

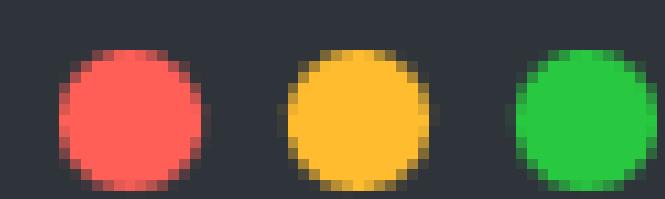
10+ JAVASCRIPT SHORTHAND CODING TECHNIQUES



Hassan Mujtaba | Full Stack Developer
@javascriptwithhassan



THE TERNARY OPERATOR



Longhand

```
const x = 20;
let answer;

if (x > 10) {
    answer = 'greater than 10';
} else {
    answer = 'less than 10';
}
```



Shorthand

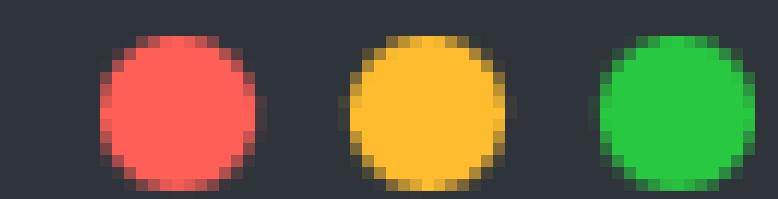
```
const answer = x > 10 ? 'greater than 10' : 'less than 10';
```



Hassan Mujtaba | Full Stack Developer
@javascriptwithhassan

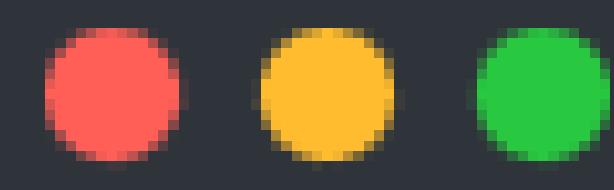


SHORT-CIRCUIT EVALUATION



Longhand

```
let dbHost;  
if (process.env.DB_HOST) {  
  dbHost = process.env.DB_HOST;  
} else {  
  dbHost = 'localhost';  
}
```



Shorthand

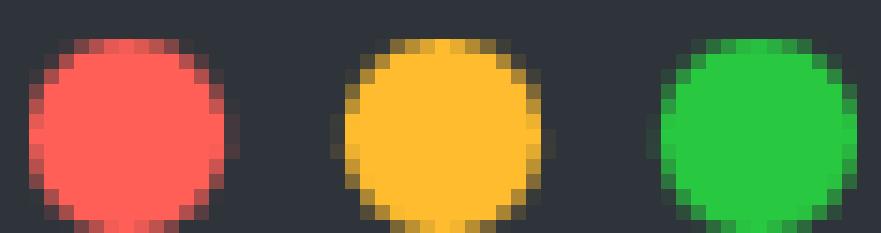
```
const dbHost = process.env.DB_HOST || 'localhost';
```



Hassan Mujtaba | Full Stack Developer
@javascriptwithhassan

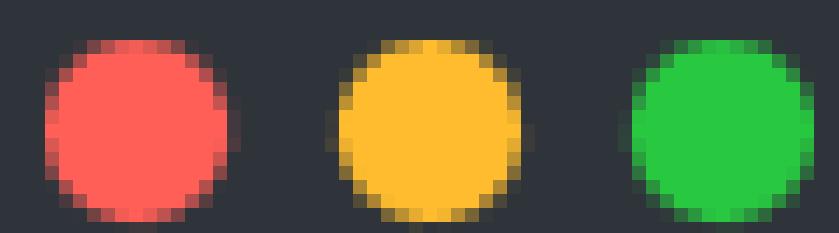


OBJECT PROPERTY



Longhand

```
const x = 1920,  
      y = 1080;  
const obj = { x: x, y: y };
```



Shorthand

```
const obj = { x, y };
```



Hassan Mujtaba | Full Stack Developer
@javascriptwithhassan



ARROW FUNCTIONS



Longhand

```
function sayHello(name) {  
    console.log('Hello', name);  
}  
  
setTimeout(function () {  
    console.log('Loaded');  
, 2000);  
  
list.forEach(function (item) {  
    console.log(item);  
});
```



Shorthand

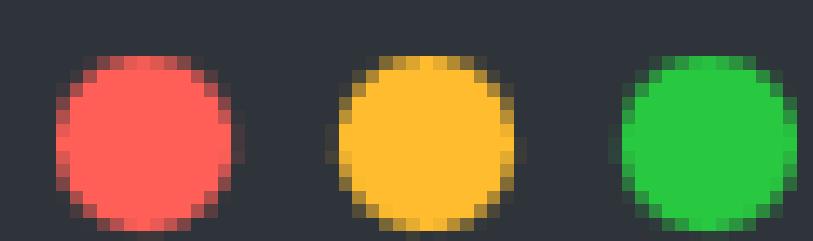
```
sayHello = (name) => console.log('Hello', name);  
  
setTimeout(() => console.log('Loaded'), 2000);  
  
list.forEach((item) => console.log(item));
```



Hassan Mujtaba | Full Stack Developer
@javascriptwithhassan

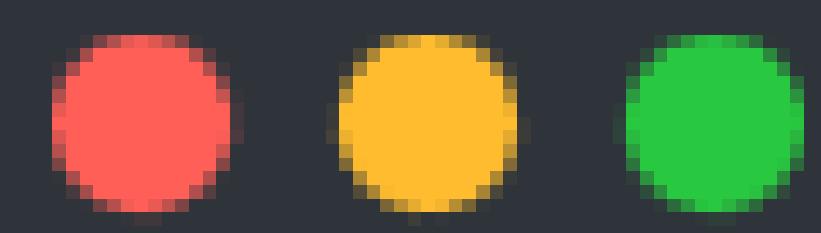


IMPLICIT RETURN



Longhand

```
function calcCircumference(diameter) {  
    return Math.PI * diameter;  
}
```



Shorthand

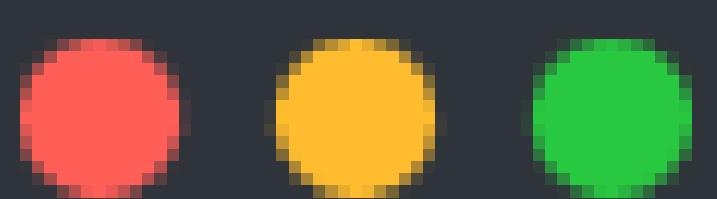
```
calcCircumference = diameter => (  
    Math.PI * diameter;  
)
```



Hassan Mujtaba | Full Stack Developer
@javascriptwithhassan

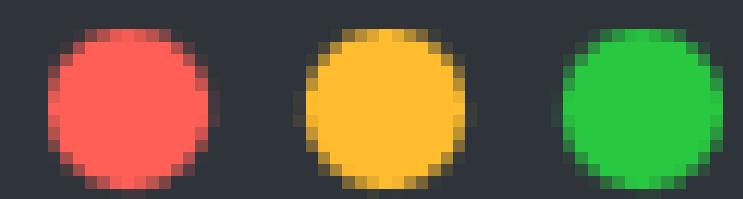


DEFAULT PARAMETER VALUES



Longhand

```
function volume(l, w, h) {  
    if (w === undefined) w = 3;  
    if (h === undefined) h = 4;  
    return l * w * h;  
}
```



Shorthand

```
volume = (l, w = 3, h = 4) => l * w * h;  
  
volume(2); // output: 24
```



Hassan Mujtaba | Full Stack Developer
@javascriptwithhassan



MULTI-LINE STRING

Longhand

```
const lorem =  
  'Lorem ipsum dolor sit amet, consectetur\n\t' +  
  'adipisicing elit, sed do eiusmod tempor incididunt\n\t' +  
  'ut labore et dolore magna aliqua. Ut enim ad minim\n\t' +  
  'veniam, quis nostrud exercitation ullamco laboris\n\t' +  
  'nisi ut aliquip ex ea commodo consequat. Duis aute\n\t' +  
  'irure dolor in reprehenderit in voluptate velit esse.\n\t';
```

Shorthand

```
const lorem = `Lorem ipsum dolor sit amet, consectetur  
adipisicing elit, sed do eiusmod tempor incididunt  
ut labore et dolore magna aliqua. Ut enim ad minim  
veniam, quis nostrud exercitation ullamco laboris  
nisi ut aliquip ex ea commodo consequat. Duis aute  
irure dolor in reprehenderit in voluptate velit esse.`;
```



Hassan Mujtaba | Full Stack Developer
@javascriptwithhassan



ARRAY. FIND

Longhand

```
const pets = [
  { type: 'Dog', name: 'Max' },
  { type: 'Cat', name: 'Karl' },
  { type: 'Dog', name: 'Tommy' },
];

function findDog(name) {
  for (let i = 0; i < pets.length; ++i) {
    if (pets[i].type === 'Dog' && pets[i].name === name) {
      return pets[i];
    }
  }
}
```

Shorthand

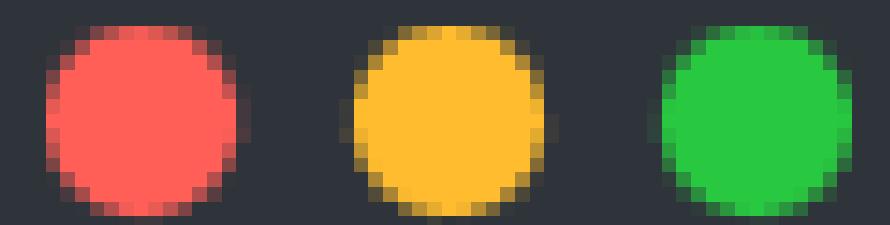
```
pet = pets.find((pet) => pet.type === 'Dog' && pet.name === 'Tommy');
console.log(pet); // { type: 'Dog', name: 'Tommy' }
```



Hassan Mujtaba | Full Stack Developer
@javascriptwithhassan

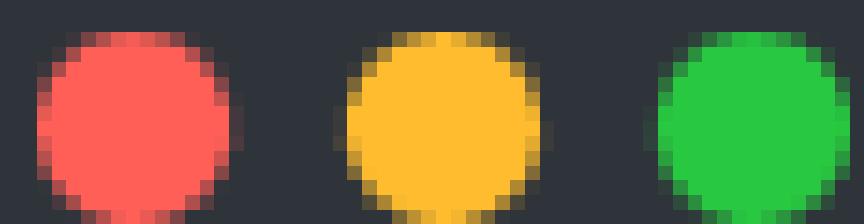


EXPONENT POWER



Longhand

```
Math.pow(2, 3); // 8  
Math.pow(2, 2); // 4  
Math.pow(4, 3); // 64
```



Shorthand

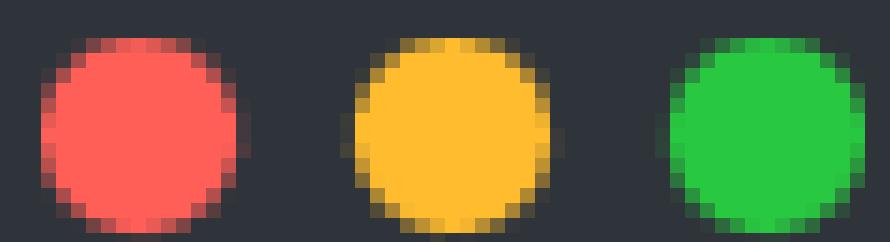
```
2 ** 3; // 8  
2 ** 4; // 4  
4 ** 3; // 64
```



Hassan Mujtaba | Full Stack Developer
@javascriptwithhassan

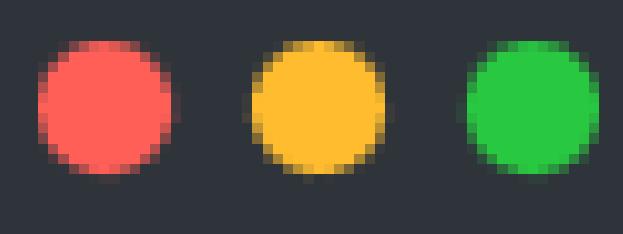


DECIMAL BASE EXPONENTS



Longhand

```
for (let i = 0; i < 10000; i++) {}
```



Shorthand

```
for (let i = 0; i < 1e7; i++) {}
```

```
// All the below will evaluate to true
1 === 1;
1e1 === 10;
1e2 === 100;
1e3 === 1000;
1e4 === 10000;
1e5 === 100000;
```



Hassan Mujtaba | Full Stack Developer
@javascriptwithhassan



DESTRUCTURING ASSIGNMENT

Longhand

```
const observable = require('mobx/observable');
const action = require('mobx/action');
const runInAction = require('mobx/runInAction');

const store = this.props.store;
const form = this.props.form;
const loading = this.props.loading;
const errors = this.props.errors;
const entity = this.props.entity;
```

Shorthand

```
import { observable, action, runInAction } from 'mobx';

const { store, form, loading, errors, entity } = this.props;
```



Hassan Mujtaba | Full Stack Developer
@javascriptwithhassan



Do you find it helpful?
Let me know in the comment section



Hassan Mujtaba | Full Stack Developer



**Follow for more tips and tricks related to
web development**