

一. 正则表达式。

```
1. /^1[3-9]\d{9}$/
```

电话号码从 1 开始 (^1)，第二位匹配 3-9([3-9])，剩下的 9 位({9})是进制任意数字 (\d)

```
2. /^[0-9A-Za-z\-\_\.\+])@([0-9a-z]+\.[a-z]{2,3}(\.[a-z]{2})?)/g
```

开始必须是一个或者多个单词字符或者是-，加上@，然后又是一个或者多个单词字符或者是-。然后是点“.”和单词字符和-的组合，可以有一个或者多个组合。

```
3. /\b([a-z]+\s+\1\b)/ig
```

\b 确保为两个单词，i 不区分大小写所以 a-z 即可，\s 确保有空格，g 全局匹配。

二. 继承。

1.原型链继承：父类的实例作为子类的原型

优点：

简单易于实现，父类的新增的实例与属性子类都能访问

缺点：

可以在子类中增加实例属性，如果要新增加原型属性和方法需要在 new 父类构造函数的后且无法实现多继承。

创建子类实例时，不能向父类构造函数中传参数

2.构造函数继承

复制父类的实例属性给子类

优点：

解决了子类构造函数向父类构造函数中传递参数

可以实现多继承

缺点：

方法都在构造函数中定义，无法复用

不能继承原型属性/方法，只能继承父类的实例属性和方法

3. Object.create 继承

接收两个参数，第一个表示要继承的对象，第二个参数表示也是一个对象，用于对新创建的对象进行初始化

三. Map、Set、Array 之间的区别和使用。

Map：类似二维数组，为 Map (key, value)。其值改变不会对原对象的值造成改变。

Set：也类似于数组，但其中的成员不可重复。

Array：即数组，对储存的成员无要求，可重复。