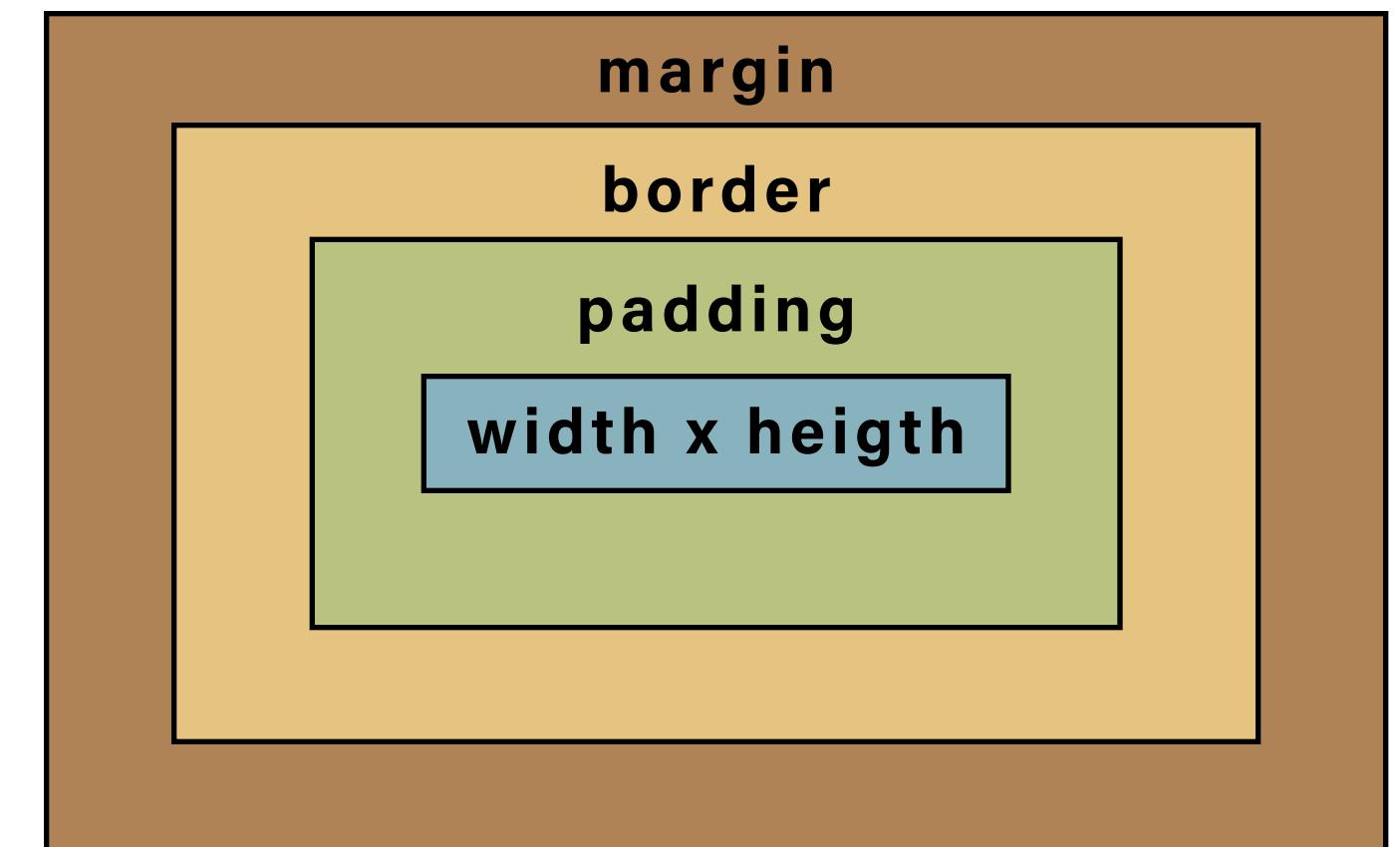


# Web プログラミング

第五回 : CSS(2)

## CSS(2) - ボックスモデルについて

「ボックスモデル」は、ブロック要素の幅や余白などを定義する時の基本です



## CSS(2) - 幅と高さ

**width** と **height** を指定して、ブロックの幅と高さを指定できる

```
p {  
    width: 400px;  
    height: 200px;  
}
```

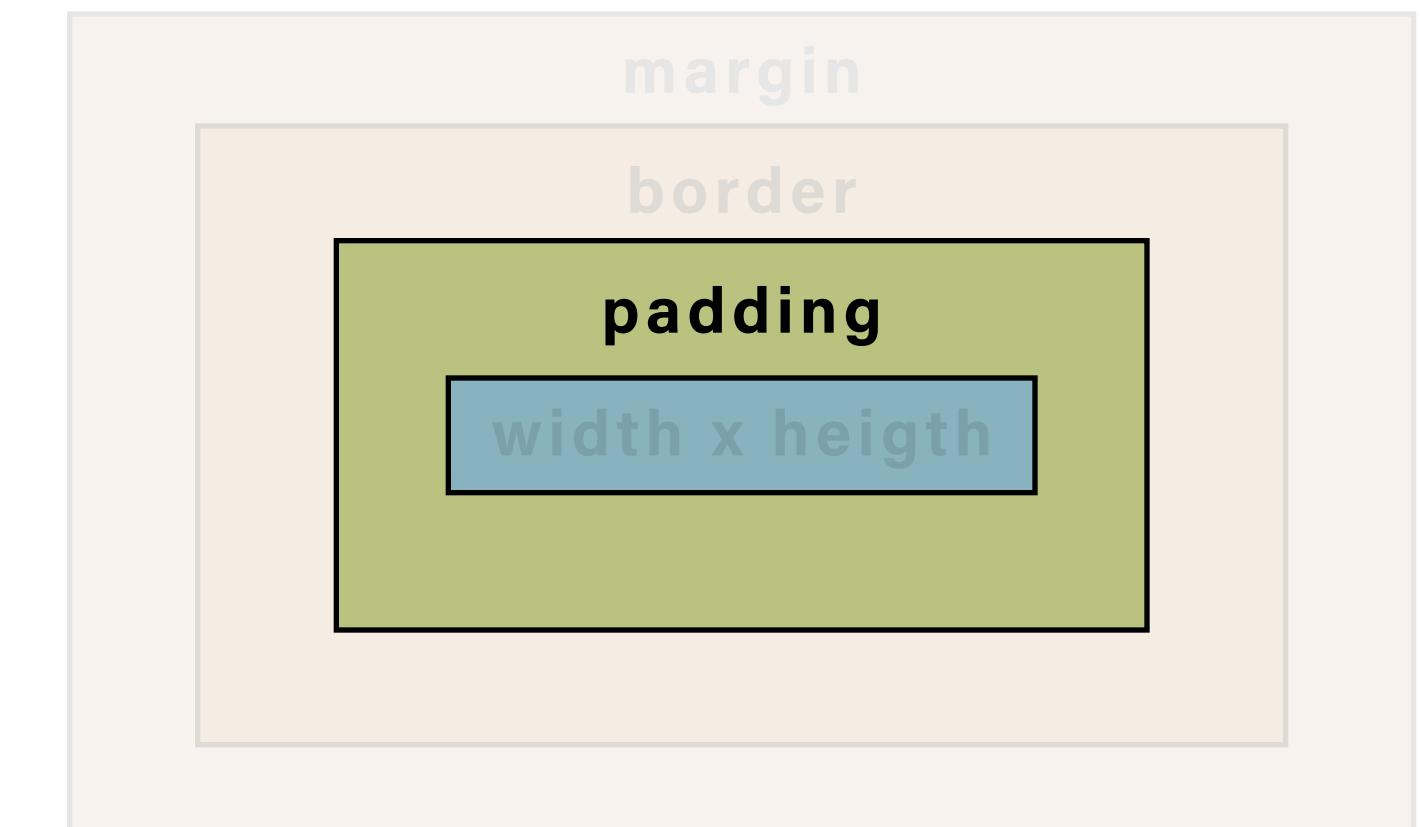


## CSS(2) - パディング

**padding** を指定して、ブロックの内側の余白を指定できる

```
p {  
    padding-top: 10px;  
    padding-right: 20px;  
    padding-bottom: 30px;  
    padding-left: 40px;  
}
```

```
p {  
    padding: 10px 20px 30px 40px;  
}
```

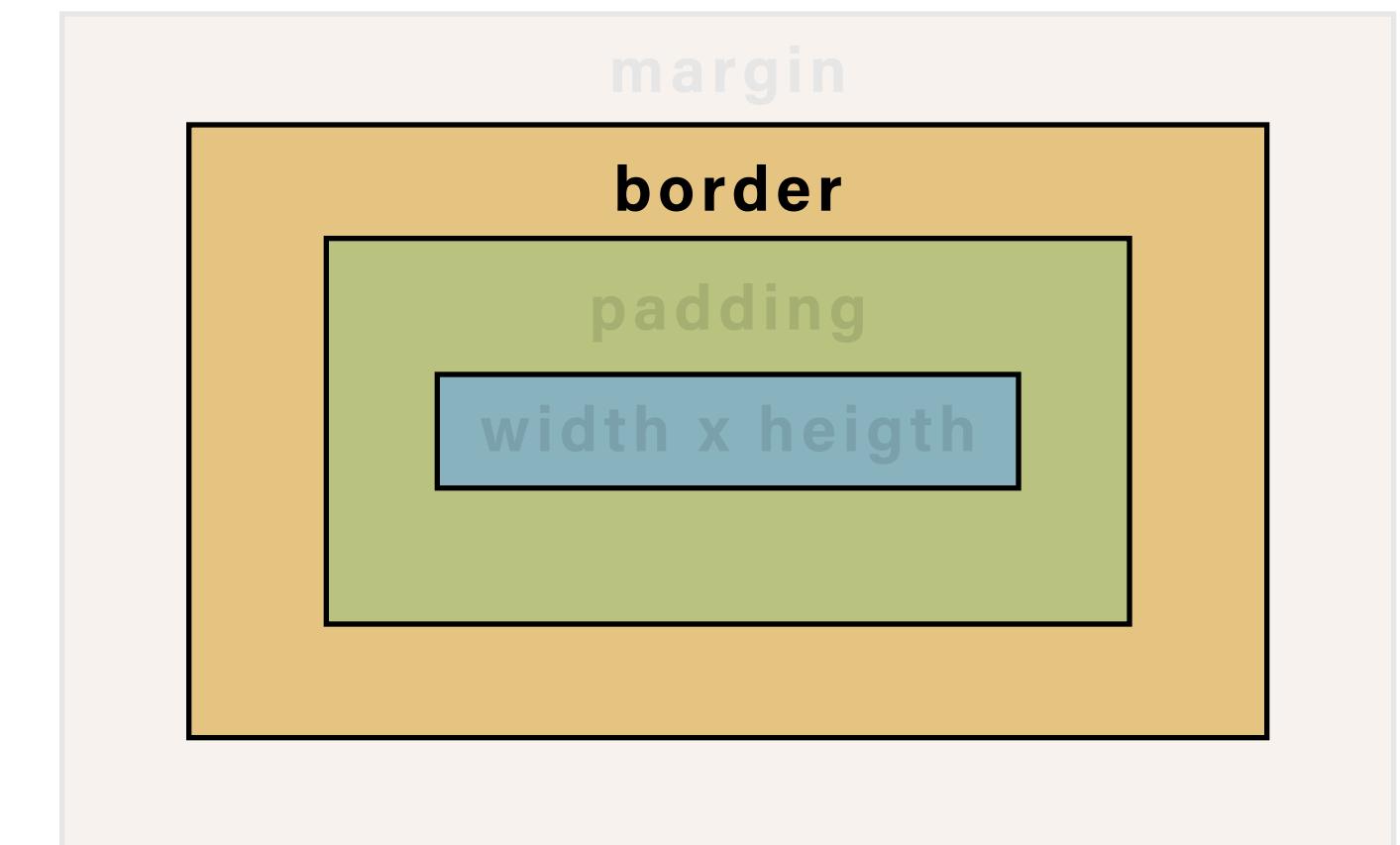


## CSS(2) - ボーダー

**border** を指定して、ブロックに枠線を付与できる

```
p {  
    border: 1px solid #000;  
}
```

```
p {  
    border-top: 1px solid #000;  
    border-right: 1px solid #000;  
    border-bottom: 1px solid #000;  
    border-left: 1px solid #000;  
}
```

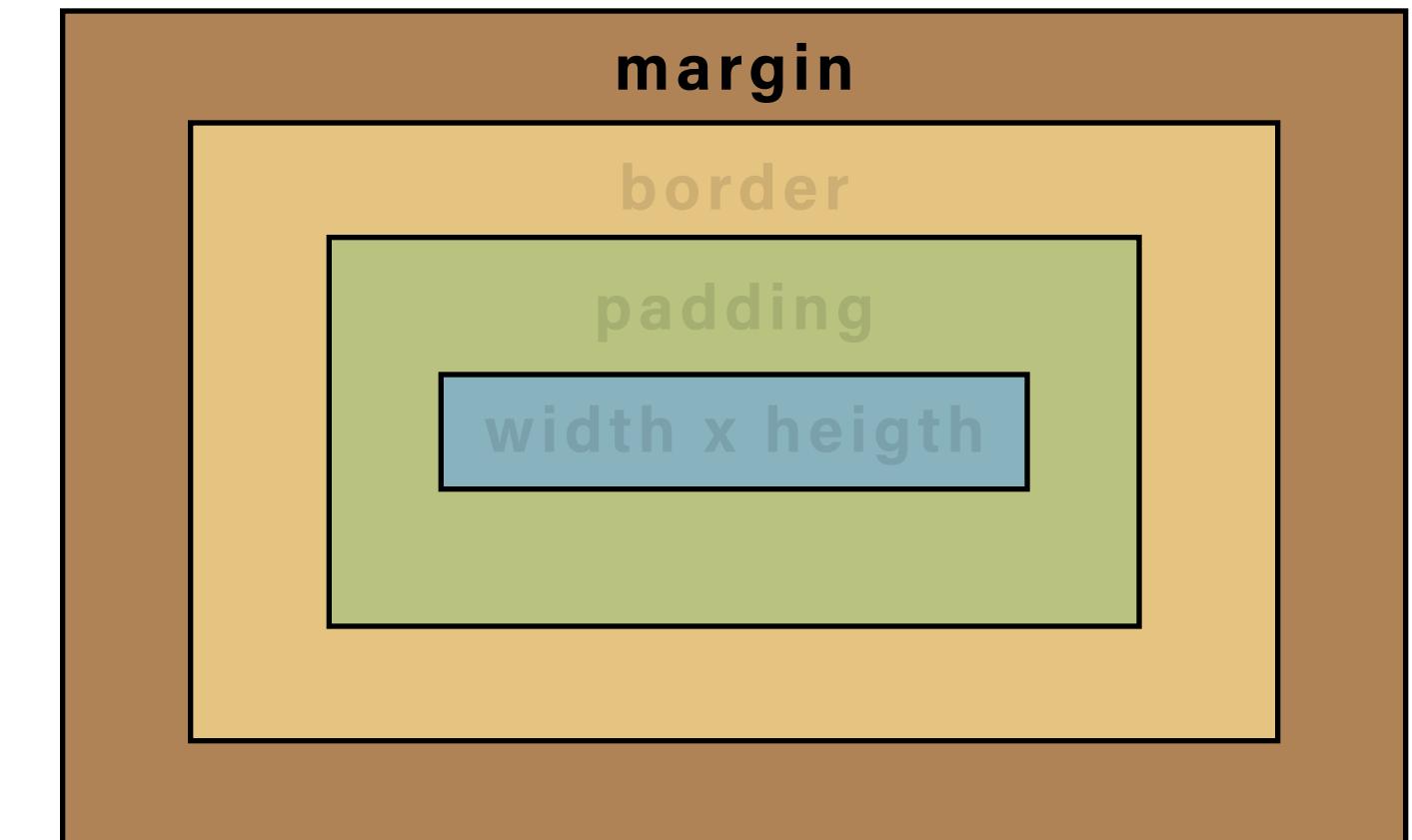


## CSS(2) - マージン

**margin** を指定して、ブロック外側の余白（他要素とのスキマ）を指定できる

```
p {  
    margin-top: 10px;  
    margin-right: 20px;  
    margin-bottom: 30px;  
    margin-left: 40px;  
}
```

```
p {  
    margin: 10px 20px 30px 40px;  
}
```



## CSS(2) - ボックスのサイズ計算

最終的なボックスの幅と高さは、  
**width/height + padding + border** のサイズになる

例 : **width:500px** 、 **padding:50px** 、 **border: 30px** の場合  
最終的な幅は **660px** になる



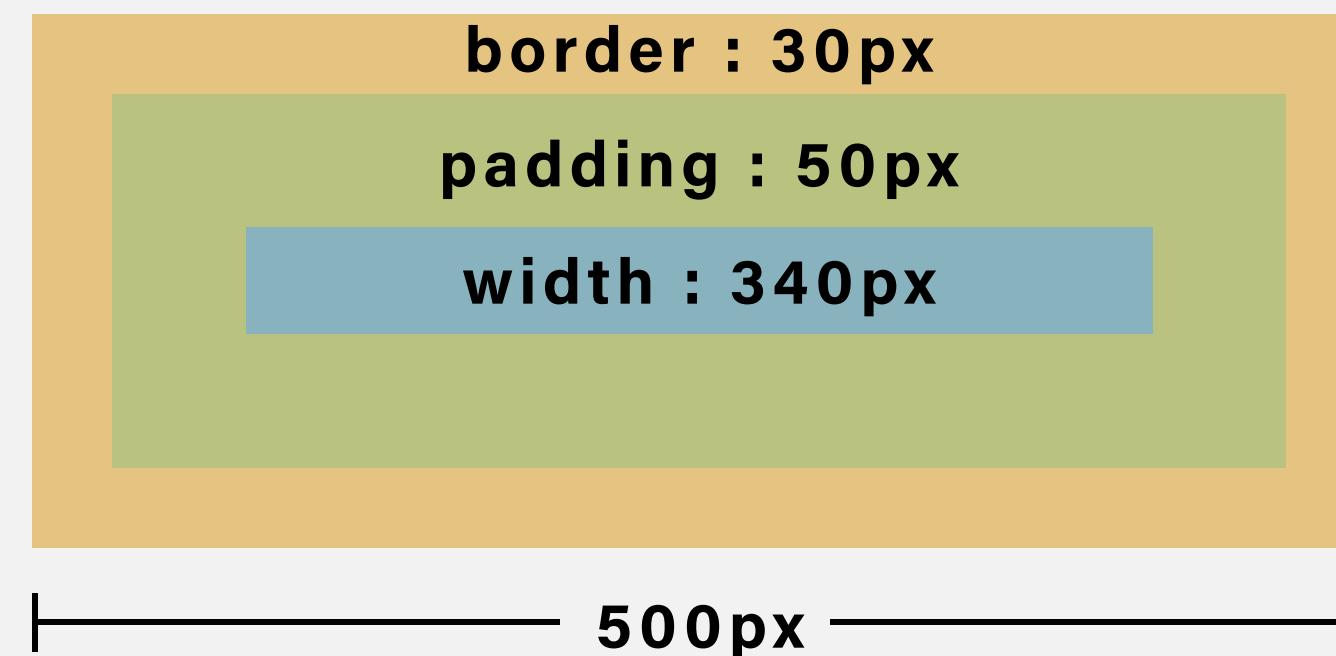
**margin** は、他要素とのスキマなので、ボックスの幅と高さには**含まない**

## CSS(2) - ボックスのサイズ計算

css で **box-sizing: border-box** を指定すれば、  
**width/height** に padding と border を含めたサイズにできる

```
p {  
  box-sizing: border-box;  
}
```

例 : **width:500px** 、 **padding:50px** 、 **border: 30px** 、 **box-sizing:border-box** の場合  
最終的な幅は 500px になり、コンテンツエリアの幅は 340px になる



## CSS(2) - インライン要素におけるボックスモデル

インライン要素では、ボックスモデルの扱いは特殊になる

- **width** と **height** は指定しても無視される
- **padding** は指定できるが、上下の **padding** はレイアウトに影響しない
- **border** は指定できる
- **margin** は左右のみ有効で、上下は無視される

インライン要素では、幅や高さなどは基本考えない方が無難

## CSS(2) - インライン要素をブロックレベルにする

a 要素など、インライン要素をブロック要素のように扱いたいときがある。

css で `display: block` を設定すると、ブロック要素のように取り扱えるようになる。

```
a {  
    display: block;  
}
```

逆に、ブロック要素をインライン要素のようになるとこもできる

```
p {  
    display: inline;  
}
```

## CSS(2) - 練習問題

### 練習問題

a 要素を「ボタン」にしてみよう



ボタン

- ・幅 300px くらい
- ・上下パディング 15px くらい
- ・枠線 2px、黒色 (#000000)
- ・文字色は白色 (#FFFFFF)
- ・フォントサイズは 24px くらい
- ・テキスト中央揃え
- ・背景色グレー (#B3B3B3)

## CSS(2) - 色の指定方法

色の指定方法は、ざっくり 3 種類あります

キーワード指定

```
p {  
    color: red;  
}
```

カラーコード指定

```
p {  
    color: #FF0000;  
}
```

関数指定

```
p {  
    color: rgb(255,0,0);  
}
```

## CSS(2) - 色の指定方法

「キーワード指定」は、色の名前を指定する。

**red**、**blue**、**green**、**gray**、**yellow** など  
色の名前はたくさんあります。

参考：<https://developer.mozilla.org/ja/docs/Web/CSS/named-color>

## CSS(2) - 色の指定方法

「カラーコード指定」は、# から始まる数値です。

赤緑青の 3 つの原色の量を指定し、  
それらを加法混色で混ぜた色を表示します。

```
color: # FF FF FF ;
```

赤緑青の各原色は十六進数で表記します。

最小は 00 (十進数では 0)、  
最大は FF (十進数では 256) です。

## CSS(2) - 色の指定方法

カラーコードの例：

白 : # FF FF FF

赤 : # FF 00 00

緑 : # 00 FF 00

青 : # 00 00 FF

黒 : # 00 00 00

グレー : # 80 80 80

黄色 : # FF FF 00

マゼンダ : # E4 00 7F

ブラウン : # 73 4E 30

参考：googleで「カラーピッカー」で検索

## CSS(2) - 色の指定方法

「関数指定」は、色空間も含めた関数の形で指定できます。  
RGB の場合、各値は十進数の 0～256 の間で指定します。

RGB 色空間で指定する場合：

```
color: rgb( 255, 255, 255 );
```

HSL 色空間で指定する場合：

```
color: hsl(50 80% 40%);
```

## CSS(2) - 色の指定方法

ちなみに：  
アルファ値を最後に追加して、透明度をすることもできます。

```
color: # FF 00 00 80 ;
```

```
color: rgb( 255, 255, 255, 0.5 ) ;
```

## CSS(2) - レイアウトを意識した HTML 構成



Web デザインを実装するためには、  
最終的なレイアウトを意識しながら  
HTML 構成を記述する必要があります。



このページでは、おいしい唐揚げのレシピを紹介します。ぜひ作ってみてください。

### 材料

鶏もも肉	250g
しょうゆ	大さじ1 ※
酒	大さじ1 ※
おろししょうが	小さじ1/2 ※
おろしにんにく	小さじ1/2 ※
片栗粉	大さじ3
サラダ油	適量

## CSS(2) - レイアウトを意識した HTML 構成

材料	量
鶏もも肉	250g
しょうゆ	大さじ1
酒	大さじ1 ※
おろししょうが	小さじ1/2 ※
おろしにんにく	小さじ1/2 ※
片栗粉	大さじ3
サラダ油	適量

### メインコンテンツ

このページでは、おいしい唐揚げのレシピを紹介します。ぜひ作ってみてください。

このようなデザインであれば、  
左のようにブロックを分割できそう

## CSS(2) - レイアウトを意識した HTML 構成



さらに詳細に分割すると、  
このような構成になる。



コンテンツ・レイアウトを  
ブロックごとに区切り、  
入れ子構造をとると、実装しやすくなる

## CSS(2) - 「ヘッダー」のレイアウト



レシピのページ

幅 100%

高さ : 成り行き

背景赤

テキスト白色、太字、18px

テキスト中央揃え

上下のパディング

## CSS(2) - 「ファーストビュー」のレイアウト

### ファーストビュー



ファーストビュー

幅 : 100%

高さ : 成り行き

背景に画像

下にパディング

見出しテキストエリア

幅 : 100%

高さ : 成り行き

テキスト白色、太字、24px

上下に大きなパディング

調理情報エリア

幅 : 680px

高さ : 成り行き

中央揃え

白背景

上にマージン

## CSS(2) - ボックスの中央揃え

幅が小さいボックスを中央に配置したい時は、  
左と右のマージンに **auto** を指定する

```
.center-box {  
    width: 680px;  
    margin-left: auto;  
    margin-right: auto;  
}
```



# CSS(2) - 「メインコンテンツ」のレイアウト

## メインコンテンツ

This block represents the main content area. It contains a header with a fork and spoon icon, a title, a materials section, and a note at the top.

このページは、おいしい串揚げのレシピを作ります。やってみてください。

### 材料

鶏もも肉	250g
しょうゆ	大さじ1 ※
酒	大さじ1 ※
おろししょうが	小さじ1/2 ※
おろしにんにく	小さじ1/2 ※
片栗粉	大さじ3
サラダ油	適量

## メインコンテンツ

幅: 680px  
中央揃え

## イントロダクションエリア

幅: 100%  
背景色グレー

## 材料エリア

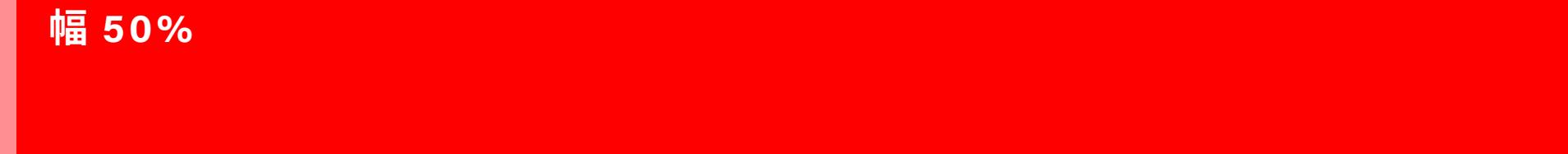
幅: 100%  
上にマージン

## CSS(2) - 親ボックスと相対値の基準

幅 80% など、相対的な値を指定した時、  
原則として、基準は親ボックスの値を基準として計算される

親ボックスの幅が 100% で、子ボックスの幅が 50% の場合

幅 100%



親ボックスの幅が 1000px で、子ボックスの幅が 50% の場合

