

CSSレイアウト基礎実習

レイアウトと初期化

株式会社ジードライブ

本講義のトピック

- レイアウトの構築
- ディベロッパーツールの利用
- CSSの初期化

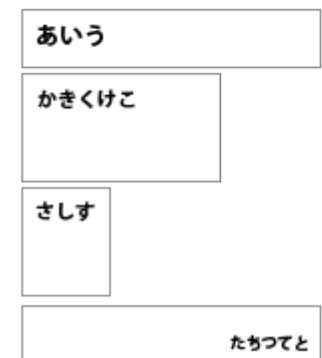
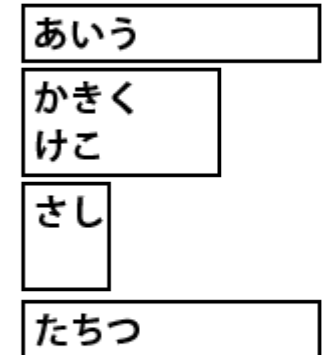
レイアウトの構築

レイアウトの構築手順

- 基本的なレイアウト構築の流れ

- ① テキストを文章としてマークアップ
- ② 情報・役割ごとに**ボックス**を作成
 - ・ HTMLでのグルーピング
 - ・ CSSで幅や高さを設定
- ③ ボックス内での見た目を整える
 - ・ 文字の大きさ、余白、境界線など

<h1>あいう</h1>



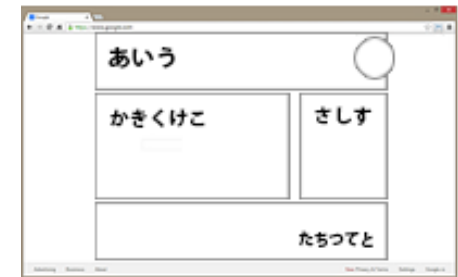
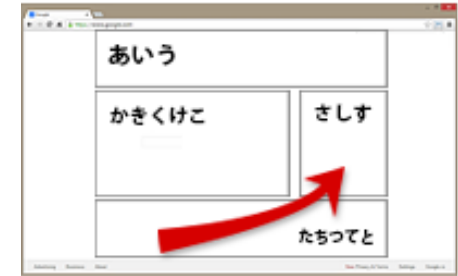
レイアウトの構築手順

- ④ ボックスを適切な位置に配置
 - ・ 大まかな位置調整
- ⑤ 細かいパーツの位置調整
- ⑥ 完成に向けたデザインの調整

レイアウト ≡ ボックスの配置



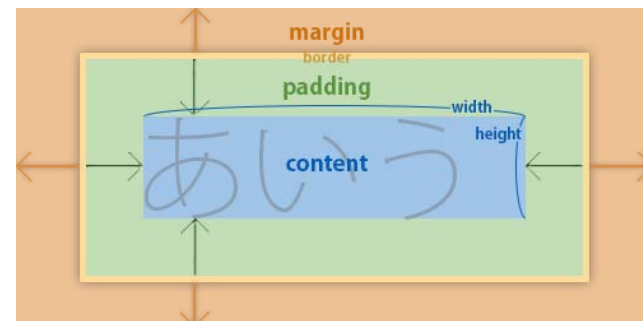
ボックスモデルの理解が大切



ボックスモデルとは

- 要素を矩形の領域（ボックス）で捉える概念
 - ボックスは、内容領域・パディング・ボーダーから成る
 - ボックス同士の間隔をマージンという


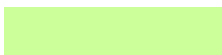
用語	説明
内容領域 Content Area	要素を表示するための領域。width, height プロパティで任意の幅と高さを指定可能
padding	内容領域とborderの間の余白。背景色や背景画像が表示される
border	ボックスの境界線。太さ、線種、色を指定することができる
margin	他のボックスとの間隔



詳しくは次のチャプターで学習する

ディベロッパーツールの利用

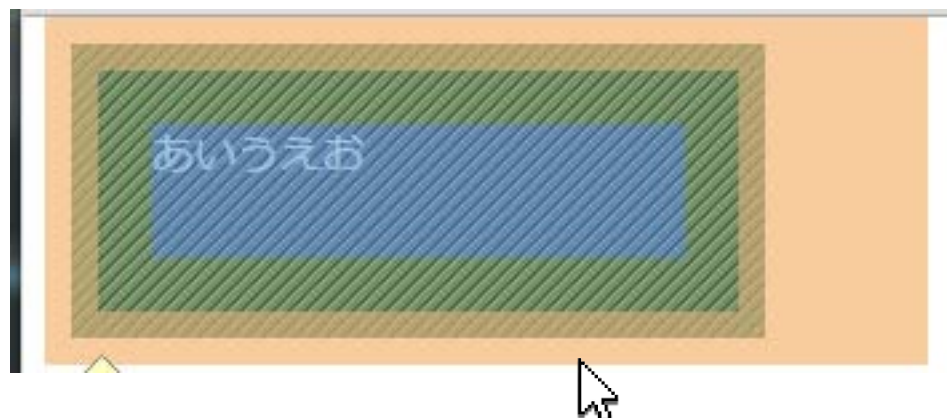
ツールの利用

- ディベロッパーツールを利用することで
ボックスの構造を視覚的に把握できる
 - ① Chromブラウザをプレビューした状態でF12キー
 - ② Ctrl + Shift + Cで「虫眼鏡ツール」の起動状態に。
 - ③ 画面上のボックスをオンマウスすると、
外側余白のmarginがオレンジ 
内側余白のpaddingがグリーン 
で表示される。

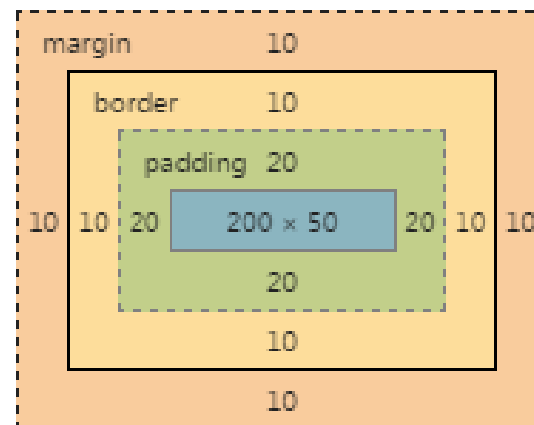
※クリックした場合は、もう一度、虫眼鏡ツールを起動する
※F12キーを省略して最初から虫眼鏡ツールの起動もできる。

ツールの利用

- ボックスモデルをブラウザ上で確認する



レイアウトがうまく反映されないとき
このツールで余白やプロパティ、クラス
名などを検証し、問題点を見つけ出す



ボックスをクリック
すると設定した数値
が右下に表示される

※ Chrome以外のブラウザにも開発者向けのツールは用意されている

CSSの初期化

初期化とは

- ブラウザには、デフォルトのCSSが設定されている
 - デフォルトのCSSはブラウザによって異なる
- ⇒ レイアウトしづらいので、最初にフラットな状態にする

初期化の考え方

名称	説明
*	全称セレクタを使用して全ての要素に同じ値を適用
リセット	主要なプロパティの値をフラットに戻す
ノーマライズ	元々の値でも必要な設定はそのまま利用
サニタイズ	ノーマライズCSSをベースにモバイル対応を追加

参考： [リセットCSSフレームワークのメモ](#)

ブラウザのもつCSSの確認

- 元々ブラウザの持っているデフォルトCSSをChrom DevToolで確認する(Ctrl+shift+C)

h1のCSS

HTML書式

<h1>タイトル</h1>

ブロックボックス
文字サイズ2倍の32px
外側余白上下に21.44px
太字

```
html  body  h1
Styles  Event Listeners  DOM Breakpoints  Properties
Filter
element.style {
}
h1 {
  display: block;
  font-size: 2em;
  -webkit-margin-before: 0.67em;
  -webkit-margin-after: 0.67em;
  -webkit-margin-start: 0px;
  -webkit-margin-end: 0px;
  font-weight: bold;
}
```

ブラウザの持つ
デフォルトCSS

ブラウザのもつCSSの確認

HTML書式

```
<p>文章</p>
<ul>
  <li>リスト</li>
</ul>
```

ブロックボックス
外側余白上下に16px

```
p {
  display: block;
  -webkit-margin-before: 1em;
  -webkit-margin-after: 1em;
  -webkit-margin-start: 0px;
  -webkit-margin-end: 0px;
}
```

user agent stylesheet

文章



ul 293px × 24px

```
ul, menu, dir {
  display: block;
  list-style-type: disc;
  -webkit-margin-before: 1em;
  -webkit-margin-after: 1em;
  -webkit-margin-start: 0px;
  -webkit-margin-end: 0px;
  -webkit-padding-start: 40px;
}
```

user agent stylesheet

リストデザイン黒丸
内側余白左に40px

全称セレクタによる初期化

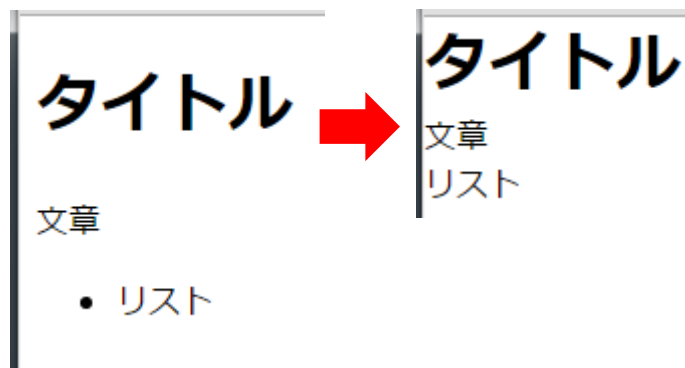
- 全ての要素に対し、同じ値をまとめて設定する

HTML書式

```
<h1>タイトル</h1>
<p>文章</p>
<ul>
  <li>リスト</li>
</ul>
```

CSS書式

```
* {
  padding: 0;
  margin: 0;
}
```



※余白は左上に全て詰まるが、
文字の設定はそのまま残っている

※以前は全要素に適用することでパフォーマンスの負荷が懸念されていたが、検証で影響がないことが示唆されている。

リセットCSSによる初期化

- 主要なプロパティの値をフラットに戻す
- 外部のCDNを利用することも多い

代表的なリセットCSS

- [Eric Meyer's Reset CSS 2.0](#)
- [HTML5 Doctor CSS Reset](#)
- [Yahoo! \(YUI 3\) Reset CSS](#)

※YUI3はHTML5の新要素に対応していない

※より細かくリセット設定をするのであればHTML5 Doctor

リセットCSSによる初期化

- HTML Doctor CSS Resetの利用

[HTML5 Doctor CSS Reset](#)

HTML書式

```
<head>
  <link rel="stylesheet" href="html5doctor.css">
</head>
<body>
  <h1>タイトル</h1>
  <p>文章</p>
  <ul>
    <li>リスト</li>
  </ul>
```

タイトル

文章

• リスト



タイトル
文章
リスト

※余白がなくなり
見出しサイズも
標準にリセット

ノーマライズCSSによる初期化

- デフォルトCSSを活かし、ブラウザ間の誤差補正(ノーマライズ)だけを行うという考え方

[Normalize.css](#)

HTML書式

```
<head>
  <link rel="stylesheet" href="normalize.css">
</head>
<body>
  <h1>タイトル</h1>
  <p>文章</p>
  <ul>
    <li>リスト</li>
  </ul>
```

タイトル



タイトル

文章

• リスト

文章

• リスト

※見た目の変化無し

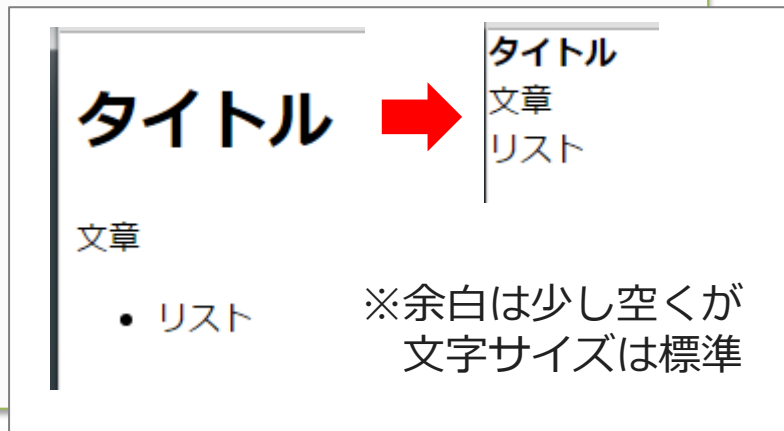
サニタイズCSSによる初期化

- ブラウザ間の誤差補正を行い、基本スタイルをリセット
- box-sizingを使用しているのでpaddingの差し引き計算をしなくてよい
- スマホの文字サイズ自動縮小を回避
- 全称セレクタを多用

sanitize.css

HTML書式

```
<head>
  <link rel="stylesheet" href="sanitize.css">
</head>
<body>
  <h1>タイトル</h1>
  <p>文章</p>
  <ul>
    <li>リスト</li>
  </ul>
```



Boilerplateの利用

- Boilerplate 【ボイラープレート】とは、Webサイト制作時に使い回すことのできる**ひな型**のこと
- サイト制作にあたっては、CSSの初期化以外にも、様々な共通の作業を行う。それを前提としたファイル群をあらかじめ用意し、ひな型として使いまわすことで、作業を効率化することができる

[html5 Boilerplate](#)

[Web Starter Kit](#)

練習問題

- 練習01 を行う