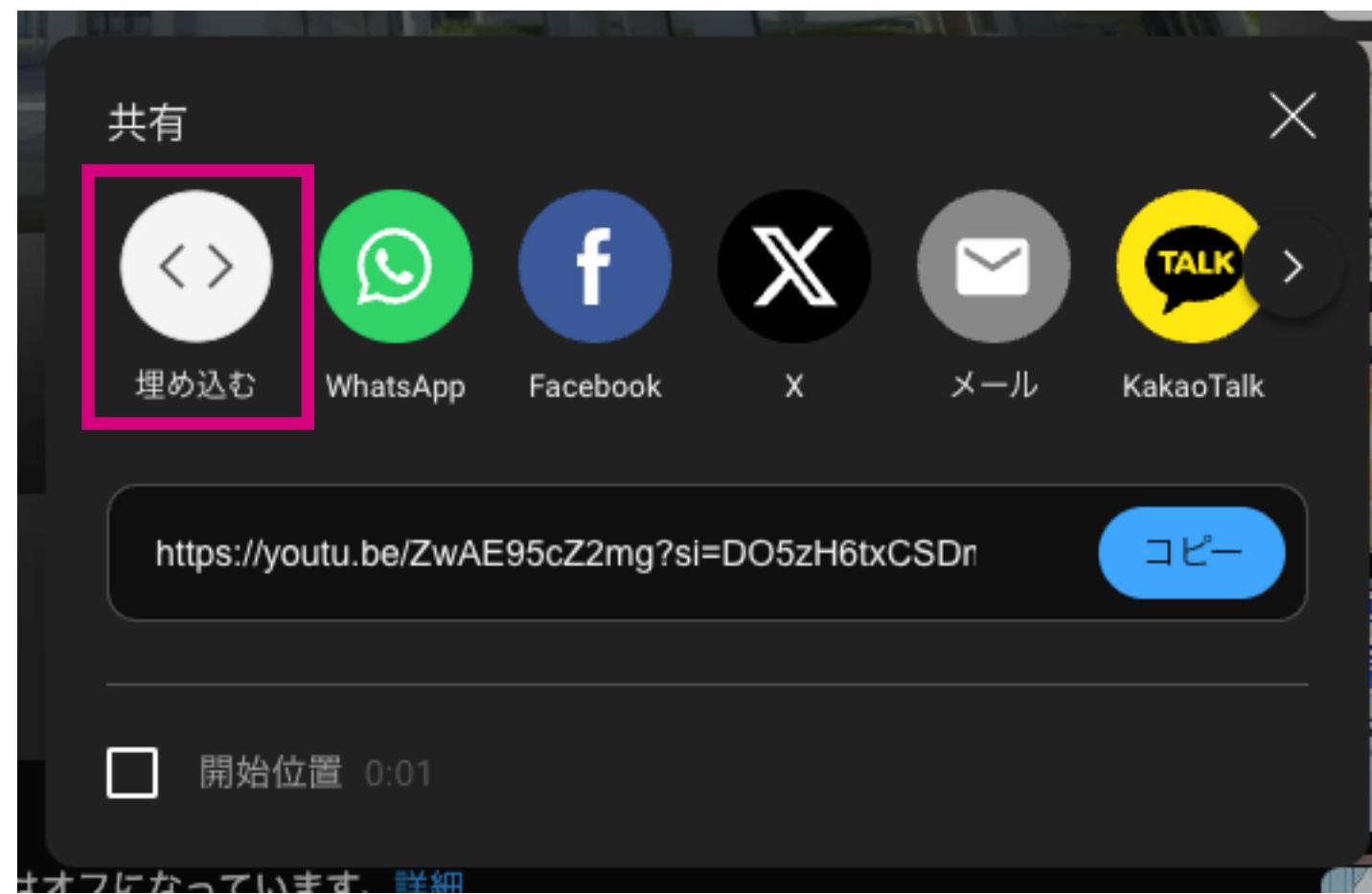


## CSS(6) - YouTube から埋め込みコードの取得



Youtube 動画再生ページにある  
「共有」ボタンを押して、  
popup-upメニューを表示します。

## CSS(6) - YouTube から埋め込みコードの取得



ポップアップメニュー内の  
「埋め込む」をクリックします。

## CSS(6) - YouTube から埋め込みコードの取得



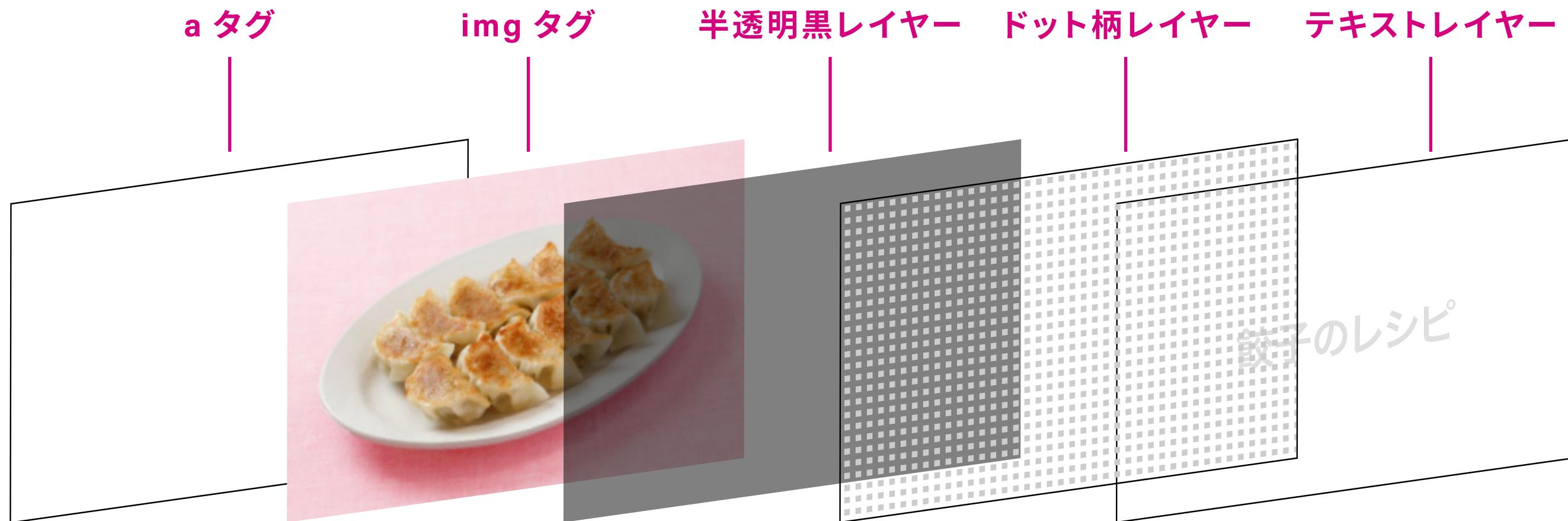
iframe タグが表示されるので、  
これをコピーして自分のページの HTML にペーストします。

## CSS(6) - 「その他のレシピ」の実装



「その他のレシピ」の実装も、  
パーツを分割し、レイアウト構成を考えましょう。  
まずは、並んでいるボタンの1つだけを作ってみます。

## CSS(6) - 「その他のレシピ」の実装



ボタンブロックの考え方とは、ファーストビューの時とほぼ同じです。

a タグを `display:block` にして幅と高さを指定し、  
その中に画像、半透明黒、ドット柄、テキストを重ね合わせる形でレイアウトします。

マウスオーバーで画像を動かすことを考慮し、  
`background-image` ではなく、img タグを配置する形で実装します。

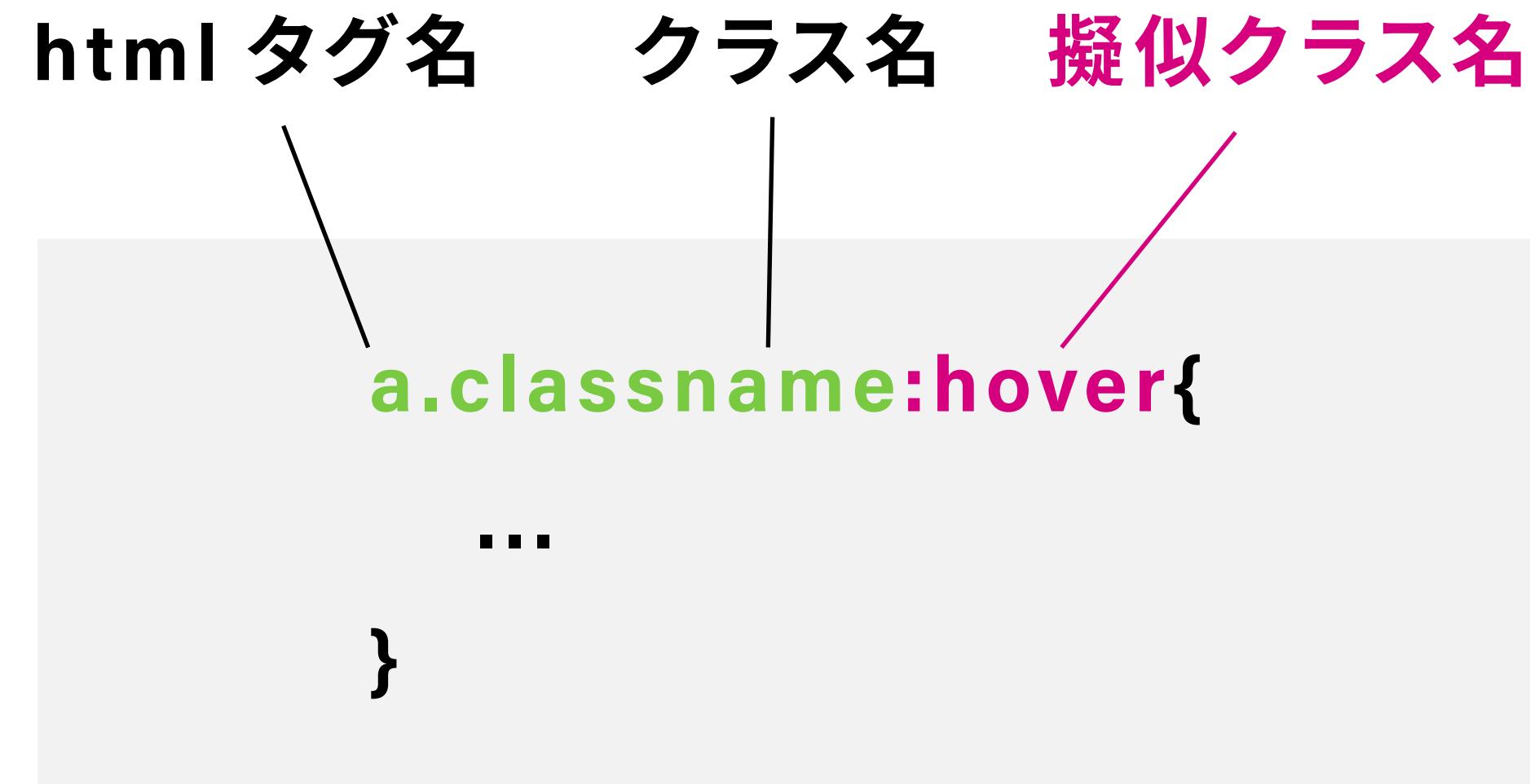
## CSS(6) - 「その他のレシピ」の実装



ボタンブロックを作成できたら、  
コピーして画像やテキスト内容を編集して複数作成します。  
その後、全体を囲むブロックに `display:flex` を指定し、  
レイアウトを調整して完了です。

## CSS(6) - 「擬似クラス」について

擬似クラスとは特殊なcssのセレクタとして使えるクラスです。  
「特定の条件にある時のみ有効になる」クラスとして使用できます。  
セミコロン(;)の後に、擬似クラス名を書いて指定します。



## CSS(6) - 「擬似クラス」について

**:hover** 擬似クラスは  
「マウスオーバーしている時」 有効になる擬似クラスです。  
これを使用すると、マウスオーバーした時に css を変化させたりできます。

リンク

リンク  


```
a{  
  color: blue;  
}  
a:hover{  
  color: red;  
}
```

## CSS(6) - 「擬似クラス」について

`a:hover{ ... }`

マウスオーバーした時に有効化

`a:visited{ ... }`

すでにクリックしたことのあるリンクのみに対して有効化

`input:focus{ ... }`

入力欄などをクリックしてアクティブにした時に有効化

`li:first-child{ ... }`

同じ階層にいる要素（兄弟要素）の中で、最初の一つ目に対して有効化

`li:last-child{ ... }`

同じ階層にいる要素（兄弟要素）の中で、最後の一つ目に対して有効化

`li:only-child{ ... }`

同じ階層にいる要素（兄弟要素）が無い時に有効化

`li:nth-child(odd){ ... }`

同じ階層にいる要素（兄弟要素）の中で、奇数番目の要素に対して有効化

`li:nth-child(even){ ... }`

同じ階層にいる要素（兄弟要素）の中で、偶数番目の要素に対して有効化

他にも色々あるので参考リンクなどから調べてみてください。

<https://developer.mozilla.org/ja/docs/Web/CSS/Pseudo-classes>