

JavaScript入門：インスタンスプロパティ (Instance Properties)

インスタンスプロパティとは？

インスタンス (Instance) とは、「実際に生成されたデータの実体」を指します。JavaScriptでは、配列・関数・文字列・オブジェクト・HTML要素などがインスタンスになります。

それぞれのインスタンスには、プロパティ (属性) やメソッド (機能) が備わっています。

配列 (Array) のインスタンスプロパティ

```
const fruits = ["apple", "banana", "cherry"];
console.log(fruits.length);
console.log(fruits[0]);
```

解説：・fruits.length → 配列の要素数を返します。この場合は 3。・fruits[0] → 配列の最初の要素 (インデックス0) を返します。この場合は "apple"。

配列は、数を数えるときやループ処理を行うときに .length をよく使います。

関数 (Function) のインスタンスプロパティ

```
function aisatsu(name, job, age){
  return `Hello 名前: ${name}, 年齢: ${age}, 仕事: ${job}`;
}

console.log(aisatsu.length);
```

解説：・関数の .length は引数 (パラメータ) の数を返します。→ この関数は (name, job, age) の3つの引数があるため、出力は 3 です。

```
function aisatsu2(name, job, age){
  return `Hello 名前: ${name}, 年齢: ${age}, 仕事: ${job}`;
}

console.log(aisatsu2.name);
```

解説：・.name は関数名を返します。→ 出力は "aisatsu2" になります。

オブジェクト (Object) のインスタンスプロパティ

```
const p = {
  name: "mario",
  age: 35,
  job: "plumber"
};

console.log(p.age);
console.log(p.length);
```

解説：・p.age はオブジェクト p 中にあるキー age の値 (35) を取得します。・ただし、p.length は存在しないため undefined になります。

配列には .length が使えますが、オブジェクトには使えません。もしオブジェクトの数を数えたいときは Object.keys(p).length を使います。

配列の中にオブジェクトがある場合

```
const people = [
  { name: "mario", age: 35, job: "plumber" },
  { name: "luigi", age: 31, job: "plumber" },
  { name: "peach", age: 21, job: "princess" }
];

console.log(people.length);
console.log(people[0].name);
```

コーディングを開始するか、AI で生成します。

解説：・people.length → 配列全体の要素数 (3)。・people[0].name → 配列の1番目のオブジェクトの name 値を取得 ("mario")。

people.name のように書くと、配列そのものに name プロパティが無いためエラーになります。必ず [n].key の形でアクセスしましょう。

文字列 (String) のインスタンスプロパティ

```
const msg = "hello";
console.log(msg.length);
console.log(msg[0]);
```

解説：・.length → 文字列の長さを返します。この場合は 5。・[0] → 最初の文字 "h" を取得します。

文字列も、実際には「Stringオブジェクト」として扱われているため、プロパティを持っています。

try...catch文とエラープロパティ

```
try {
  let num = "";
  console.log(num.toFixed(0));
} catch(e) {
  console.log("エラー:", e.name, e.message);
}
```

解説：・toFixed() は数値型にしか使えません。空文字 "" に対して使うとエラーになります。・try ブロック : エラーが発生するかもしれない処理を実行。・catch(e) ブロック : エラーが起きたときに実行される部分。

e.name にはエラーの種類 (例 : TypeError) e.message にはエラーの説明があります。

HTML要素のインスタンスプロパティ

HTMLのタグ (要素) も、JavaScriptから操作できる インスタンスです。

```
<div id="box" class="red"></div>

<script>
const box = document.getElementById("box");
console.log(box.className); // red
box.className = "blue"; // classを書き換え
</script>
```

解説：・.className → クラス名を取得・変更できます。・.id → 要素のID。・.innerText → テキスト内容のみ。・.innerHTML → HTMLタグを含んだ内容。・.outerHTML → 自分自身を含めたHTML構造。・.style → 直接CSSを変更 (例 : box.style.color = "red")。

よく使うHTML要素のプロパティ一覧

HTML要素もオブジェクトの一種。

JavaScriptからアクセスして値を読み取ったり変更したりできます。

■ プロパティ	● 説明	✓ 使用例
.id	要素のIDを取得・変更	box.id = "newBox"
.className	クラス名を取得・変更	box.className = "blue"
.title	ツールチップに表示されるタイトル	box.title = "ヒント"
.tagName	タグ名を取得 (例: DIV, P)	console.log(box.tagName)
.innerText	テキストのみ取得	console.log(box.innerText)
.innerHTML	HTMLタグを含む内容を取得・変更	console.log(box.innerHTML)
.outerHTML	自己自身のタグごと文字列に変換	console.log(box.outerHTML)
.value	入力値を取得 (inputなど)	input.value
.selectedIndex	<select> の選択位置	select.selectedIndex
.checked	チェック状態 (true / false)	checkbox.checked
.files	選択されたファイル情報	fileInput.files
.style	inline CSS を直接指定	box.style.background = "red"

まとめ

「インスタンスプロパティ」は中身を読む力。

オブジェクトの情報を「取り出す」「確認する」「操作する」ための基本要素です。

種類	例	主なプロパティ	説明
配列 (Array)	["a", "b"]	.length, [0]	要素数・特定の要素を取得
関数 (Function)	(function f() {})	.name, .length	関数名・引数の数を取得
オブジェクト (Object)	{key: value}	.key	任意のプロパティ値を取得
文字列 (String)	"hello"	.length, [0]	文字数・特定文字を取得
HTML要素 (HTMLElement)	<div>	.id, .className, .innerText	要素の情報を取得・変更