

▼ JavaScript/TypeScriptの「コンストラクター関数」って何？

オブジェクトを量産するための「設計図」です！

イメージ：

設計図 (constructor) → 製品（オブジェクト）

```
function Human(...) → const hito = new Human(...)
```

「同じ型のオブジェクトを何回も作るなら、設計図を使った方が楽だよ！」

書き方と使い方（基本形） ステップ①：関数名を大文字でスタート

```
[ ] ➤ function Human(name, age) {  
    this.name = name;  
    this.age = age;  
}
```

thisは「作られるオブジェクト自身」を表す。

ステップ②：newを使って実体（オブジェクト）を作る

```
[ ] const taro = new Human("太郎", 30);  
const hanako = new Human("花子", 25);
```

ステップ③：プロパティを呼び出す

```
console.log(taro.name); // 太郎  
console.log(hanako.age); // 25
```

▼ どうしてnewを使うの？

newをつけることで「Human関数が特別な関数（コンストラクター）」として動きます！

自動的に1.空のオブジェクト{}を作ってくれる！

```
const obj = {};
```

2.thisをその新しいオブジェクトに結びつけてくれる！

```
function Person(name) {  
    this.name = name; // ← thisが新しいオブジェクトを指すようになる  
}  
const p = new Person("たろう");  
console.log(p.name); // "たろう"
```

3.returnしなくとも、自動でそのオブジェクトを返してくれる！

```
function Person(name) {  
    this.name = name;  
    // return はいらない！  
}  
const p = new Person("はなこ");  
console.log(p); // Person { name: 'はなこ' }
```

▼ 同じパターンでたくさん作れる！

```
const ninjai = new Human("シノビ丸", 28);
```

```
const ninja1 = new Human("シノビ丸", 22);
const ninja2 = new Human("カゲマル", 33);

console.log(ninja1.name); // シノビ丸
console.log(ninja2.name); // カゲマル
```

▼ メソッドを追加して動きをつけよう

```
function Human(name, age) {
  this.name = name;
  this.age = age;

  this.greet = function() {
    console.log(`こんにちは、私は${this.name}です！`);
  };
}

const alice = new Human("アリス", 22);
alice.greet(); // こんにちは、私はアリスです！
```

▼ コンストラクターを使わないパターン

```
const taro = {
  name: "太郎",
  age: 30
};

const hanako = {
  name: "花子",
  age: 25
};
```

▼ コンストラクター 一覧

String コンストラクター

```
const str = String("こんにちは");
```

Number コンストラクター

```
const num = Number(42);
```

Array コンストラクター

```
const arr = new Array("りんご", "バナナ");
```

Object コンストラクター

```
const obj = new Object({ name: "太郎", age: 20 });
```

Date コンストラクター

```
const now = new Date();
```

RegExp コンストラクター

```
const pattern = new RegExp("abc");
```

Map コンストラクター

```
const myMap = new Map();
myMap.set("key", "value");
```

Set コンストラクター

```
const mySet = new Set([1, 2, 2, 3]);
```

URL コンストラクター (URL解析)

```
const url = new URL("https://example.com?lang=ja");
console.log(url.searchParams.get("lang")); // "ja"
```

Error コンストラクター (エラー生成)

```
throw new Error("問題が発生しました!");
```

Colab の有料サービス - 契約解除はこちら