JavaScript Tips

JavaScript Tips

- Equally

```
if ("300" == 300){
  console.log(true)
}else{
  console.log(false)
}
```

```
if ("300" === 300){
  console.log(true)
}else{
  console.log(false)
}
```

JavaScript Tips

- Promise/Async/Await

```
let myPromise = new Promise(function (resolve, reject) {
  resolve(); // when successful
  reject();  // when error
});

myPromise.then(
  function (value) { console.log("code if successful") },
  function (error) { console.log("code if some error") }
);
```

```
const getData = async () => {
  return new Promise(function (resolve, reject) {
    setTimeout(() => {
      console.log("I'm here 😜")
      resolve(); // when successful
    }, 1000);
  })
}
//== 1st case ==//
console.log(1);
getData();
console.log(2);

//== 2nd case ==//
console.log(1);
await getData();
console.log(2);
```

JavaScript Tips

- Error handling

```
function toUppercase(string) {
  if (typeof string !== "string") {
    throw TypeError("Wrong type given, expected a string");
  }

  return string.toUpperCase();
}

toUppercase(4);
```

```
▶ Uncaught (in promise)          ?editor_console=true:115
TypeError: Wrong type given, expected a string
    at toUppercase (?editor_console=true:115:11)
    at ?editor_console=true:121:1
    at ?editor_console=true:123:3
```

- Try to catch error

```
try {
  toUppercase(4);
} catch (error) {
  console.error(error.message);
  // or log remotely
} finally {
  // clean up
}
```

JavaScript Tips

- **JavaScript ES6 syntax:** ECMAScript 2015 was the second major revision to JavaScript.

```
const person = {
  firstName: "SOK",
  lastName: "SAO"
}

// 1st
const firstName = person.firstName
const lastName = person.lastName
// 2nd
const { firstName, lastName } = person
```

Accessing object's properties

```
const names = ['Jame', "Sok"];
const otherNames = ['Paul', "Sao"];

const allNames = [...names, ...otherNames]
// ["Jame", "Sok", "Paul", "Sao"]
```

Spread array elements into another

```
// String Interpolation
const person = {
  name: "Sok",
  age: 20
}
const message = `The name is ${person.name} and I'm ${person.age} years old`;
```

String interpolation (Backtick)

Spread object properties into another

```
const person = {
  firstName: "Sok",
  lastName: "Sao"
}
const personInfo = {
  address: "Phnom Penh",
  school: "ITC",
  age: 20
}

const completePersonInfo = {
  ...person,
  ...personInfo
}
/*
{
  address: "Phnom Penh",
  age: 20,
  firstName: "Sok",
  lastName: "Sao",
  school: "ITC"
}
*/
```

JavaScript Tips

- **JavaScript ES6 syntax:** ECMAScript 2015 was the second major revision to JavaScript.

*Find

```
let prices = [500, 600, 700, 800, 900, 1000, 1500];
let result = prices.find(price => price > 777);
console.log(result);

// 800
```

*FindIndex

```
let prices = [500, 600, 700, 800, 900, 1000, 1500];
let result = prices.findIndex(function (price) {
    return price == this;
}, 1000);
console.log(result);

// 5
```

*Get indexes only

```
let words = ['Lenovo', 'Tablet', 'Coffee'];
console.log(...words.keys());

// 0 1 2
```

*Get values only

```
let words = ['Lenovo', 'Tablet', 'Coffee'];
console.log(...words.values());

// Lenovo Tablet Coffee
```

JavaScript Tips

- **JavaScript ES6 syntax:** ECMAScript 2015 was the second major revision to JavaScript.

*Map

```
let numbers = [2, 4, 6, 8, 10];

// apply square() function to each item of the numbers list
let square_numbers = numbers.map(function (num) {
  return number * number;
});
console.log(square_numbers);
// Output: [ 4, 16, 36, 64, 100 ]
```

*indexOf

```
let languages = ["Java", "JavaScript", "Python", "JavaScript"];

// get the index of the first occurrence of "JavaScript"
let index = languages.indexOf("JavaScript");
console.log(index);

// Output: 1
```

*includes

```
let languages = ["JavaScript", "Java", "C", "C++", "Python", "Lua"];

let check = languages.includes("Java");
console.log(check); // true
```

JavaScript Tips

- Functions

```
/**
 * Func Delcaration
 */
function makePizza(qty) {
  const pizza = '🍕'.repeat(qty)
  return pizza;
}

const hotPizza = makePizza(6)
console.log(hotPizza); // 🍕🍕🍕🍕🍕🍕
```

```
/**
 * Arrow func
 */
// 1st
const makePizza = (qty) => '🍕'.repeat(qty);

// 2nd
const makePizza = (qty) => {
  return '🍕'.repeat(qty);
};
console.log(makePizza(6))
```

```
/**
 * Func Expression
 */

const makePizza = function(qty){
  return '🍕'.repeat(qty);
}
const hotPizza = makePizza(6)
```

```
/**
 * Invocable Func expression
 */
(function(){
  // to do
  console.log("hey! I'm here👋")
})();
```

Good Luck