

JavaScript 入門 テキスト課題 最終補足 オーバーライド

継承元クラスのメソッド（コンストラクタ含む）を上書きし、違う機能にすることができます。これをオーバーライドといいます。なお、コンストラクタをオーバーライドする際は、**super** を使用して継承元クラスのコンストラクタを呼び出して下さい（呼び出さないと **this** がつかえなくなる）。

保存時ファイル名 : [texta-over-sample-1.html](#) 、 [texta-over-1.html](#) 等

<サンプルプログラム>

```
class Cat {
  constructor( name) {
    this.name = name;
  }
  naku(){
    alert("ニャー");
  }
}

class BakeNeko extends Cat{
  constructor( name,age) {
    super(name);
    // this.name = name;
    this.age = age;
  }
  naku(){
    super.naku();
    alert("俺は化け猫" + this.name +
          this.age + "歳だ");
  }
}

//オブジェクト生成
let cat = new BakeNeko("タマ",100);
cat.naku();
```

Cat を継承して BakeNeko を定義

constructor をオーバーライド

スーパークラスの **constructor** を呼び出す

スーパークラスで行うのでここではいらない

naku をオーバーライド

スーパークラスの **naku** を呼び出す

<練習問題>

練習 1

車クラス (Car) を作成して下さい。

以下のようなメソッド「printInfo」を定義して下さい。

処理概要: 「Car は時速 100km まで出ます」を画面表示する。

引数: 無し 戻り値: 無し

車クラスを継承し、早い車クラス (HayaiCar) を作成して下さい。

メソッド「printInfo」をオーバーライドして下さい。スーパークラスの printInfo を呼び出し、その後、「HayaiCar は、ターボ機能を使うとプラス 50km!」を、画面表示して下さい。

HayaiCar クラスのオブジェクトを生成し printInfo を実行して下さい。

結果例: Car は時速 100km まで出ます

HayaiCar は、ターボ機能を使うとプラス 50km!

練習 2

食事クラス (Syokuji) を作成して下さい。

以下のようなメソッド「printInfo」を定義して下さい。

処理概要: 「今日の食事は、卵かけごはん」を画面表示する。

引数: 無し 戻り値: 無し

食事クラスを継承し、良い食事クラス (YoiSyokuji) を作成して下さい。

メソッド「printInfo」をオーバーライドして下さい。スーパークラスの printInfo を呼び出し、その後、「おかずは、ステーキ」を、画面表示して下さい。

YoiSyokuji クラスのオブジェクトを生成し、printInfo を実行して下さい。

結果例: 今日の食事は、卵かけごはん

おかずは、ステーキ