

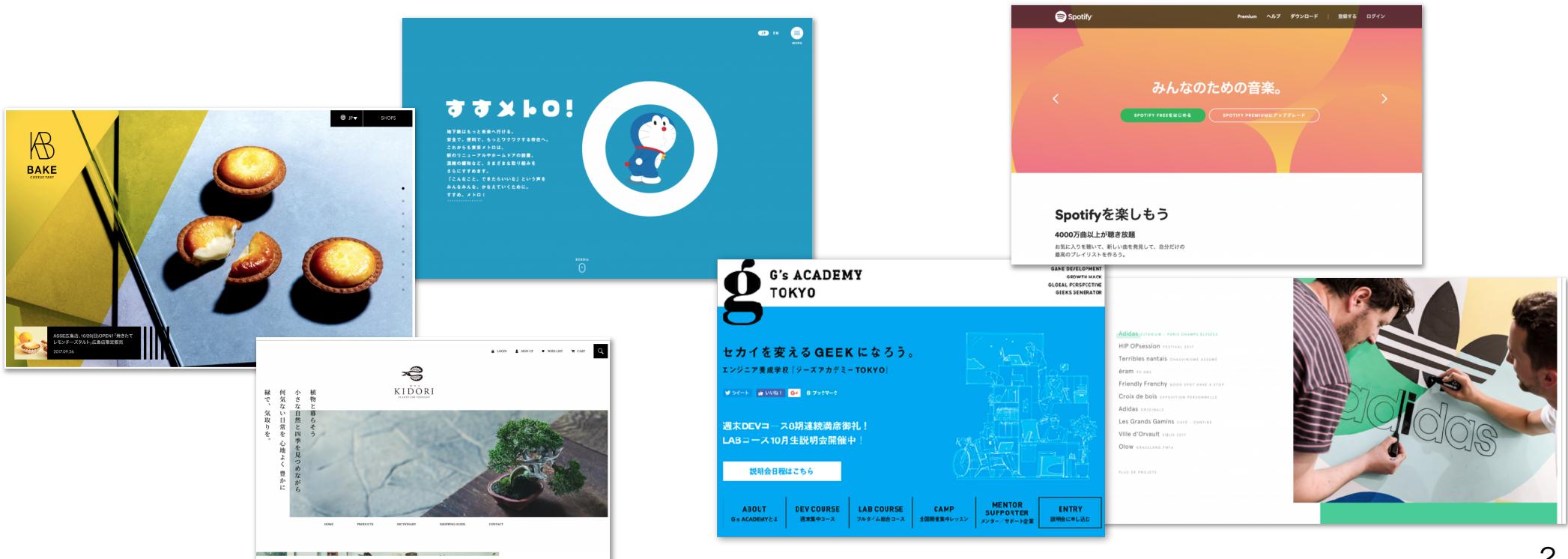
HTML,CSSの基礎

WEBページを作れるようになろう

HTMLとCSSに触れよう！

さあ、プログラミングをはじめよう！

WEBページは「HTML」と「CSS」という言語を組み合わせることによって作られています。この2つを学ぶことで↓のようなサイトを作ることができます。



HTMLとCSSに触れよう！

HTMLの仕組みを理解しよう

まずは基本である「HTML」を学びましょう。

HTMLでは「タグ」と呼ばれるものを使います。

図のように「タグ」で囲むことで「HTML」として認識されるようになります。

index.html

<**h1**>はじめてのプログラミングへようこそ</**h1**>



タグで囲むことが必須

<**p**>はじめてのプログラミングへようこそ</**p**>

HTMLとCSSに触れよう！

開始タグと終了タグを理解する

文章（テキスト）を入力する際は必ず「タグ」で囲むことが必須です！

最初のタグを**開始タグ**、終わりのタグを**終了タグ**といいます。

終了タグ「/」※スラッシュ

index.html

<**h1**>はじめてのプログラミングへようこそ</**h1**>



タグで囲むことが必須

<**p**>はじめてのプログラミングへようこそ</**p**>

HTMLとCSSに触れよう！

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. 「はじめてのプログラミングスタート」という文章を入力する
2. 開始タグと終了タグに<h1>を使う
3. ブラウザと一緒にチェックをする

タグの種類

見出しタグとは？

タグには様々なものがありますが、

基礎となる<h1>,<h2>,<h3>…

..といった見出しタグについて理解を深めましょう。

見出しタグのことを**heading**といいます。

index.html

```
<h1>これはテキストです</h1>
<h2>これはテキストです</h2>
<h3>これはテキストです</h3>
<h4>これはテキストです</h4>
<h5>これはテキストです</h5>
```

タグの種類

段落のタグとは？

段落を表すものは

タグを使います。

pは”paragraph”の意味の略になります。

index.html

```
<p>プログラミングを学習しよう</p>
<p>pタグはよく使うので覚える</p>
```

タグの種類

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. 「見出しタグを利用する」を<**h1**>を使って囲む
2. 「段落タグを利用する」を<**p**>を使って囲む
3. ブラウザで一緒にチェックをする

画像を利用しよう

画像を表示させるには？

WEBサイトでは画像を利用するシーンがほとんどです！

画像を利用するタグは<**img**>タグを利用します。

index.html

```

```

画像のURLが入ります

画像を表示させる

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. 好きな画像をグーグルやヤフーから探そう
2. 「**img**」タグを利用して画像を表示させてみよう
3. ブラウザと一緒にチェックをする

HTMLでのCSSの書き方 (classとid)

class編

xxxに好きな名前をつける

```
<div class="xxx">ここにテキストが入ります</div>
<p class="xxx">ここにテキストが入ります</p>
<h1 class="xxx">ここにテキストが入ります</h1>
```

例

```
<div class="a">ここにテキストが入ります</div>
<p class="b">ここにテキストが入ります</p>
<h1 class="c">ここにテキストが入ります</h1>
```

HTMLでのCSSの書き方 (classとid)

id編

xxxに好きな名前をつける

```
<div id="xxx">ここにテキストが入ります</div>
<p id="xxx">ここにテキストが入ります</p>
<h1 id="xxx">ここにテキストが入ります</h1>
```

例

```
<div id="aa">ここにテキストが入ります</div>
<p id="bb">ここにテキストが入ります</p>
<h1 id="cc">ここにテキストが入ります</h1>
```

CSSファイルへの書き方

classの場合

[.](ドット)を先頭につけます

例) css名がaのclassの場合

```
.a{  
    color: red;  
}
```

idの場合

[#](ハッシュ)を先頭につけます

例) css名がbのidの場合

```
#b{  
    color: blue;  
}
```

point

最初のうちは[class][id]のどちらを使うべきかと悩んだら[class]を使ってください
※授業で解説します

CSSの注意事項

名前の付け方に関して

class,idは好きな名前でもよい。

ただし「数字から始めるのはNG」です。

それ以外なら基本的にどんな名前をつけても問題ありません。

idの注意

htmlファイル内で必ず
「1つだけ」しかつけてはいけない

※画面で実際に説明します

CSSの注意

CSSの命令の後は
必ずセミコロン[;]
を忘れない！

よく使うCSSを実際に使ってみよう

- background(バックグラウンド) -color -image -repeat -position -attachment -clip -size -origin;

背景画像、背景色などを指定できる命令です。実際にはショートバインドと言ってまとめて書くことができます。

使い方

背景色をつける場合

```
.a{  
background: red(好きな色で大丈夫);  
}
```

背景画像を使う場合

```
.a{  
background: url(urlを貼り付けます) no-repeat center center;  
height: ○○;  
}
```

※no-repeatやcenter centerなどまた諸注意については授業で解説します。

よく使うCSSを実際に使ってみよう

- **text-align(テキストアライン) -left -right -center**

テキストの表示位置（正式にいうと行の揃え位置）

使い方

文字を右寄せにしたい場合

```
.a{  
    text-align: right;  
}
```

文字を中心寄せにしたい場合

```
.a{  
    text-align: center;  
}
```

※何も設定しないときはそもそも「左並び」になっています。

CSSに触れよう！

よく使うCSS

これをみなさん調べてどんな挙動でどんな変化がするのかを、エディターに書いて提出してください。

style.css

width
height
margin
padding
border

line-height
text-shadow
box-shadow
border-radius
opacity

cursor
letter-spacing
font-weight
font-size
hover

課題 自己紹介サイトを作る

今までのおさらいをしよう

WEBサイトは極端に言えば、今お伝えした内容をうまく組み合わせて
サイトを作っていくます。

ここで今までのおさらいとして自己紹介サイトを作りましょう！

必須条件！

- 1.見出しタグを利用する
- 2.画像を表示させる
- 3.自分の趣味や「知ってほしいこと」を3つ以上アピールする！！
- 4.CSSで何かしらの変化をつける(スタイルをつける)

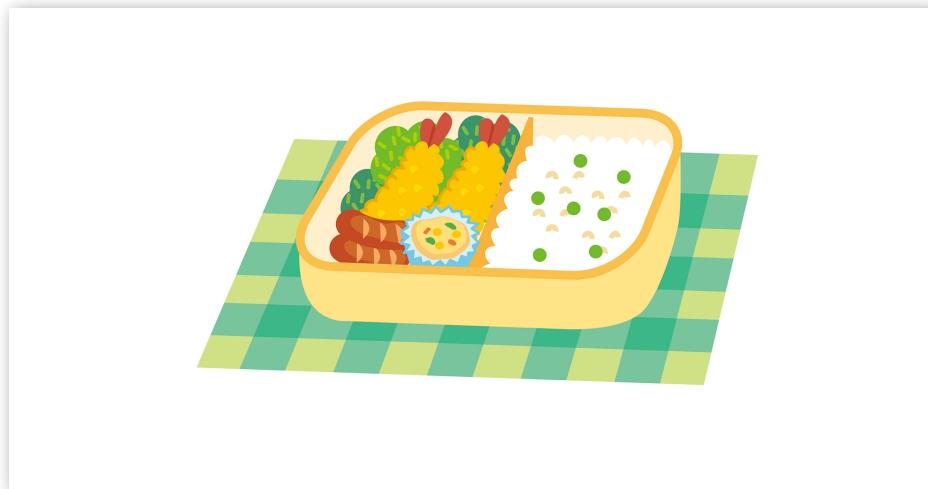
HTML,CSSの実践

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

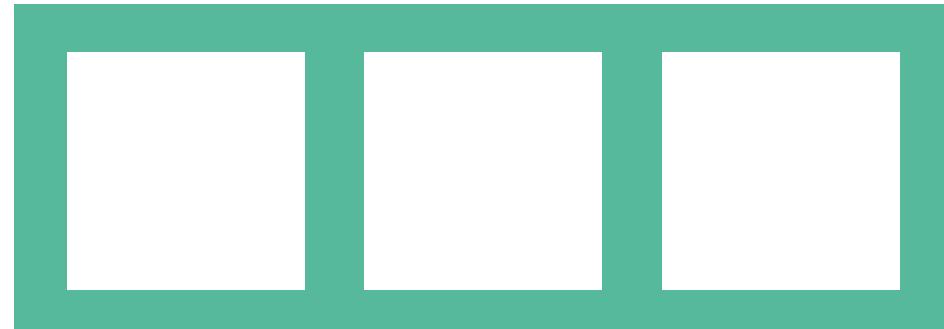
横並びをマスターしよう

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

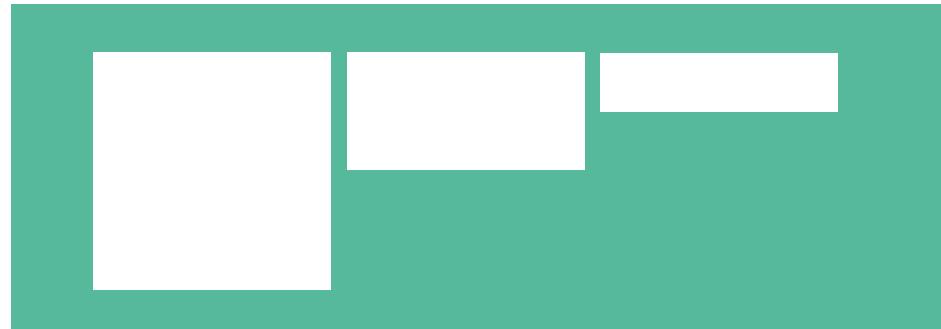
flexboxレイアウト

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

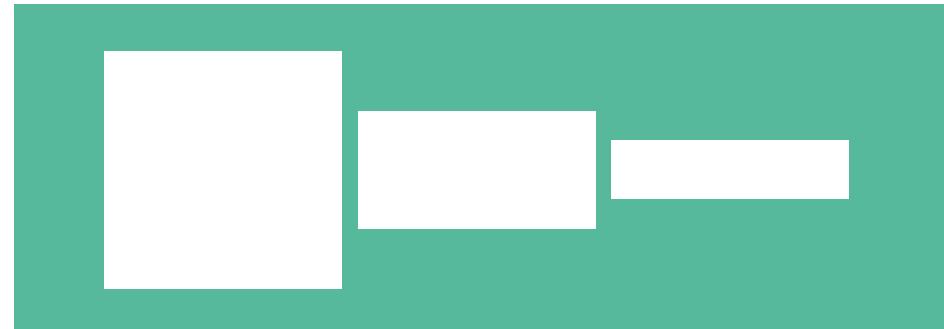
縦並びを理解しよう

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。