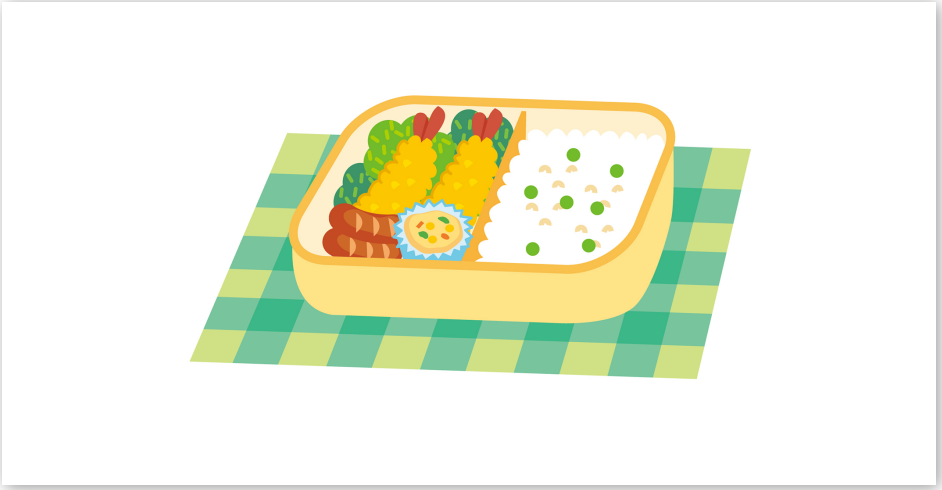
HTML,CSSの実践

レイアウトテクニックを理解する

HTML・CSS実践!

ブロック要素とインライン要素とは？

- **ブロック要素**⇒見出し、段落など文書を構成する基本要素
- **インライン要素**⇒主としてブロックレベル要素の内容として用いられるもの



お弁当箱・仕切りが**ブロック要素**
食べ物が**インライン要素**

HTML・CSS実践!

ブロック要素とインライン要素とは？

ではここで実際にどういったものになるのかご紹介していきます。

基本的な形をこれから紹介していきます。

index.html

```
<div class=”box”>
</div>
<!-- これはブロック要素 -->
※CSSでレイアウトしていきます
```

ブロック要素とインライン要素補足

ブロック要素とインライン要素とは？

さて、ブロック要素とインライン要素についてはいまいちピンと来ないかもしれません。
まずは基礎的な意味と現場で使われている
ベーシックな「パターン」を理解することから始めましょう。

例　オーソドックスな例

index.html

```
<div class="outer-box">
    <p>テキスト</p>
    <a href="">gs</a>
</div>
```

※divを「上手に活用」して綺麗に収納していくイメージです
※またこの収納の別の言い方としてwrap（ラップ）すると言ったりもします（人によっては言わないかも）

ブロック要素とインライン要素補足

例　divを「お弁当の箱」のように活用して上手に領域を作る

index.html

```
<div class="outer-box">
  <div class="in-box">
    <p>テキスト1</p>
    <p>テキスト1</p>
  </div>
  <div class="in-box">
    <p>テキスト1</p>
    <p>テキスト1</p>
  </div>
</div>
```

※一番外側のdivの中で領域を作り、さらにその中にdivを上手に使って収納していく

※divの所でcssのスタイルを与えることで綺麗にコーディングができる

ブロック要素とインライン要素補足

あまり推奨されない例と現場でないパターン

例　pタグで領域を確保するパターン

index.html

```
<p class="outer-box">
  <div class="in-box">
    <p>テキスト1</p>
    <p>テキスト1</p>
  </div>
  <div class="in-box">
    <p>テキスト1</p>
    <p>テキスト1</p>
  </div>
</p>
```

※<p>タグも同じブロック要素だが、「領域を作る」際はdivを使うのが一般的（現場でも推奨）

※html5になってからは[インライン要素]でも「領域」を作っても良いがdivを使うのが一般的

ブロック要素とインライン要素補足

あまり推奨されない例と現場でないパターン

例　aタグ,spanタグで領域を確保するパターン

index.html

```
<a class="outer-box">
  <span class="in-box">
    <p>テキスト1</p>
    <p>テキスト1</p>
  </span>
  <span class="in-box">
    <p>テキスト1</p>
    <p>テキスト1</p>
  </span>
</a>
```

※<a>タグはlink機能（画面遷移）なため、ワードプレスやnewsサイトでよく見かける例
　ただ、これは使い方をよく考えて利用する
※タグなどでラップすることもあるが、ワードプレスなどで見かけるパターンが多いが
divを使うのが一般的　　※ワードプレス等は授業で解説します

ブロック要素とインライン要素補足

あまり推奨されない例と現場でないパターン

例　全部をdivにするパターン

index.html

```
<div class="outer-box">
  <div class="in-box">
    <div>テキスト1</p>
    <div>テキスト1</p>
  </div>
  <div class="in-box">
    <div>テキスト1</p>
    <div>テキスト1</p>
  </div>
</div>
```

※<divタグで全てラップするコーディングは基本的には行いません。
　理由はSEOの関係があります。授業で解説します。
※きちんと「正しいタグ」を[使い分けて]コーディングすることでSEOの評価が上がります。
　（これを内部SEOといいます）

HTML・CSS実践!

演習1

ではここで実際にパソコンで作業を進めていきましょう！

1.今からモニターに書き出すことをコーディングしてください

2.できた方は挙手して、先生にチェックしてもらってください

flexboxレイアウト

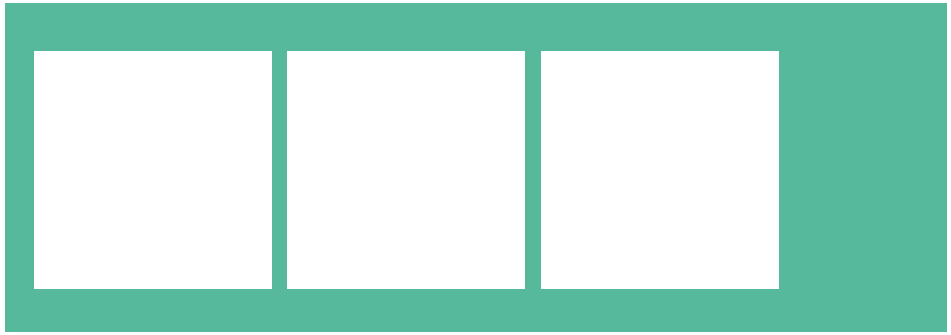
flexboxとは

flexboxとは横並びにしたいときに活用する「レイアウト」テクニックの一つです。

他にも様々なテクニックがありますが、今はflexboxが活用されています。

webブラウザ

フレックスボックスを使えば「横並び」になる！



※ただし、ただ使うだけだとこの状態

flexboxレイアウト

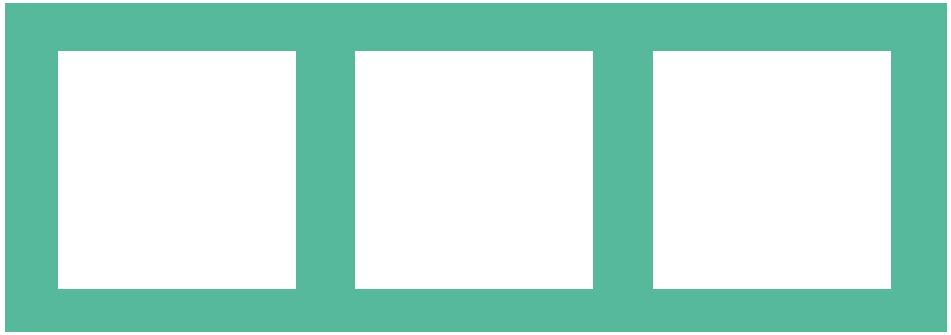
横並びをマスターしよう

flexboxをただ使うだけではレイアウトができません。

そこで次に登場するのが「**justify-content**」です！

webブラウザ

justify-contentを使うと例えば↓のようになる



※他にもたくさんの変化があるのでチェック！

flexboxレイアウト

演習2

ではここで実際にパソコンで作業を進めていきましょう！

1.ブロック要素を作り[**display:flex**]を使って横並びにする

2. **justify-content** を使って動きをチェックしよう

3.ブラウザで一緒にチェックをする

flexboxレイアウト

縦並びを理解しよう

flexboxを使うことで「横方向」だけではなく「縦方向」もレイアウトすることができます！その方法をチェックしよう！

webブラウザ

align-itemsによる変化

※上下の並びを整えることができる

flexboxレイアウト

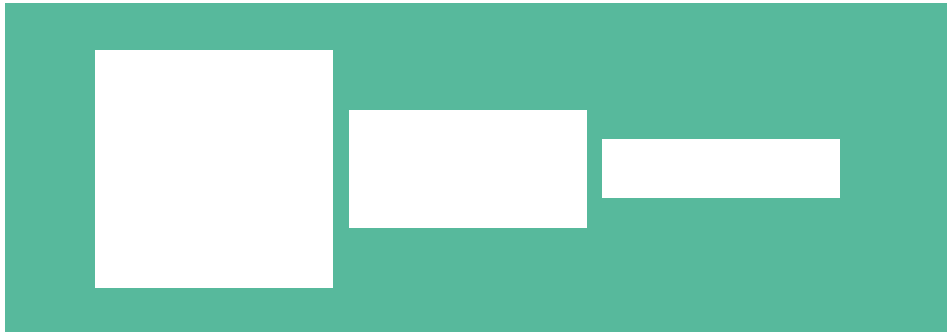
縦並びを理解しよう

縦方向を制御する方法は「**align-items**」を活用する！

例えば、「**align-items:center**」を使うと↓のようになる

webブラウザ

align-itemsによる変化



※他にもたくさんの変化があるのでチェック！

flexboxレイアウト

演習3

ではここで実際にパソコンで作業を進めていきましょう！

1.ブロック要素を作り[display:flex]を使って横並びにする

2. align-items を使って動きをチェックしよう

3.ブラウザで一緒にチェックをする

課題　資料課題を取り組む

今までのおさらいをしよう

flexboxを使うことで「レイアウト」を自由自在に操ることができるようになります！
今日やってきたことの応用として課題フォルダの問題を全て回答しよう！

必須条件!

1.配布されている当日課題資料を行う

2.わからない箇所は、質問または動画をチェックする

3.今日できなかった人は宿題。提出日は別途伝えます。