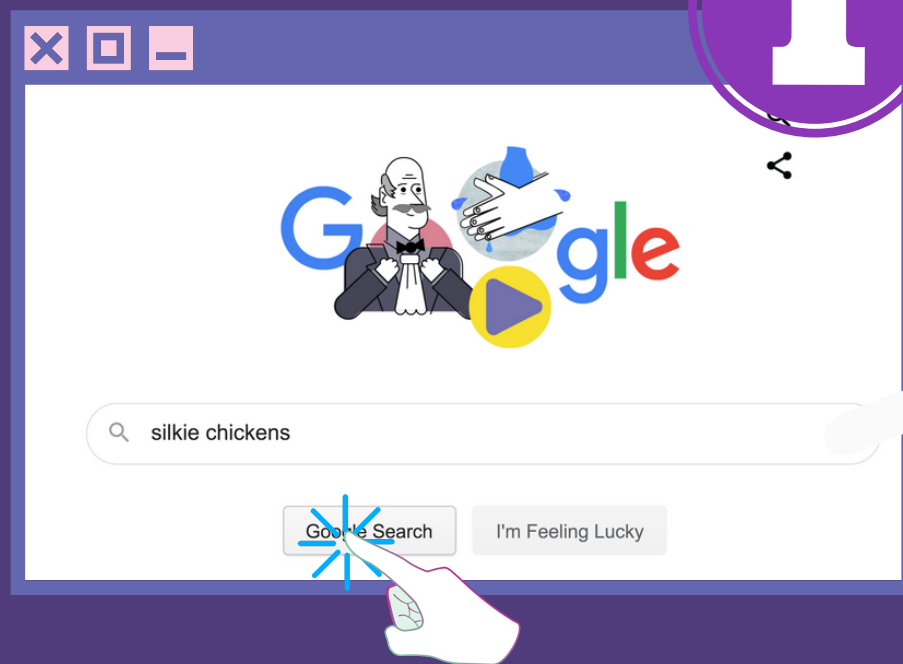


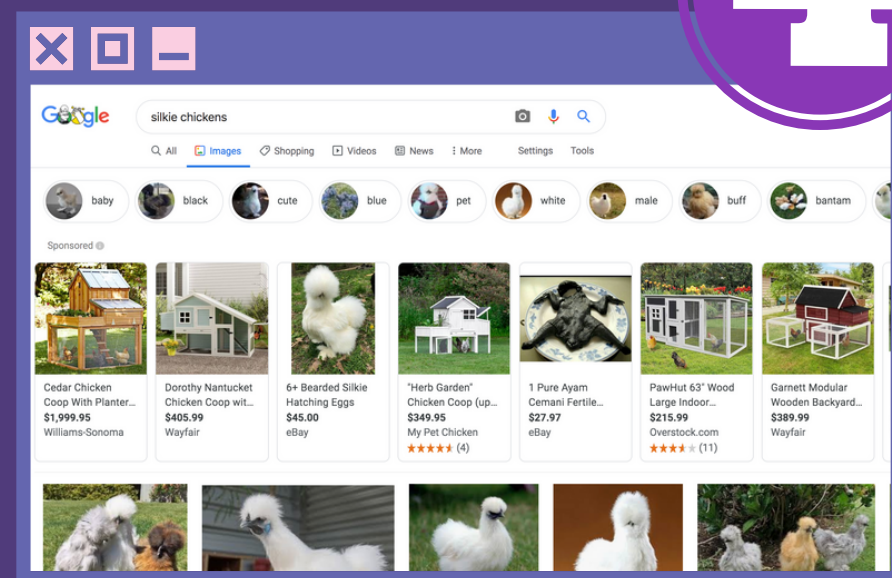
# JSの世界





1

↓ ちょうだい  
GOOGLE.COM/SEARCH?Q=CHICKENS



4

フロントエンド

2

ちょっと待  
ってね



3

はいよ！



バックエンド

紫色の CSS - 形容詞

恐竜が HTML - 名詞

踊ってる JS - 動詞



Web Developer Bootcamp

# JavaScript の基本

値と変数



1

JavaScript単体で  
学ぶ。HTML/CSS  
は無し。



2

JavaScriptを使っ  
てHTML/CSSを操  
作する。

# プリミティブ型

基本となる**JavaScript**の構成要素

Number

String

Boolean

Null

Undefined

\* SymbolとBigIntもプリミティブ型



N

## 様々なデータの種類

### Hall & Oates - When The Morning Comes

517,272 回視聴・2011/04/03

 3084

 104

 共有

 保存

...



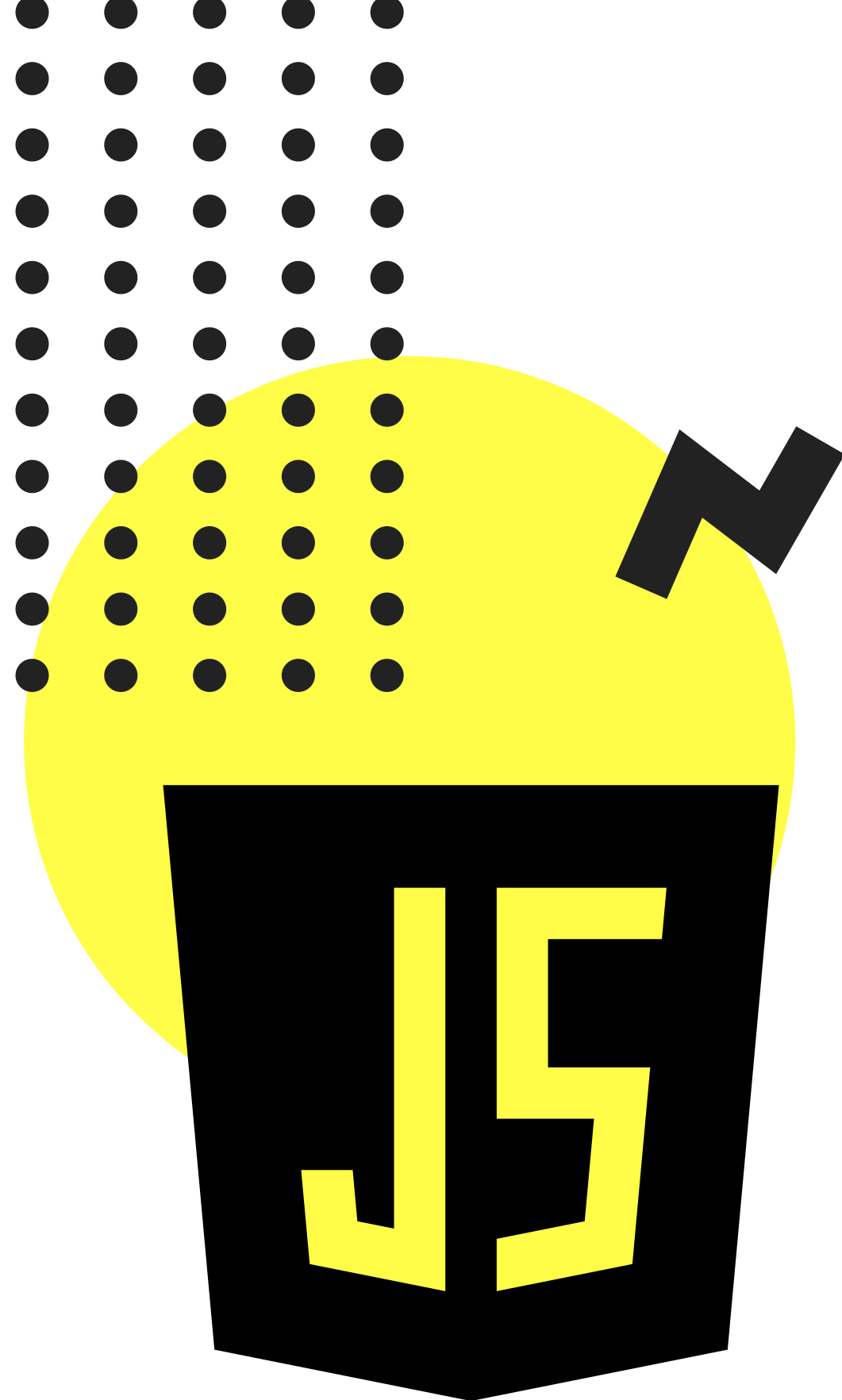
**mickey castle**

チャンネル登録者数 1790人

チャンネル登録

Best Songs From 1970's Hall & Oates

もっと見る

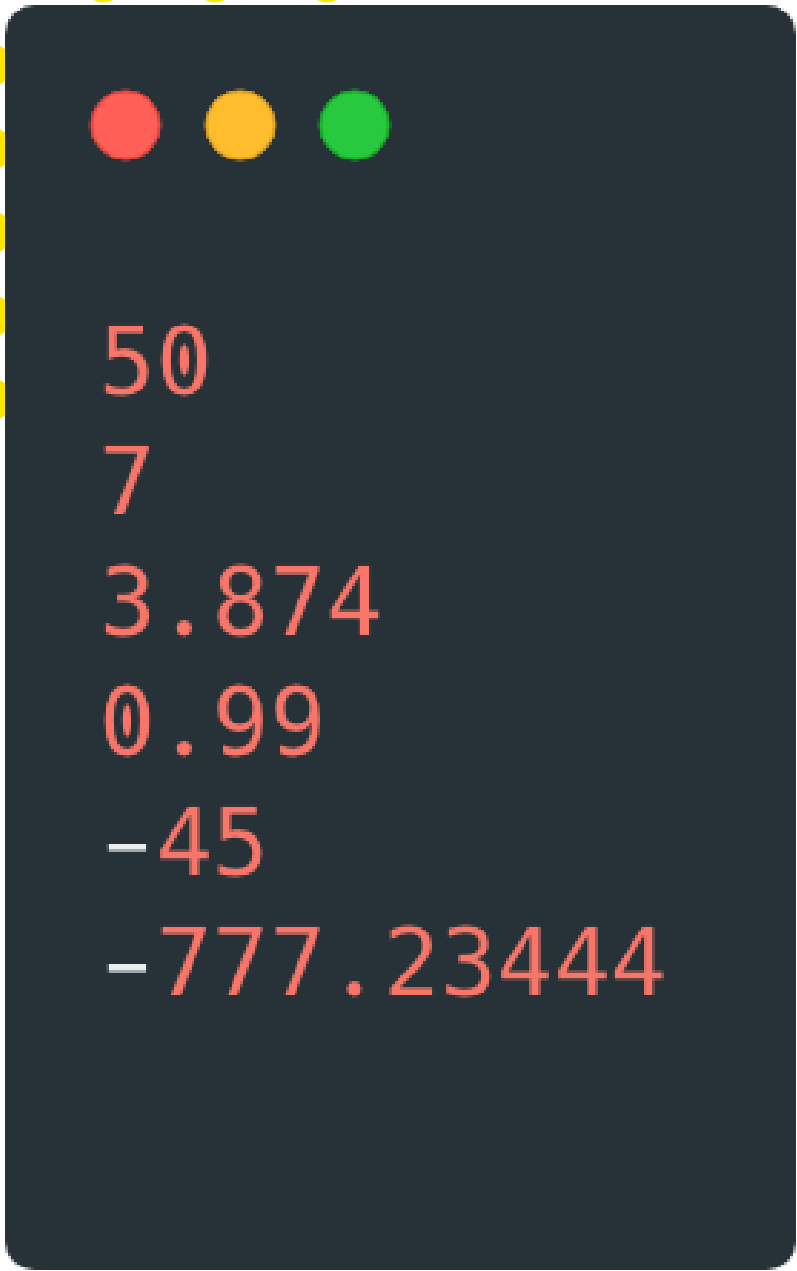
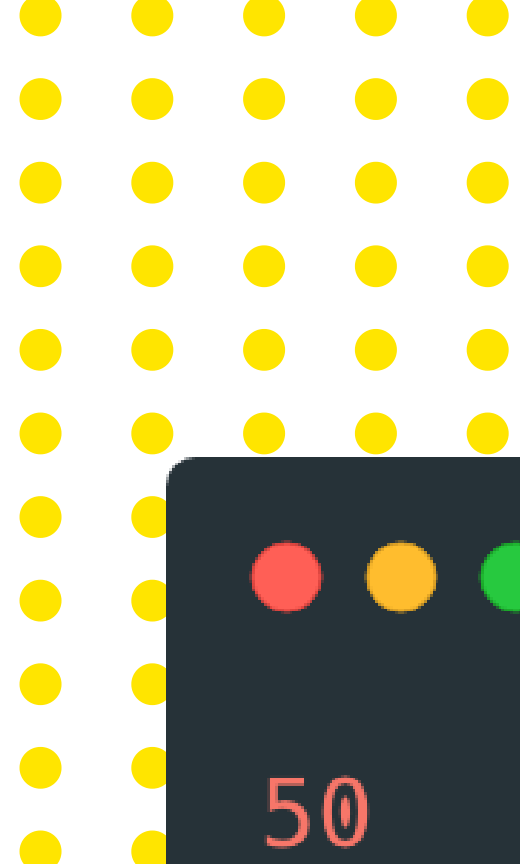


# コンソールで コードを実行

簡単に始められる

まずはChromeの開発者ツールのコンソールでコードを実行します。それから外部ファイルにJSを書く方法を学びます。





```
50
7
3.874
0.99
-45
-777.23444
```

JavaScriptにおける

# 数値 (Number)

- JSには数値型は一つしかない
- 正の数
- 負の数
- 整数
- 浮動小数点数

# 演算

```
// 加算
50 + 5 // 55

// 減算
90 - 1 // 89

// 乗算
11111 * 7 // 77777

// 除算
400 / 25 // 16

// 剰余演算
27 % 2 // 1

// べき演算
2 ** 3 // 8
```

// はコメント  
(この行は無視される)



NOT A NUMBER

**NaN**

NaNは「非数」を表す値



**N**

# Not A Number



```
0/0 //NaN
```

```
1 + NaN //NaN
```

この計算結果は  
どうなるでしょう？



4 + 3 \* 4 / 2

この計算結果は  
どうなるでしょう？



```
(13 % 5) ** 2
```

この計算結果は  
どうなるでしょう？



200 + 0/0



# 変数

変数は値につけておくラベルのようなもの

変数に値を入れておくことで:

- 後で参照できる
- 何かしら処理をするために使える
- 後から違う値に変えることもできる



# 基本的な構文

```
let someName = value;
```

# 基本的な構文

```
let year = 1985;
```

yearという変数を作ってその中に1985という値を入れる

# 変数を使う



```
let hoge = 4;
```

```
let moge = 2;
```

```
hoge + moge // 6
```

# 変数を使う



```
let hoge = 4;
```

```
hoge - 1; // 3
```

```
hoge; // 4のまま!
```

```
// hogeの値を変える方法
```

```
hoge = hoge - 1;
```

```
hoge; // 3
```

この方法ではhogeの値は  
変わらない

これなら変わる！

# CONST



```
const hoge = 4;  
hoge = 20; // エラー!
```

```
const age = 17;  
age = age + 1; // エラー!
```

`const`は`let`のように宣言できるけど、後から値を変えることができない

# なぜCONSTを使うか



```
const pi = 3.14159;  
  
const daysInWeek = 7;  
  
const minHeightForRide = 60;
```

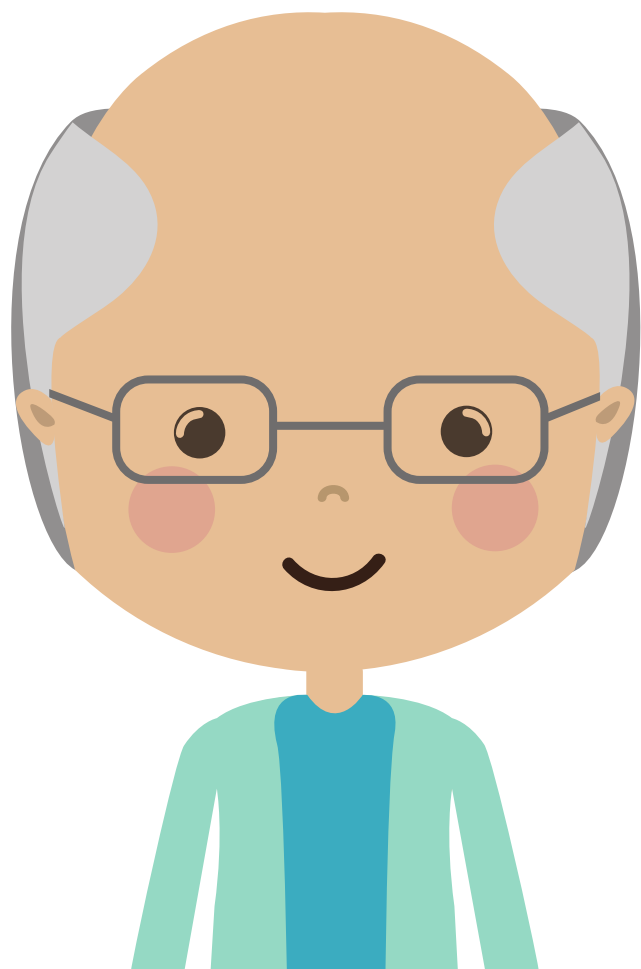
配列やオブジェクトを学ぶときに、なぜletよりもconstを使うことがあるのかまた詳しく見ます



# VAR

昔は**var**しか使えなかった

letとconstの登場以前はvarでしか変数を宣言できなかった。letとconstの登場により、varを使う理由はほとんど無くなった。



# totalScoreの値は？



```
let totalScore = 199;  
totalScore + 1;
```



# totalScoreの値は？



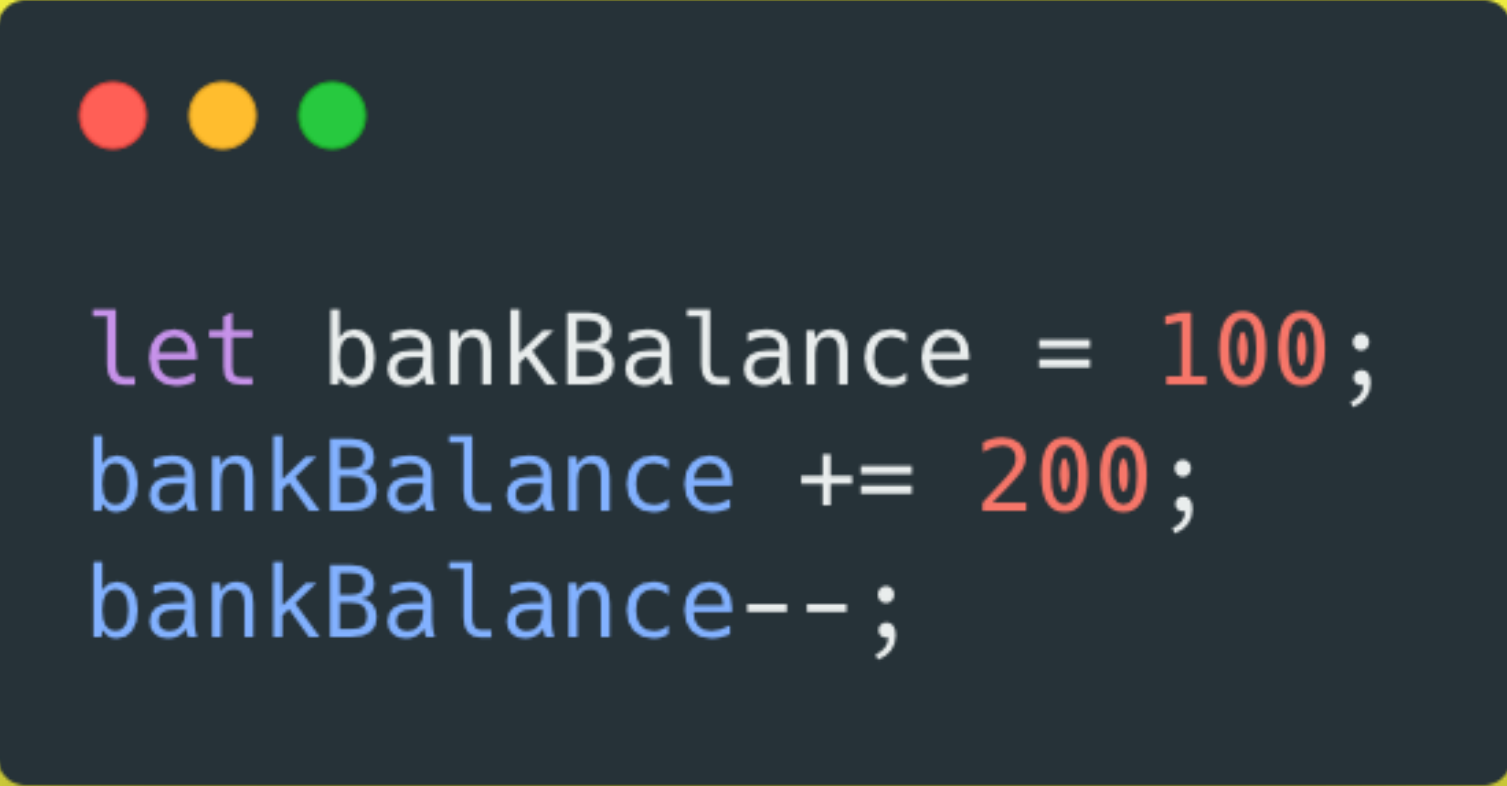
```
let totalScore = 199;  
totalScore + 1;
```

# temperatureの値は？



```
const temperature = 83;  
temperature = 85;
```

# bankBalanceの値は？



```
let bankBalance = 100;  
bankBalance += 200;  
bankBalance--;
```

# BOOLEAN

TRUE


or

FALSE



# Boolean

TRUE OR FALSE



```
let isLoggedIn = true;  
let gameOver = false;  
const isWaterWet = true;
```

booleanはシンプルで、trueかfalseの2種類の値しかありません！

# 変数のデータ型は変えられる！



```
let numPuppies = 23; // Number  
numPuppies = false; // Booleanに変身！  
numPuppies = 100;    // 再びNumber！
```

使う場面は稀ですが、変えることができるということを覚えておきましょう