

函数

arguments 函数内置的实参集合

任意数求和（执行函数的时候，传递N个值实现求和）

```
/*
 * 任意数求和
 * 1. 传递实参的个数不定
 * 2. 传递的值是否为有效数字不定
 * => 把传递的有效数字进行相加求和
 *
 * arguments: 函数内置的实参集合
 * 1. 类数组集合，集合中存储着所有函数执行时，传递的实参信息
 * 2. 不论是否设置形参，arguments都存在
 * 3. 不论是否传递实参，arguments也都存在
 * arguments.callee: 存储的是当前函数本身（一般不用的，JS严格模式下禁止使用这些属性）
 */
function sum() {
  let total = null;
  for (let i = 0; i < arguments.length; i++) {
    // 获取的每一项的结果都要先转换为数字（数学运算）
    let item = Number(arguments[i]);
    // 非有效数字不加
    if (isNaN(item)) {
      continue;
    }
    total += item;
  }
  return total;
}

let total = sum(10, 20, 30, 40);
console.log(total);

total = sum(10, 20);
console.log(total);

total = sum(10, 20, '30');
console.log(total);

total = sum(10, 'AA');
console.log(total);
```

arrow function 箭头函数

简单

```

/* function sum(n, m) {
    return n + m;
} */

// 改写成箭头函数
let sum = (n, m) => {
    return n + m;
};
// 如果函数体中只有一行RETURN，可以省略RETURN和大括号，一行搞定
let sum = (n, m) => n + m;

```

```

function fn(n) {
    return function (m) {
        return n + m;
    }
}
// 改写成箭头函数
let fn = n => m => n + m;

```

形参赋值默认值：当没有给形参传递实参的时候，执行默认值

```

/* function sum(n, m) {
    if (typeof n === 'undefined') {
        n = 0;
    }
    if (typeof m === 'undefined') {
        m = 0;
    }
    return n + m;
} */
// 形参赋值默认值：当没有给形参传递实参的时候，执行默认值
let sum = (n = 0, m = 0) => n + m;
sum();
sum(10);

```

箭头函数中没有ARGUMENTS

```

/* let sum = () => {
    console.log(arguments); //=>Uncaught ReferenceError: arguments is not
defined 箭头函数中没有ARGUMENTS
}; */

// 但是我们可以使用剩余运算符获取到传递的实参集合（它是数组，比ARGUMENTS更好玩）
let sum = (...arg) => eval(arg.join('+'));

```

箭头函数中的THIS某些场景也是方便我们操作的

.....

考试安排

基础课结束后考试

- 上机实战测试 (稳过)
 - 选项卡
 - ...
- 基础知识笔试 (≥ 70 分)
 - 第一周和第二周基础知识
 - 留的作业中也会抽出一部分题

上机测试: 周三\周四\周五\周六 三天晚上

笔试题: 周六下午开始 (补考时间周日)
