

日期和MATH

题目

- 获取xxxx-xx-xx格式的日期
- 获取随机数，要求是长度一致的字符串
- 写一个能遍历对象和数组的通用forEach函数

日期

```
console.log(Date.now()); // 1582544029837 ms 当前时间
let dt=new Date()
console.log(dt.getTime()); // 也是毫秒，但是他是dt被new出来时的毫秒
console.log(dt.getFullYear()); //2020
console.log(dt.getMonth()); //1 为什么是一月份？因为类似数组，只有0-11
console.log(dt.getDate()); //24
console.log(dt.getHours()); //19
console.log(dt.getMinutes()); //38
console.log(dt.getSeconds()); //16
```

知识点

- 对象API
 - for (x in y)
- Math
 - 最常用的是 : Math.random()

数组API

```
let obj={
  x:100,
  y:200,
  z:300
}
// 这里给obj 加一个迷惑条件，让key in obj 时能得到 a
Object.prototype.a=100
console.log(obj) //{ x: 100, y: 200, z: 300 }
// 为什么这里 obj 没有打印出来 a 呢？
// 因为for...in 循环只遍历可枚举属性（包括它的原型链上的可枚举属性）
for(let key in obj){
  console.log(key)
  // 判断obj这个对象是否有key这个属性
  if(obj.hasOwnProperty(key)){
    console.log(key,obj[key])
  }
}
```

```
let array=[1,2,3,4]
array.forEach(element => {
  element+=element
});
console.log(array); // [1,2,3,4] forEach不会对原数组造成影响
```

- forEach 遍历所有元素

```
const array1 = [1, 30, 39, 29, 10, 41];
console.log(array1.every(item=>item>40)); // false
// every返回一个boolean 如果array为空，会返回true
// 表示一个数组内所有的元素通过函数调用后判断是否符合条件some
```

```
const array2 = [1, 30, 39, 29, 10, 41];
console.log(array2.some(item=>item>40)); // true
// 和 every 刚好相反 只需要有一个元素符合条件就返回ture
```

- every 判断所有元素是否都符合条件

- some 判断是否至少有一个元素符合条件

```
const array1 = [1, 30, 4, 21, 100000];
array1.sort();
console.log(array1);
// expected output: Array [1, 100000, 21, 30, 4]
// 以上可以看出 sort 在没有传入排序条件时，会按照unicode顺序进行排列

const array2 = [7,2,6,4];
console.log(array2.sort((item1,item2)=>{
  // 从小到大排序
  return item1-item2

  // 从大到小排序
  // return item2-item1
}));
```

- sort 排序

```
const array1 = [1, 4, 9, 16];
```

```
// map对每个元素进行遍历，然后将处理后的元素生成一个新的数组，原数组不变
```

```
console.log(array1.map(x => x * 2));[ 2, 8, 18, 32 ]
console.log(array1);[ 1, 4, 9, 16 ]
```

- map 对元素进行重新组装，生成一个新的数组

```
const words = ['spray', 'limit', 'elite', 'exuberant'];
```

```
const result = words.filter(item => item.length > 6);
```

```
console.log(result); // [ 'exuberant' ]
```

- filter 过滤符合条件的元素