

Web プログラミング

第七回 : CSS(4)

CSS(3) - メインコンテンツエリアの構成



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

作り方

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。

2



鶏肉を漬け込む

ボウルに※を混ぜ合わせ、切った鶏肉を入れて汁気がなくなるまでもみ、15分おく。

3



鶏肉に粉をまぶす

パットに片栗粉をいれ、鶏肉に粉をまぶし、余分な粉はふるいおとす。

鶏肉に粉がつさすぎると食感がわるくなります。鶏肉を粉の上で軽がすようにまぶしましょう。

4



鶏肉を揚げる

フライパンに揚げ油を2cmの深さまで入れ、油を170°Cに熱し、鶏肉を揚げる。

2～3分して鶏肉の下部においしそうな褐色がついたら箸で裏返し、さらに2～3分揚げ取り出す。

器に盛りつけ、せん切りしたキャベツ、くし形切りにしたトマトとレモンを添える。

作り方の動画



基本の鶏の唐揚げ

メインコンテンツエリアの構成を考えます

CSS(3) - メインコンテンツエリアの構成



メインコンテンツには

- ・ イントロダクション
- ・ 材料
- ・ 作り方
- ・ 動画

の四つのセクションにわけることができる。

そして、4つのセクションは
全て同じ幅で揃っている。



基本の鶏の唐揚げ

CSS(3) - メインコンテンツエリアの構成



セクションを包括する
メインコンテンツブロックを
680px／中央揃えで配置する。

4つのセクションをその中に配置することで、
全てのセクションの幅と位置を
簡単に揃えることができる。

CSS(3) - 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%

幅 50%

親ボックスの幅が 1000px で、子ボックスの幅が 50% の場合

幅 1000px

幅 50%

CSS(3) - 相対配置と絶対配置



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。

イントロダクションエリアの
レイアウトについて考えます

CSS(3) - 相対配置と絶対配置



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。

この食器のアイコンは、通常のフローレイアウトから外れた位置にいます。
このように要素を特殊な配置にしたい時、
css の **position** プロパティを使います。

CSS(3) - 相対配置と絶対配置

```
.icon{  
  position: static;  
}
```

position:static

初期状態。通常のフローレイアウトで配置される。
位置指定はできない。

```
.icon{  
  position: relative;  
}
```

position:relative

通常のフローレイアウトで配置されるが、
相対的な位置指定ができる

```
.icon{  
  position: absolute;  
}
```

position:absolute

通常のフローレイアウトから外れ、
親ボックスを基準とした絶対的な位置指定ができる

CSS(3) - 相対配置と絶対配置

```
.icon{  
  position: relative;  
  left: 20px;  
  top: 10px;  
}
```

position が **relative**、または **absolute** などのとき、
left、**top**、**right**、**bottom** のプロパティを指定して
位置をしています。

CSS(3) - position: relative



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。

```
.icon{  
  position: relative;  
}
```

position: relative のとき、
基本的に要素は通常の配置方法と同じ位置にいますが、
left、**top**、**right**、**bottom** のプロパティで
位置を指定できるようになります。

CSS(3) - position: relative

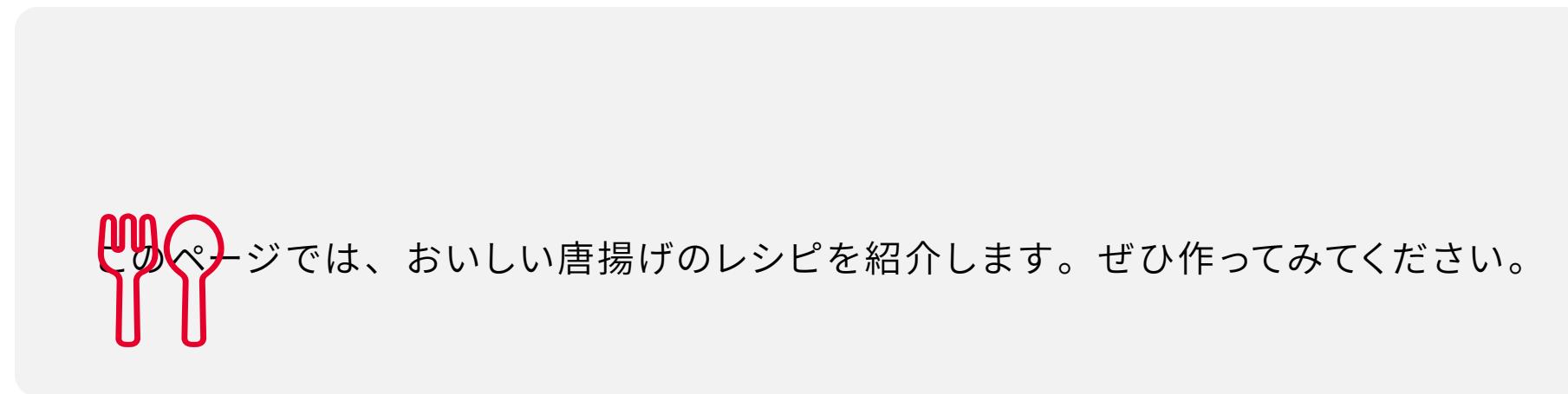


このページでは、おいしい唐揚げのレシピを紹介します。ぜひ作ってみてください。

```
.icon{  
  position: relative;  
  left: 50px;  
}
```

left:50px のように指定すると、
デフォルトの位置から左へ 50px の距離をとります。

CSS(3) - position: relative



```
.icon{  
  position: relative;  
  top: 50px;  
}
```

top:50px のように指定すると、
デフォルトの位置から上へ 50px の距離をとります。

CSS(3) - position: relative

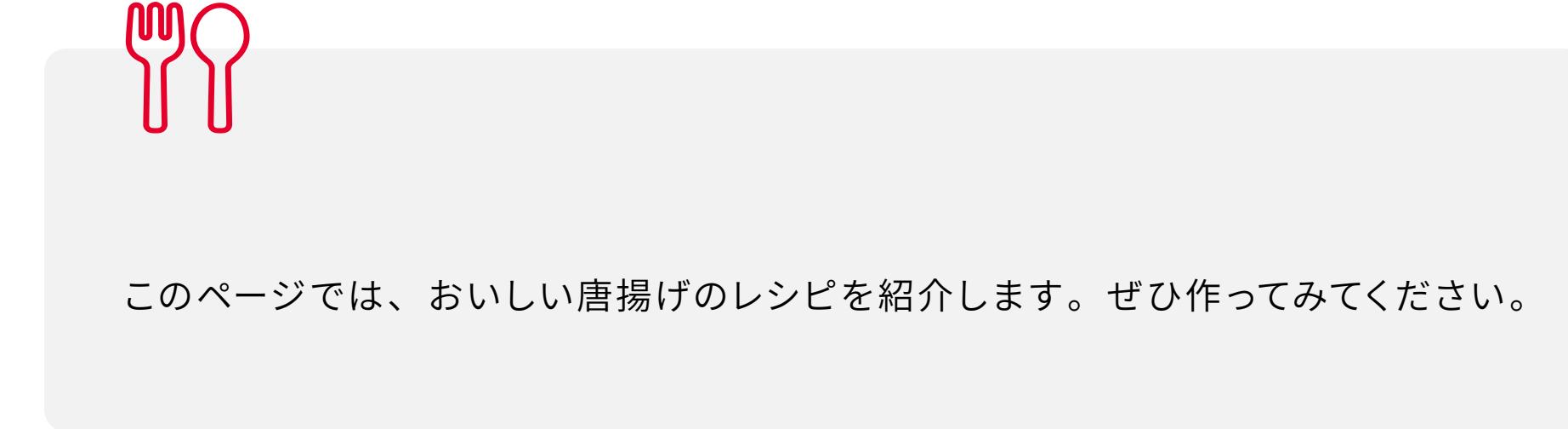


このページでは、おいしい唐揚げのレシピを紹介します。ぜひ作ってみてください。

```
.icon{  
  position: relative;  
  right: 50px;  
}
```

right:50px のように指定すると、
デフォルトの位置から右へ 50px の距離をとります。

CSS(3) - position: relative



```
.icon{  
  position: relative;  
  bottom: 50px;  
}
```

bottom:50px のように指定すると、
デフォルトの位置から下へ 50px の距離をとります。

CSS(3) - position: relative

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。



```
.icon{  
  position: relative;  
  left: 50px;  
  top: 50px;  
}
```

left:50px、top:50px ように、複数の位置を指定することもできます。

CSS(3) - position: relative



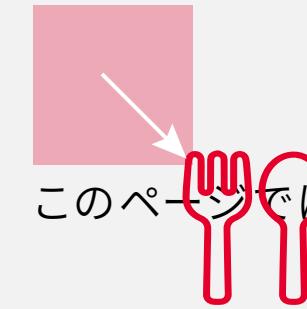
このページでは、おいしい唐揚げのレシピを紹介します。ぜひ作ってみてください。

```
.icon{  
  position: relative;  
  left: 30px;  
  right: 60px;  
}
```

left と **right** では **left** が優先され、
top と **bottom** では **top** が優先されます。
左のように指定した場合、**right** の値は無視され、
left の値が適用されます。

CSS(3) - position:relative

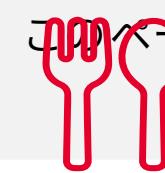
元々要素がいた場所



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。

position:relative で要素を移動したときは、
あくまで見ためのみをずらします。
そのため、移動後も元いた位置に要素がいるようなレイアウトになります。

CSS(3) - position:absolute



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。

```
.icon{  
  position: absolute;  
}
```

position: absolute のとき、要素はフローレイアウト上にいないものとみなされます。そのため、他の要素と重なったり、親ボックスの高さが小さくなったりします。

CSS(3) - position:absolute



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。

```
.icon{  
  position: absolute;  
  left: 0px;  
  top: 0px;  
}
```

left:0px、**top:0px** ように指定すると、
親ボックスのを基準として左から 0px、上から 0px の
位置に配置されます。

CSS(3) - position:absolute

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。



```
.icon{  
  position: absolute;  
  right: 0px;  
  top: 0px;  
}
```

right:0px、top:0px ように指定すると、
親ボックスのを基準として右から 0px、上から 0px の
位置に配置されます。

CSS(3) - position:absolute

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。



```
.icon{  
  position: absolute;  
  right: 0px;  
  bottom: 0px;  
}
```

right:0px、**bottom:0px** ように指定すると、
親ボックスのを基準として右から 0px、下から 0px の
位置に配置されます。

CSS(3) - position:absolute



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。

```
.icon{  
  position: absolute;  
  left: 0px;  
  bottom: 0px;  
}
```

left:0px、**bottom:0px** ように指定すると、
親ボックスのを基準として右から 0px、下から 0px の
位置に配置されます。

CSS(3) - position:absolute



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。

```
.icon{  
  position: absolute;  
  left: -30px;  
  top: -30px;  
}
```

マイナスの値も指定できます。
left:-30px、top:-30px ように指定すると、
お手本のようなレイアウトになります。

CSS(3) - 基準となる親ボックス

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。

```
.introduction{  
  width: 680px;  
  background-color : gray;  
}  
  
.icon{  
  position: absolute;  
  left: -30px;  
  top: -30px;  
}
```

position: absolute の基準となるボックスは、
position が **static** (初期値) 意外に指定されている
直近の先祖要素です。

つまり、初期状態では、任意の要素を基準に
absolute の位置を指定できません。

CSS(3) - 基準となる親ボックス



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。

```
.introduction{  
  width: 680px;  
  background-color : gray;  
  position: relative;  
}  
  
.icon{  
  position: absolute;  
  left: -30px;  
  top: -30px;  
}
```

absolute の基準にしたい要素に、
position:relative などを指定して
基準ボックスになるようにしましょう。

CSS(3) - 要素の重ね順

```
.greenbox{  
    width: 600px;  
    height: 200px;  
    background-color: green;  
    position: absolute;  
    left: 30px;  
    right: 30px;  
}  
  
.redbox{  
    width: 200px;  
    height: 400px;  
    background-color: red;  
}
```



position などを利用して位置を変えると、他の要素と重なることがあります。

CSS(3) - 要素の重ね順

```
.greenbox{  
    width: 600px;  
    height: 200px;  
    background-color: green;  
    position: absolute;  
    left: 30px;  
    right: 30px;  
    z-index: 1;  
}  
  
.redbox{  
    width: 200px;  
    height: 400px;  
    background-color: red;  
}
```



z-index プロパティを指定することで、
その要素の重ね順を決めることができます。
重ね順は、数値が大きいほど手前に置かれます。
(初期値 : 0)