# Apply Call Bind

# Class

## //es5

var Person5 = function(name, yearOfBirth, job) {

this.name = name;

This.yearOfBirth = yearOfBirth;

This.job = job;

}

Person5.prototype.calculateAge = function () {

var age = new Date().getFullYear() - this.yearOfbirth;

Console.log(age);

}

var person5 = new Person5(‘john’, 1995, ‘teacher’);

## //es6

class Person6 {

constructor(name, yearOfBirth, job) {

this.name = name;

this.yearOfBirth = yearOfBirth;

this.job = job;

}

calculateAge() {

const age = new Date().getFullYear() - this.yearOfBirth;

console.log(age)

}

}

const person6 = new Person6(‘john’, 1997, ‘teacher’);

# Class and Subclasses

## //ES5

var Person5 = function(name, yearOfBirth, job) {

this.name = name;

this.yearOfBirth = yearOfBirth;

this.job = job;

}

Person5.prototype.calculateAge = function() {

var age = new Date().getFullYear() - this.yearOfBirth;

console.log(age);

}

var Athlete5 = function(name, yearOfBirth, job, olymicGames, medals)

{

Person5.call(this, name, yearOfBirth, job);

this.olymicGames = olymicGames;

this.medals = medals;

}

Athlete5.prototype = Object.create(Person5.prototype);

Athlete5.prototype.wonMedals = function() {

this.medals++;

console.log(this.medals);

}

var john = new Athlete5('john', 1990, 'swimmer', 3, 2)

john.calculateAge();

john.wonMedals();

## //ES6

class Person6 {

constructor (name, yearOfBirth, job) {

this.name = name;

this.yearOfBirth = yearOfBirth;

this.job = job;

}

calculateAge() {

var age = new Date().getFullYear() - this.yearOfBirth;

console.log(age);

}

}

class Athlete6 extends Person6 {

constructor(name, yearOfBirth, job, olympicGames, medals) {

super(name, yearOfBirth, job);

this.olympicGames = olympicGames;

this.medals = medals;

}

wonMedals() {

this.medals++;

console.log(this.medals);

}

}

const mike = new Athlete6('Mike', 1984, 'runner', 4, 3);

mike.calculateAge();

mike.wonMedals();

# Maps

const question = new Map();

question.set('question', 'What is the offical name of the lateste major javascript version?');

question.set(1, 'ES5');

question.set(2, 'ES6');

question.set(3, 'ES2015');

question.set(4, 'ES7');

question.set('correct', 3);

question.set(true, 'Correct answer:D');

question.set(false, 'Wrong, please try again!');

console.log(question.get('question'))

console.log(question.size)

if(question.has(4)) {

question.delete(4);

}

question.forEach((value, key) => console.log(`This is the key : ${key}, and set to the value ${value}`))

for (let [key, value] of question.entries()) {

if (typeof(key) === 'number') {

console.log(`Answer ${key} : ${value}`);

}

}

const ans = parseInt(prompt('Please choose the answer'));

console.log(question.get(ans === question.get('correct')));

# Default Parameters

## //ES5

function SmithPerson(firstName, yearOfBirth, lastName, nationality) {

lastName === undefined ? lastName = 'Smith' : lastName = lastName;

nationality === undefined ? nationality = 'american' : nationality = nationality;

this.firstName = firstName;

this.lastName = lastName;

this.yearOfBirth = yearOfBirth;

this.nationality = nationality;

}

## //ES6

function SmithPerson(firstName, yearOfBirth, lastName = 'Smith', nationality = 'American') {

this.firstName = firstName;

this.lastName = lastName;

this.yearOfBirth = yearOfBirth;

this.nationality = nationality;

}

var john = new SmithPerson('John', 1990);

var emily = new SmithPerson('Emily', 1983, 'Diaz', 'spanish');

# Rest parameters Rest参数

## //ES5

function isFullAge5() {

//console.log(arguments);

var argsArr = Array.prototype.slice.call(arguments);

argsArr.forEach(function(cur) {

console.log((2019 - cur) >= 18);

})

}

isFullAge5(1990, 2011, 2000);

function isFullAge5(limit) {

//console.log(arguments);

var argsArr = Array.prototype.slice.call(arguments, 1);

argsArr.forEach(function(cur) {

console.log((2019 - cur) >= limit);

})

}

isFullAge5(20, 1990, 2011, 2000);

## //ES6

function isFullAge6(...years) {

// console.log(years);

years.forEach(cur =>console.log((2019 - cur) >= 18));

}

isFullAge6(1990, 2011, 2000)

function isFullAge6(limit, ...years) {

// console.log(years);

years.forEach(cur =>console.log((2019 - cur) >= limit));

}

isFullAge6(16, 1990, 2011, 2000)

# Spread operator

function addfourAges(a, b, c, d) {

return a + b + c + d;

}

var sum1 = addfourAges(13, 25, 46, 33);

console.log(sum1);

var age = [13, 22, 44, 3];

var sum2 = addfourAges.apply(null,age)

console.log(sum2);

var sum3 = addfourAges(...age);

console.log(sum3);

var familyJohn = ['bob', 'Mike', 'lucy'];

var familySmith = ['Jane', 'Lily', 'Mark'];

var bigFamily = [...familyJohn, ...familySmith];

console.log(bigFamily);

const b = document.querySelector('h1');

const c = document.querySelectorAll('.box');

const d = [b, ...c];

Array.from(d).forEach(cur => cur.style.color = 'purple');

# Arrarys

## Array.from

//ES5

var boxesArr5 = Array.prototype.slice.call(boxes);

boxesArr5.forEach(function(el,index) {

el.style.background = 'dodgerblue';

});

//ES6

const boxesArr6 = Array.from(boxes);

Array.from(boxes).forEach(el => el.style.backgroundColor = 'dodgerblue');

## for(const cur of boxArr){}

//ES5

/\*

for(var i = 0; i < boxesArr5.length; i++) {

if(boxesArr5[i].className === 'box blue') {

continue; //跳出该次循环剩下的不执行进入下一循环

}

boxesArr5[i].textContent = "i'm changed to blue!";

}

//ES6

for(const cur of boxesArr6) {

if (cur.className === 'box blue') {

continue;

}

cur.textContent = 'Hello! blue!'

};

## findIndex && find

//ES5

var ages = [4, 9, 17, 16, 22, 8];

var full = ages.map(function(cur) {

return cur >= 18;

})

console.log(full);

console.log(full.indexOf(true));

console.log(ages[full.indexOf(true)]);

//ES6

console.log(ages.findIndex(cur => cur >= 18));

console.log(ages.find(cur => cur >= 18));

# Destrucuring

## //ES5

var john = ['John', 26];

var name = john[0];

var age = john[1];

## //ES6

const [name, age] = ['john', 26];

console.log(name);

console.log(age);

const obj = {

firstName: 'John',

lastName: 'Smith'

};

const {firstName, lastName} = obj;

console.log(firstName);

console.log(lastName);

const {firstName: a, lastName: b} = obj;

console.log(a);

function calcAgeRetirement(year) {

const age = new Date().getFullYear() - year;

return [age, 55 - age];

}

const [age2, retirement] = calcAgeRetirement(1997);

console.log(retirement);

console.log(age2);

# Arrow function

const years = [1990, 1965, 1985, 1937];

## //ES5

var ages5 = years.map(function(el) {

return 2017 - el;

});

console.log(ages5);

## //ES6

let ages6 = years.map(el => 2016 - el);

console.log(ages6);

ages6 = years.map((el, index) => `Age element ${index + 1 }: ${2016-el}.`);

console.log(ages6);

ages6 = years.map((el, index) => {

const now = new Date().getFullYear();

const age = now - el;

return `Age element ${index + 1 }: ${age}.` //加｛｝默认不return

});

console.log(ages6);

# Arrow functions 2 THIS

## //ES5 var self = this

var box5 = {

color: 'green',

position: 1,

clickMe: function() {

var self = this;

document.querySelector('.green').addEventListener('click', function() {

alert('this box color is:' + self.color + ' and it on the ' + self.position);

})

}

}

//box5.clickMe();

## //ES6

const box6 = {

color: 'green',

position: 1,

clickMe: function() {

document.querySelector('.green').addEventListener('click',() => {

alert('this box color is:' + this.color + ' and it on the ' + this.position);

});

}

}

box6.clickMe();

/\*

const box66 = {

color: 'green',

position: 1,

clickMe: () => {

document.querySelector('.green').addEventListener('click',() => {

alert('this box color is:' + this.color + ' and it on the ' + this.position);

});

}

}

box66.clickMe(); //undefind

function Person(name) {

this.name = name;

}

//ES5

Person.prototype.myFriends5 = function(friends) {

var arr = friends.map(function(el) {

return this.name + ' is friends with ' + el;

}.bind(this))

console.log(arr);

}

var friends = ['Bodb', 'Jane', 'Smith'];

new Person('john').myFriends5(friends);

//ES6

Person.prototype.myFriends6 = function(friends) {

var arr = friends.map(el =>`${this.name} is friends with ${el} ` )

console.log(arr);

}

new Person('Mike').myFriends6(friends);

# Strings

let firstName = 'John';

let lastName = 'Simth';

const yearOfBirth = 1990;

function calcAge(year) {

return 2016 - year;

}

## //ES5

console.log('This is ' + firstName + ' ' + lastName + '. He was born in ' + yearOfBirth + '. Today, he is '+ calcAge(yearOfBirth) + ' years old.');

## //ES6

console.log(`This is ${firstName} ${lastName}. He was born in ${yearOfBirth}. Today he is ${calcAge(yearOfBirth)} years old.`);

const n = `${firstName} ${lastName}`; //John Simth

console.log(n.startsWith('J')); // true

console.log(n.endsWith('th')); true

console.log(n.includes('im')); true

console.log(`${n} `.repeat(5));

# Blocks and IIFEs

## // ES6

{

const a = 1;

let b = 2;

var c = 3;

}

console.log(b);

## //ES5

(function() {

var c = 3;

})();

console.log(c);

# lecture: let and const

//es5

var name5 = 'Jane Smith';

var age5 = 23;

name5 = 'Jane Miller';

console.log(name5);

//es6

const name6 = 'Jane Smith';

let age6 = 26;

age6 = 27;

console.log(age6);

//ES5

function passtestFunction5(passtest) {

if(passtest) {

console.log(name); // hosting and set undefine

var name = 'John';

var yearOfBirth = 1990;

}

console.log(name + ', born in ' + yearOfBirth + ' is now offically allowed to dirve a car.');

}

passtestFunction5(true);

//ES6

function passtestFunction6(passtest) {

//console.log(name); // error: tempral-dead zone

let name;

const yearOfBirth = 1990;

if(passtest) {

name = 'John';

}

console.log(name + ', born in ' + yearOfBirth + ' is now offically allowed to dirve a car.');

}

passtestFunction6(true);

let i = 23;

for (let i = 0; i < 5; i++) {

console.log(i); 0 1 2 3 4

}

console.log(i); //23

var x = 23;

for (var x = 0; x < 5; x++) {

console.log(x); 0 1 2 3 4

}

console.log(x); //5