1. 解释一下Rigidbody组件中的Is kinematic参数在什么情况下使用？答：RigidBody组件的Is Kinematic属性打上勾（设为true，使其不受物理引擎驱动，Wall是为了防止其移动，Person是为了防止其受到力不断旋转）。Is  kinematic 是否开启动力学，开启此项，物体不受力的作用。

2.编写一个脚本对刚体的几种常用方法进行测试。(脚本源码及其注释)

答：using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class rigidbodyTest : MonoBehaviour {

     private Rigidbody \_rigidbody;

     // Use this for initialization

     void Start ()

     {

        \_rigidbody = transform.GetComponent<Rigidbody>();

    }

    // Update is called once per frame

    void Update ()

    {

        if (Input.GetKeyDown(KeyCode.Q))

        {

            \_rigidbody.velocity = new Vector3(1,1,1); //给其一个速度矢量

           // \_rigidbody.position = new Vector3(5,5,5);  //给其一个位置

        }

        if (Input.GetKeyDown(KeyCode.W))

        {

            \_rigidbody.Sleep();/制性使刚体休眠，不动了;休眠是性能优化的一个措施，物理引擎不会处理处于休眠状态的刚体；

            //刚体在以下情况会被唤醒：1,其他刚体碰撞器作用于休眠刚体。2，被其他刚体通过移动的关节连接

            //3，修改了刚体的属性。4，添加外力时

        }

        if (Input.GetKeyDown(KeyCode.E))

        {

            \_rigidbody.WakeUp(); /制唤醒一个刚体

        }

        if (Input.GetKeyDown(KeyCode.A))

        {

            \_rigidbody.MovePosition(new Vector3(10,100,200));//给其一个位置

        }

        if (Input.GetKeyDown(KeyCode.S))

        {

            \_rigidbody.freezeRotation = true; //开启則  刚体的XYZ轴全部冻结

        }

        if (Input.GetKeyDown(KeyCode.D))

        {

          \_rigidbody.constraints = RigidbodyConstraints.FreezePositionY; //选择性冻结某一轴

        }

        if (Input.GetKeyDown(KeyCode.F))

        {

            \_rigidbody.AddExplosionForce(50,new Vector3(0,0,0),20); //添加一个爆炸力

        }

        if (Input.GetKeyDown(KeyCode.R))

        {

            \_rigidbody.AddForce(Vector3.forward\*3,ForceMode.Acceleration); //沿着某一方向给刚体添加一个力

        }

        if (Input.GetKeyDown(KeyCode.T))

        {

            \_rigidbody.AddTorque(transform.forward\*10); //沿着某一方向添加一个扭矩

        }

        //\_rigidbody.transform.Rotate(transform.up, Time.deltaTime); //基于transform的旋转

        //\_rigidbody.angularVelocity = transform.right \* Time.deltaTime;//基于刚体的旋转

    }

    private void OnTriggerEnter(Collider other)

    {

        print(11);

    }

    private void OnTriggerStay(Collider other)

    {

        print(22);

    }

    private void OnTriggerExit(Collider other)

    {

        print(33);

    }

    private void OnCollisionEnter(Collision other)

    {

        print(other.relativeVelocity); //两个碰撞物体的相对线性速度

    }

}

3.了解Unity 3D游戏引擎自带的规则碰撞器,并导入一个模型为其添加合适的碰撞器(回答有哪些碰撞器？）

答：Box Collider ：盒子膨胀器。  
 Sphere Collider： 球体碰撞器。  
 Capsult Collider：胶囊碰撞器。  
 Mesh Collider：网格碰撞器。  
 Wheel Collider：车轮碰撞