

# HTML5プロフェッショナル認定 レベル--- 技術解説無料セミナー

2021/7/11 開催

主題 1.2.2 CSSデザイン  
副題 ボックス関連仕様の基本をおさえよう！

本日の講師

NTTテクノクロス株式会社  
鈴木 雅貴

## ■ 鈴木雅貴(SUZUKI Masataka)

- NTTテクノクロス株式会社に所属
- デザインの力で会社をよくしていく仕事
- Web標準技術が得意
- アヒル好き



## ■ NTTテクノクロス

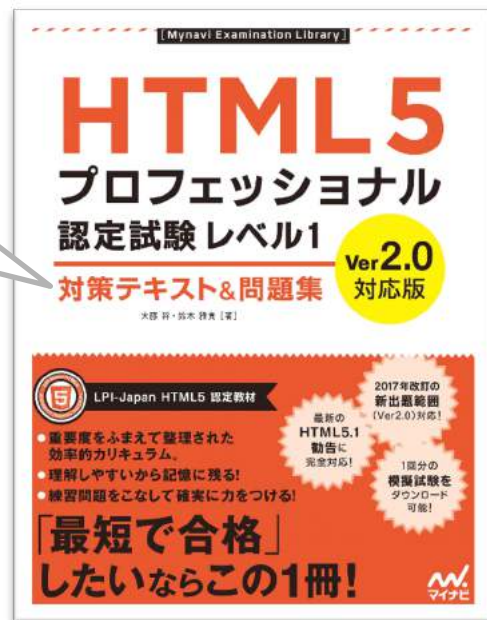
- <https://www.ntt-tx.co.jp/>
- NTT研究所の技術を軸として、世の中の優れた技術・サービスを掛け合わせ、お客様と共にみらいを共創していきます。

## ■ HTML5プロフェッショナル認定試験の対策テキスト

『HTML5プロフェッショナル認定試験 レベル1 対策テキスト&問題集』

『HTML5プロフェッショナル認定試験 レベル2 対策テキスト&問題集』

私の著書は  
こちら



## ■ HTML5プロフェッショナル認定とは

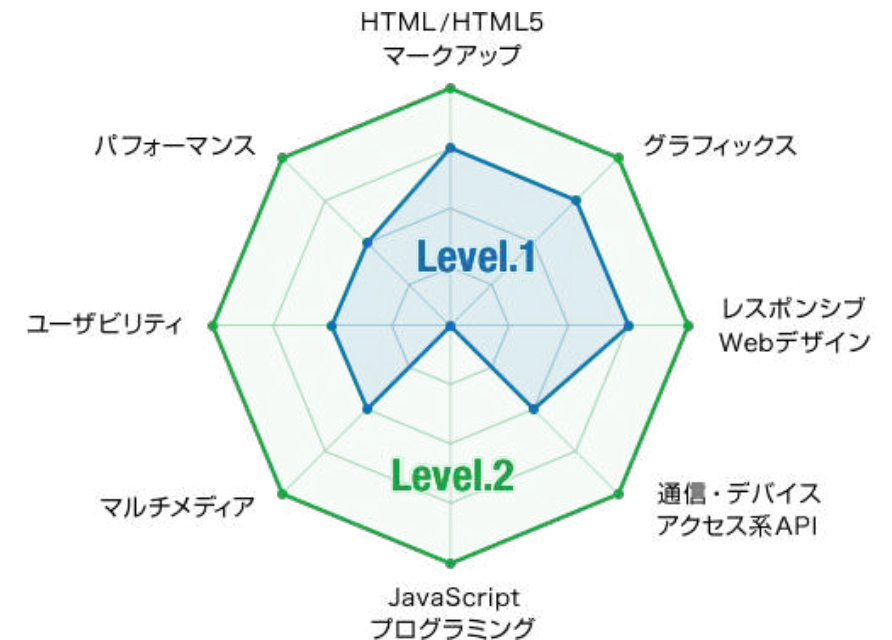
WEBサイトやWEBアプリケーションを開発する上で必須である HTML5／CSS／JavaScriptなどについての技術力を証明する認定です。基礎から網羅的に学ぶことは、効率的に開発を行う上できっと役立つことでしょう。

### ✓ レベル1 はHTMLとCSS

HTMLの基本的な部分からレスポンスデザインが中心で、サイト制作のためのスキルの証明

### ✓ レベル2 ではJavaScript

JavaScriptを使ってWEBアプリケーションを構築できるだけのスキルの証明



## 今回のテーマ

# ボックス関連仕様の基礎をおさえる

- **ボックスとは**
- **CSSでのボックス操作**
- **ボックスタイプ**
- **CSSでのボックスタイプ指定**
- **まとめ**

# ボックスとは

- Webブラウザは、HTMLの要素を長方形の箱として配置する
- この長方形の箱が**ボックス**

## HTML

```
<p>文章です</p>  
<p>これも文章です</p>
```

## 画面イメージ

文章です

これも文章です

これがボックス  
(便宜上黒背景と  
余白を追加)



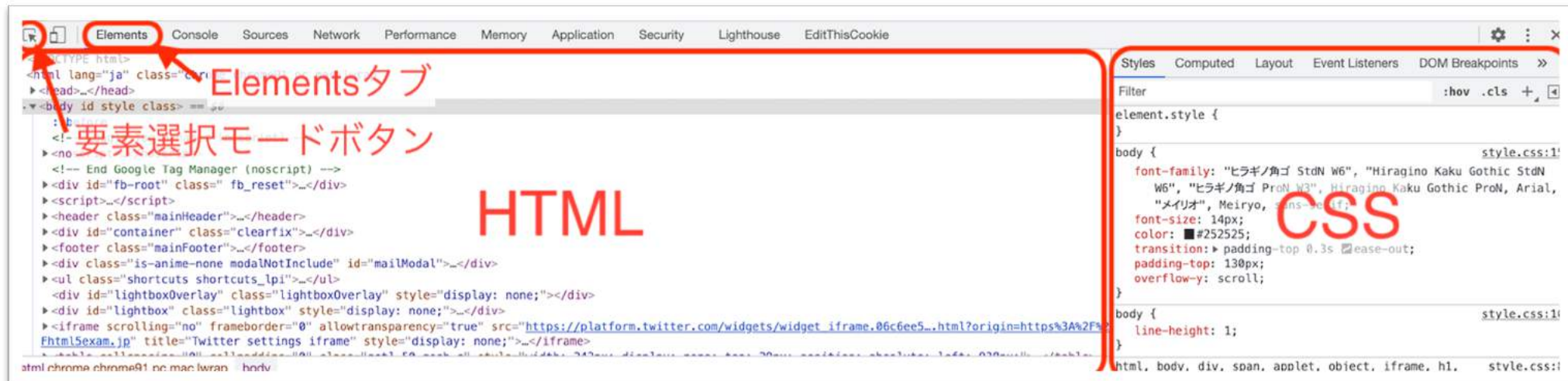


## ■ 「Elements」タブからHTMLとCSSを確認できる

- 一時的な編集も可能

## ■ Chromeでは、確認したいパーツを右クリック → 「検証」で、対応する要素を選択した状態にできる

- 要素選択モードボタンを使っても同様に選択可能





- ボックスは、Webブラウザが、HTMLの要素を長方形の箱として配置したもの
- Webブラウザの開発者ツールで視覚的に確認できる

# CSSでのボックス操作

## ■ ボックスは以下4つで構成されている

コンテンツ領域	コンテンツが表示される領域
パディング	ボーダー内側の余白
ボーダー	ボックスの境界
マージン	ボーダー外側の余白(外の余白なので大きさに含まない)

Chrome開発者ツール  
に色を合わせてます



## ■ ボックス構成要素のサイズは以下のプロパティで指定可能

コンテンツ領域	width(幅), height(高さ)
パディング	padding
ボーダー	border(ショートハンド), border-width
マージン	margin



## ■ boxクラスを以下のボックスにする

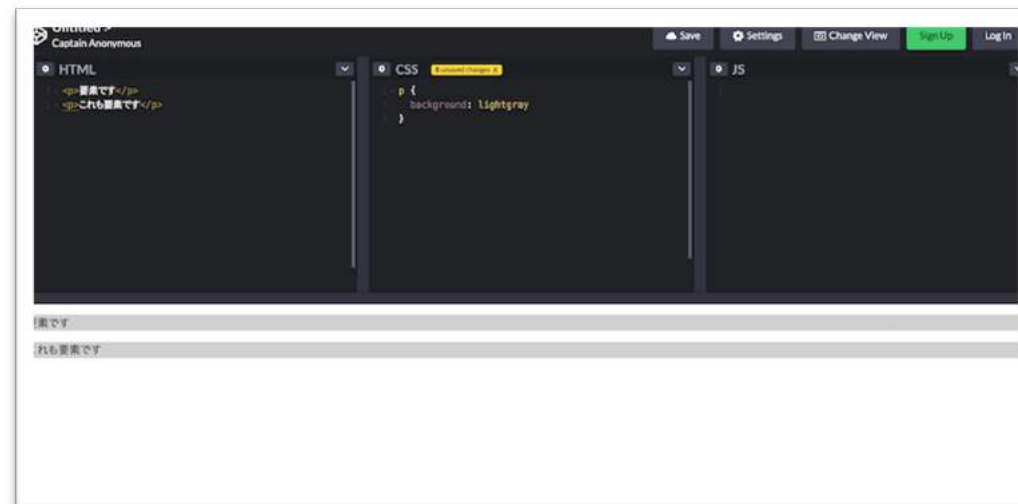
- コンテンツ領域の幅100px、パディング16px、ボーダーは幅8pxの黒い直線、マージン16px

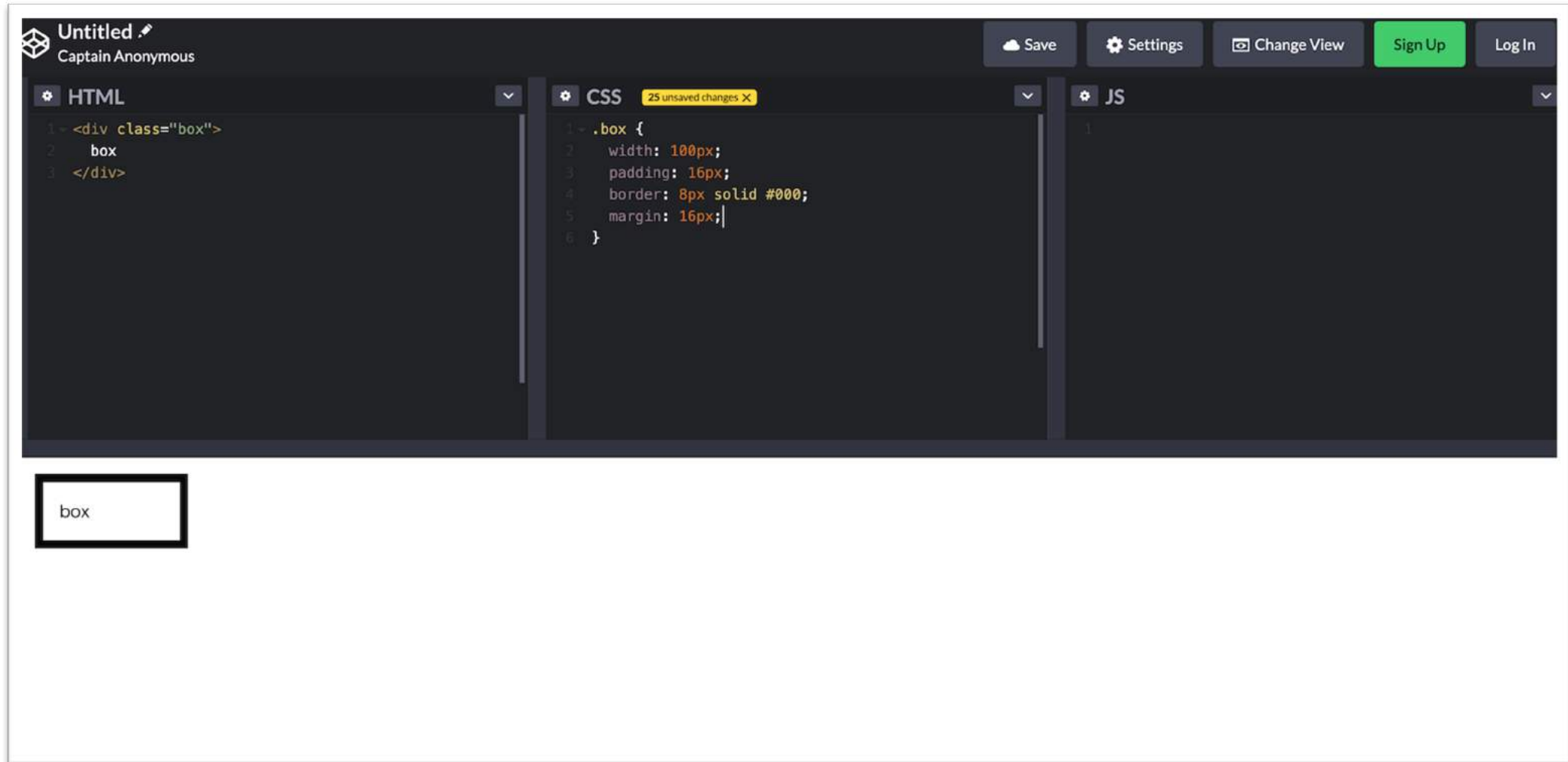
### CSS

```
.box {  
  width: 100px;  
  padding: 16px;  
  border: 8px solid #000;  
  margin: 16px;  
}
```



- ブラウザ上でHTML, CSS, JavaScriptを書いてリアルタイムで確認できるCodePenが便利
  - <https://codepen.io/>
- 左上の「Start Coding」からコードを書く画面にいける
- ユーザ登録すると作成したコードを残せる





The screenshot shows the CodePen editor interface. At the top, there's a header with "Untitled" and "Captain Anonymous" on the left, and "Save", "Settings", "Change View", "Sign Up", and "Log In" buttons on the right. Below the header, there are three tabs: "HTML", "CSS", and "JS".

The HTML tab contains the following code:

```
1 <div class="box">
2   box
3 </div>
```

The CSS tab contains the following code:

```
1 .box {
2   width: 100px;
3   padding: 16px;
4   border: 8px solid #000;
5   margin: 16px;
6 }
```


The JS tab is empty.

Below the code editor, there is a preview area showing a white box with a black border and the text "box" inside.

## ■ 以下のCSSが指定されたボックスの横幅は？

```
.box {  
  width: 100px;  
  padding: 16px;  
  border: 8px solid #000;  
  margin: 16px;  
}
```

右の1~4から選択してください

- 
1. 100px
  2. 132px
  3. 148px
  4. 180px

- ボックスの横幅はコンテンツ領域+パディング+ボーダー
  - コンテンツ領域の幅: 100px
  - パディングの幅:  $16\text{px} * 2 = 32\text{px}$
  - ボーダーの幅:  $8\text{px} * 2 = 16\text{px}$

上記の計:  $100\text{px} + 32\text{px} + 16\text{px} = 148\text{px}$

答え: 148px

- ボックスの幅は、 $\text{width} + \text{padding} + \text{border}$
- 直感的には、widthがそのままボックスの幅になってほしい
  - レイアウトを考えるとときにいちいち計算するのは面倒
  - heightにおいても同様
- この要望に答えるべく、サイズの算出方法を変更するCSSプロパティ **box-sizing** が登場

- **box-sizing**プロパティで、要素全体の幅と高さの算出方法を変更することができる
- 指定できる値は以下

content-box	width, heightで指定した値が、コンテンツ領域の幅と高さになる(デフォルト)
border-box	width, heightで指定した値が、コンテンツ領域+パディング+ボーダーを含んだボックスの幅と高さになる

- さきほどのboxクラスに**box-sizing: border-box**を追加
  - ボックスの幅はwidthで指定した100pxになる


## CSS

```
.box {  
  box-sizing: border-box;  
  width: 100px;  
  padding: 16px;  
  border: 8px solid #000;  
  margin: 16px;  
}
```

## ■ 以下のCSSが指定されたボックスのコンテンツ領域幅は？

```
.box {  
  box-sizing: border-box;  
  width: 100px;  
  padding: 16px;  
  border: 8px solid #000;  
  margin: 16px;  
}
```

右の1~4から選択してください

- 
1. 100px
  2. 68px
  3. 52px
  4. 20px



- **box-sizing: border-box**なので、ボックスの横幅はwidthで指定された100pxで、コンテンツ領域はこの100pxからパディングとボーダー分を除いたもの
  - **パディングの幅:  $16\text{px} * 2 = 32\text{px}$**
  - **ボーダーの幅:  $8\text{px} * 2 = 16\text{px}$**

上記から:  $100\text{px} - 32\text{px} - 16\text{px} = 52\text{px}$

**答え: 52px**

- ボックスは「コンテンツ領域」「パディング」「ボーダー」「マージン」の4つから構成される
  - サイズ計算にマージンは含まない
- CSSでのサイズ指定はwidth,heightと、paddingと、borderと、marginを使う
- box-sizingプロパティでサイズの算出方法を変えられる

# ボックスタイプ

## ■ CSSにおけるボックスタイプには2種類ある

ブロック	<ul style="list-style-type: none"><li>• ボックスは横に伸び、可能な限りスペースを埋める</li><li>• ボックスが新しい行を作る</li><li>• widthとheightでサイズ指定可能</li><li>• コンテンツを入れる枠となるイメージ(例: p要素)</li><li>• 中にブロックとインラインを入れられる</li></ul>
インライン	<ul style="list-style-type: none"><li>• コンテンツによりサイズが決まる</li><li>• ボックスは新しい行を作らず、横に並ぶ</li><li>• widthとheightでサイズを指定できない</li><li>• 行の一部となるイメージ(例: a要素)</li><li>• 中にはインラインしか入れられない</li></ul>

## ブロック

ブロックです

これもブロックです

横いっぱいの幅で、要素ごとに新しい行となり、縦に積まれる

## インライン

コンテンツ幅で横並び

インライン | これもインライン | インライン

ブロックです | インライン | ブロックです

行の一部となる

- HTML4.01まで、ブロックかインラインかを要素ごとに指定
- HTML5から、ブロックとインラインは表示側の機能ということで、要素に指定されなくなった
  - ブロックとインラインはCSSの機能という扱い
- しかし、現在でもWebブラウザのデフォルトCSSでは、要素ごとにブロックとインラインを指定している
  - 参考: MDNの[ブロックレベル要素](#)と[インライン要素](#)

- ボックスには大きく「ブロック」と「インライン」の2つがある
- ブロックはコンテンツを入れる枠のイメージ
- インラインは行の一部のイメージ

# CSSでのボックスタイプ指定

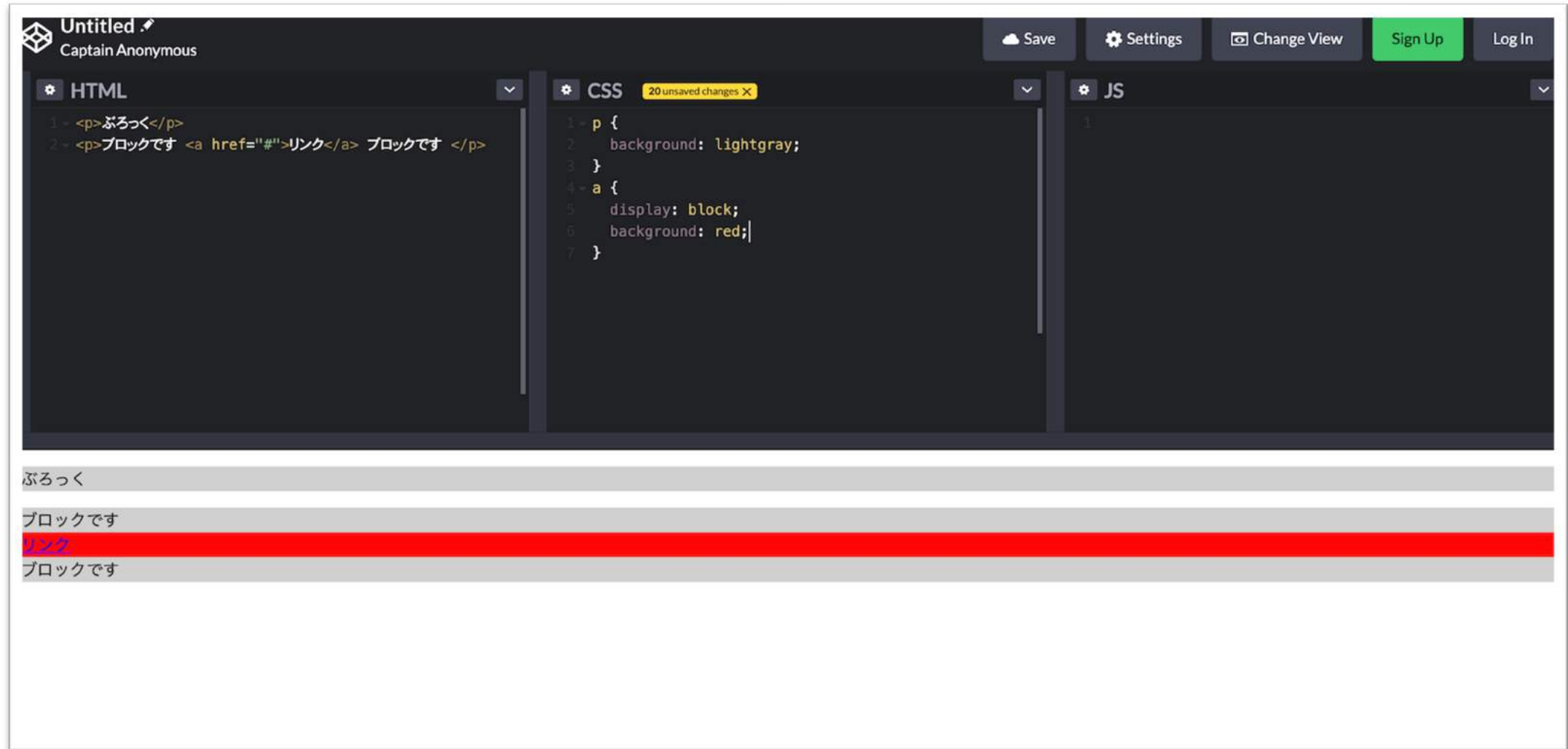


- **display**プロパティで、ボックスタイプを指定できる
- 値はブロックならblock、インラインならinline

CSS

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

p要素はデフォルトではブロック  
だが、インラインに変更できる



The screenshot shows the CodePen editor interface. At the top, there's a header with "Untitled" and "Captain Anonymous" on the left, and "Save", "Settings", "Change View", "Sign Up", and "Log In" buttons on the right. Below the header are three panels: HTML, CSS, and JS.

**HTML Panel:**

```
1 <p>ぶろっく</p>
2 <p>ブロックです <a href="#">リンク</a> ブロックです </p>
```

**CSS Panel:**

```
1 p {
2   background: lightgray;
3 }
4 a {
5   display: block;
6   background: red;
7 }
```

**JS Panel:**

```
1
```

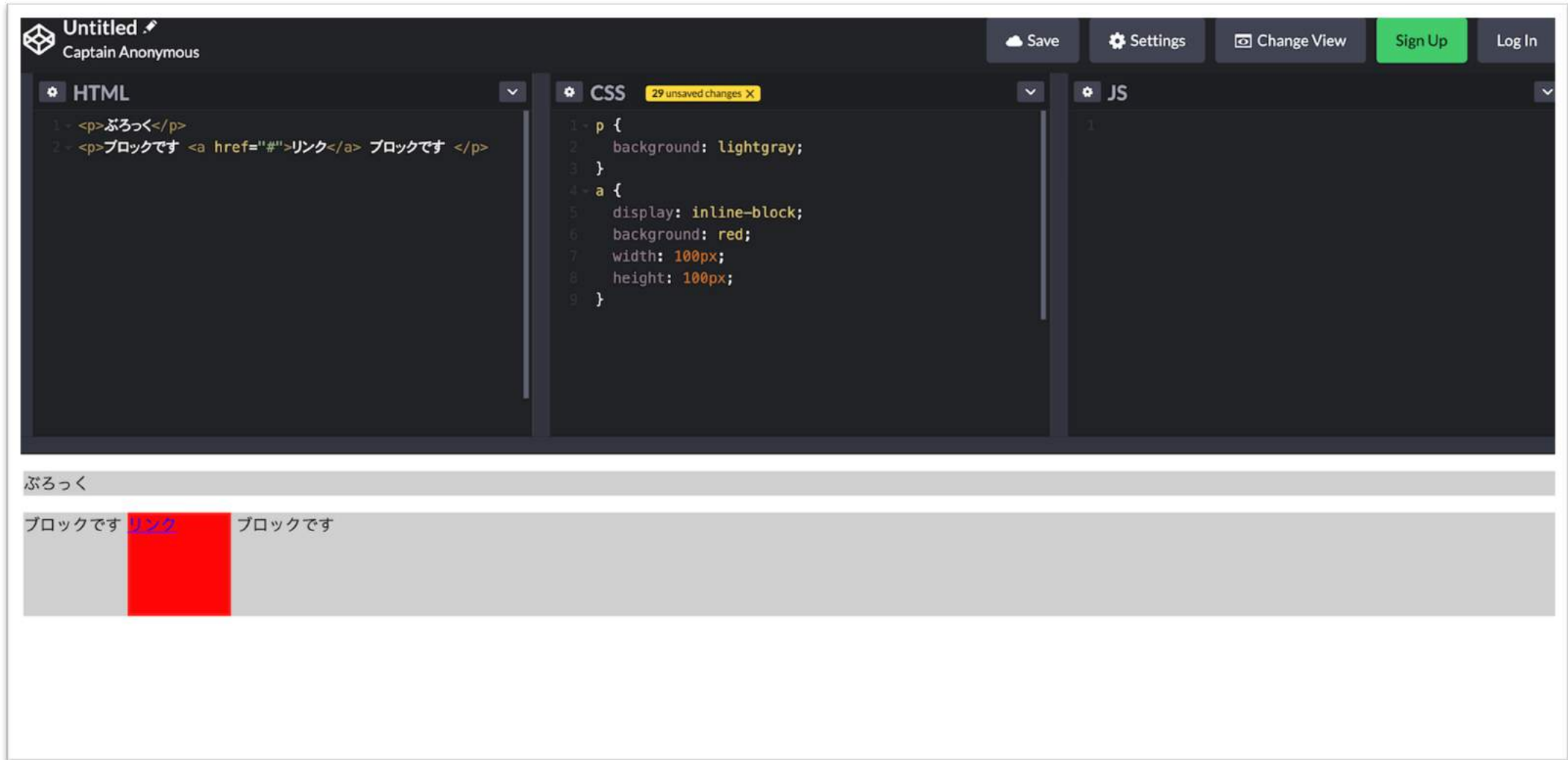
Below the code panels is a preview area showing the rendered output:

- ぶろっく
- ブロックです
- [リンク](#)
- ブロックです

- インラインではwidth, heightによるサイズ指定ができない
- ブロックのようにサイズ指定できて、インラインのようにレイアウトされる**inline-block**がある

## CSS

```
a {
  display: inline-block;
  width: 200px;
  height: 100px;
}
```



The screenshot shows the CodePen editor interface. At the top, there's a header with "Untitled" and "Captain Anonymous" on the left, and "Save", "Settings", "Change View", "Sign Up", and "Log In" buttons on the right. Below the header are three panels: HTML, CSS, and JS.

**HTML Panel:**

```
1 - <p>ぶろっく</p>
2 - <p>ブロックです <a href="#">リンク</a> ブロックです </p>
```

**CSS Panel:** (29 unsaved changes X)

```
1 - p {
2   background: lightgray;
3 }
4 - a {
5   display: inline-block;
6   background: red;
7   width: 100px;
8   height: 100px;
9 }
```

**JS Panel:**

```
1
```

**Preview:**

ぶろっく

ブロックです [リンク](#) ブロックです

- displayプロパティの値は、block, inline, inline-block以外にもいろいろある
- 詳細は[MDNのdisplayプロパティ解説](#)を参照
- いろいろあるが、要素のレイアウトは上3つが基本となる

- **displayプロパティでブロックやインラインを指定可能**
- **サイズ指定可能で、かつインラインのようにレイアウトされる inline-blockがある**

- ボックスとは、Webブラウザが、HTMLの要素を長方形の四角として配置したもの
- ボックスはコンテンツ領域、パディング、ボーダー、マージンから構成され、それぞれCSSでサイズを指定できる
- ボックスタイプにはブロックとインラインがある
- CSSでブロックとインラインを指定できる
- ブロックのようにサイズ指定ができ、インラインのようにレイアウトされるinline-blockがある

## ■ MDN

- Mozillaが中心となって作成するWeb標準ドキュメントで、情報の信頼性はとても高い！
- <https://developer.mozilla.org/ja/>

## ■ CodePen

- Webブラウザ上からHTML, CSSを編集し確認できる
- <https://codepen.io/>

## ■ Webブラウザ開発者ツール