JavaScript ES6 Cheat Sheet

@codingtute

Arrow function

```
const sum = (a, b) => a + b
console.log(sum(2, 6)) // prints 8
```

Default parameters

```
function print(a = 5) {
 console.log(a)
}
print() // prints 5
print(22) // prints 22
```

Let Scope

```
let a = 3
if (true) {
 let a = 5
 console.log(a) // prints 5
}
console.log(a) // prints 3
```

Const

```
// can be assigned only once
const a = 55
a = 44 // throws an error
```

Multiline string

```
console.log(`
 This is a
 multiline string
`)
```

Template strings

```
const name = 'World'
const message = `Hello ${name}`
console.log(message)
// prints "Hello World"
```

Exponent operator

```
const byte = 2 ** 8
// Same as: Math.pow(2, 8)
```

Spread operator

```
const a = [ 1, 2 ]
const b = [ 3, 4 ]
const c = [ ...a, ...b ]
console.log(c) // [1, 2, 3, 4]
```

String includes()

```
console.log('apple'.includes('pl'))
// prints true
console.log('apple'.includes('tt'))
// prints false
```

String startsWith()

```
console.log('apple'.startsWith('ap'))
//prints true
console.log('apple'.startsWith('bb'))
//prints false
```

String repeat()

```
console.log('ab'.repeat(3))
//prints "ababab"
```

Destructuring array

```
let [a, b] = [3, 7];
console.log(a); // 3
console.log(b); // 7
```

Destructuring object

```
let obj = {
 a: 55,
 b: 44
};
let { a, b } = obj;
console.log(a); // 55
console.log(b); // 44
```

Object property assignment

```
const a = 2
const b = 5
const obj = { a, b }
// Before es6:
// obj = { a: a, b: b }
console.log(obj)
// prints { a: 2, b: 5 }
```

Object.assign()

```
const obj1 = { a: 1 }
const obj2 = { b: 2 }
const obj3 = Object.assign({},
    obj1, obj2)
console.log(obj3)
// { a: 1, b: 2 }
```

Promises with finally

```
promise
 .then((result) => { ··· })
 .catch((error) => { ··· })
 .finally(() => { /* logic
independent of success/error */ })
/* The handler is called when the
promise is fulflled or rejected.*/
```

Object function assignment

```
const obj = {
 a: 5,
 b() {
 console.log('b')
 }
}
obj.b() // prints "b"
```

Object.entries()

```
const obj = {
 firstName: 'FirstName',
 lastName: 'LastName',
 age: 24,
 country: 'India',
};
const entries = Object.entries(obj);
/* returns an array of [key, value]
 pairs of the object passed */
console.log(entries);
/* prints
 [
 ['firstName', 'FirstName'],
 ['lastName', 'LastName'],
 ['age', 24],
 ['country', 'India']
 ]; */
```

Spread operator

```
const a = {
 firstName: "FirstName",
 lastName: "LastName1",
}
const b = {
 ...a,
 lastName: "LastName2",
 canSing: true,
}
console.log(a)
//{firstName: "FirstName", lastName: "LastName1"}
console.log(b)
/* {firstName: "FirstName", lastName: "LastName2",
canSing: true} */
/* great for modifying objects without side
effects/affecting the original */
```

Destructuring Nested Objects

```
const Person = {
 name: "Harry Potter",
 age: 29,
 sex: "male",
 materialStatus: "single",
 address: {
 country: "USA",
 state: "Nevada",
 city: "Carson City",
 pinCode: "500014",
 },
};
const { address : { state, pinCode }, name } = Person;
console.log(name, state, pinCode)
// Harry Potter Nevada 500014
console.log(city) // ReferenceError
```

JavaScript ES6 Cheat Sheet

@codingtute

Arrow function

```
const sum = (a, b) => a + b
console.log(sum(2, 6)) // prints 8
```

Default parameters

```
function print(a = 5) {
 console.log(a)
}
print() // prints 5
print(22) // prints 22
```

Let Scope

```
let a = 3
if (true) {
 let a = 5
 console.log(a) // prints 5
}
console.log(a) // prints 3
```

Const

```
// can be assigned only once
const a = 55
a = 44 // throws an error
```

Multiline string

```
console.log(`
 This is a
 multiline string
`)
```

Template strings

```
const name = 'World'
const message = `Hello ${name}`
console.log(message)
// prints "Hello World"
```

Exponent operator

```
const byte = 2 ** 8
// Same as: Math.pow(2, 8)
```

Spread operator

```
const a = [ 1, 2 ]
const b = [ 3, 4 ]
const c = [ ...a, ...b ]
console.log(c) // [1, 2, 3, 4]
```

String includes()

```
console.log('apple'.includes('pl'))
// prints true
console.log('apple'.includes('tt'))
// prints false
```

String startsWith()

```
console.log('apple'.startsWith('ap'))
//prints true
console.log('apple'.startsWith('bb'))
//prints false
```

String repeat()

```
console.log('ab'.repeat(3))
//prints "ababab"
```

Destructuring array

```
let [a, b] = [3, 7];
console.log(a); // 3
console.log(b); // 7
```

Destructuring object

```
let obj = {
 a: 55,
 b: 44
};
let { a, b } = obj;
console.log(a); // 55
console.log(b); // 44
```

Object property assignment

```
const a = 2
const b = 5
const obj = { a, b }
// Before es6:
// obj = { a: a, b: b }
console.log(obj)
// prints { a: 2, b: 5 }
```

Object.assign()

```
const obj1 = { a: 1 }
const obj2 = { b: 2 }
const obj3 = Object.assign({},
    obj1, obj2)
console.log(obj3)
// { a: 1, b: 2 }
```

Promises with finally

```
promise
 .then((result) => { ··· })
 .catch((error) => { ··· })
 .finally(() => { /* logic
independent of success/error */ })
/* The handler is called when the
promise is fulflled or rejected.*/
```

Object function assignment

```
const obj = {
 a: 5,
 b() {
 console.log('b')
 }
}
obj.b() // prints "b"
```

Object.entries()

```
const obj = {
 firstName: 'FirstName',
 lastName: 'LastName',
 age: 24,
 country: 'India',
};
const entries = Object.entries(obj);
/* returns an array of [key, value]
 pairs of the object passed */
console.log(entries);
/* prints
 [
 ['firstName', 'FirstName'],
 ['lastName', 'LastName'],
 ['age', 24],
 ['country', 'India']
 ]; */
```

Spread operator

```
const a = {
 firstName: "FirstName",
 lastName: "LastName1",
}
const b = {
 ...a,
 lastName: "LastName2",
 canSing: true,
}
console.log(a)
//{firstName: "FirstName", lastName: "LastName1"}
console.log(b)
/* {firstName: "FirstName", lastName: "LastName2",
canSing: true} */
/* great for modifying objects without side
effects/affecting the original */
```

Destructuring Nested Objects

```
const Person = {
 name: "Harry Potter",
 age: 29,
 sex: "male",
 materialStatus: "single",
 address: {
 country: "USA",
 state: "Nevada",
 city: "Carson City",
 pinCode: "500014",
 },
};
const { address : { state, pinCode }, name } = Person;
console.log(name, state, pinCode)
// Harry Potter Nevada 500014
console.log(city) // ReferenceError
```