# 背景前提

## 什么是expression

解释1：An expression is a phrase of JavaScript that can be evaluated to produce a value(表达式是可以计算出值的JavaScript短语).——《JavaScript: The Definitive Guid》p135

定义2：An expression is any valid unit of code that resolves to a value.(表达式是可以解析为值的任何有效代码单元)— —MDN(https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Expressions\_and\_Operators#expressions)

## 什么是statement

定义1： Statements are JavaScript sentences or commands.Expressions are evaluated to produce a value, but statements are executed to make something happen. One way to “make something happen” is to evaluate an expression that has side effects. ——《JavaScript: The Definitive Guid》p197

## 方法

1. 定义

定义在对象内部的函数称为方法(method)。

方式1：常规写法

let square = {

area: function () { return this.side \* this.side; },

side: 10

};

square.area() // => 100

方式2：简写(前提：当属性的值是一个函数的时候才可以简写，其它情况不能简写。方式：the function keyword and the colon are omitted)

let square = {

area () { return this.side \* this.side; },

side: 10

};

square.area() // => 100

1. 参考

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/Method_definitions>

# 函数表达式(function expression)

## 定义

The function keyword can be used to define a function inside an expression.So the expression called function expression。

## 语法

function [name]([param1[, param2[, ..., paramN]]]) {

statements

}

## 示例

const getRectArea = function(width, height) {

return width \* height;

};

console.log(getRectArea(3, 4));

// expected output: 12

// Function expressions can also be used as arguments to

other functions:

[3,2,1].sort(function(a,b) { return a-b; });

// Function expressions are sometimes defined and immediately

invoked:

let tensquared = (function(x) {return x\*x;}(10));

## 参考

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/function>

# 函数声明(function declaration)

## 定义

函数声明是定义函数的一种方式。

## 语法

function name([param[, param,[..., param]]]) {

[statements]

}

注：函数声明时函数名词必须存在。

# 箭头函数表达式(arrow function expression)

## 定义

## 基础语法

## 高级语法

params => ({foo: "a"}) // returning the object {foo: "a"}

## 示例

const sum = (x, y) => { return x + y; };

变形1（如果只有单个返回语句，可以省略花括号、分号、return关键字）：

const sum = (x, y) => x + y;

变形2（如果只有一个参数，可以省略圆括号）：

const polynomial = x => x\*x + 2\*x + 3;

变形3（如果没有参数，必须保留圆括号）：

const constantFunc = () => 42;

## 参考

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/Arrow\_functions

# 问题

## 函数声明和带名字的函数表达式怎么区分？

答：根据上下文进行判断。函数声明必须带有标示符（Identifier）（就是大家常说的函数名称），而函数表达式则可以省略这个标示符。

参考：<https://www.cnblogs.com/tomxu/archive/2011/12/29/2290308.html>