1. **面向对象：**

**面向过程**：分析出解决问题所需要的步骤，然后用函数把这些步骤一步一步实现，使用的时候依次调用即可；

**面向对象**：把构成问题事务分解成各个对象，建立对象的目的不是为了完成一个步骤，而是为了描述某个事物在整个解决问题的步骤中的行为。面向对象有三个基本特征：封装、继承、多态；

**封装**：就是把客观事物封装成抽象的类，并且类可以把自己的数据和方法让可信的类或对象操作，对不可信的进行信息隐藏；

**继承**：面向对象编程（OOP）的主要功能就是继承，继承就是一种能力，即使用现有类的所有功能，并且无需重新编写原来的类就可以对其进行扩展；通过继承创建的新类称为子类或派生类，被继承的类称为父类或基类；

OO开发范式：划分对象→抽象类→继承和合成→实例化；

**多态**：允许将子类类型的指针赋值给父类类型的指针；两种方式：覆盖、重载；覆盖就是子类重新定义父类的虚函数，重载就是允许存在多个同名函数但是参数不同；

封装可以隐藏实现细节，使得代码模块化；继承可以扩展已存在的代码模块；作用都是为了——代码重用，多态则是为了接口重用，保证继承和派生的时候正确调用。

1. **闭包：**

函数套函数，外函数return内函数，这样可以访问到外部函数定义的变量供内部函数使用；如：

function foo(x) {

    var tmp = 3;  
    return function (y) { // 这个匿名函数就是闭包  
        alert(x + y + (++tmp));  
    }  
}  
var bar = foo(2); // bar 现在是一个闭包  
bar(10);

这里return的内部函数访问到了foo()函数内声明的变量，而直接访问foo()函数内的变量却是不行的。

**变量作用域**：全局变量和局部变量；函数内部可以直接读取全局变量，函数外部无法读取函数内用var声明的局部变量，即Javascript语言特有的“链式作用域”结构，子函数可以依次向上使用父函数的变量，反之则不行；有时需要在函数外部使用函数内的变量，可以在内函数中使用外函数的变量，而把内函数作为外函数的return值，这时候内函数就是闭包。

闭包的用途：读取函数内部变量；变量始终保持在内存中；

**注意点**：

变量保存在内存中，内存消耗大不能滥用，否则会造成网页性能问题，在IE中还可能导致内存泄漏，解决方法就是退出函数之前，将不使用的局部变量全部删除。

闭包会在父函数外部改变其变量的值；