

# ROBLOX SCRIPTS

## M1L5 – 12

Script para cambiar el **material** de una parte:

```
function button1()  
    game.Workspace.Button1.Material = Enum.Material.Neon  
end  
game.Workspace.Button1.Touched:Connect(button1)
```

---

Script para cambiar el **color** de una parte:

```
game.Workspace.Button1.BrickColor = BrickColor.new('Lime green')
```

---

Script para cambiar la **transparencia** de una parte:

```
game.Workspace.Button1.Transparency = 0.5
```

---

Script para cambiar la **Health** al tocar un **objeto** (dynamite):

```
--[[-- El script comprueba si una pieza de Dynamite toca con otra pieza  
de Humanoid y cambia la Health del objeto Humanoid a 0. --]]--
```

```

function onTouchBam(who)
    if who.Parent.Humanoid then
        who.Parent.Humanoid.Health = 0
    end
end

game.Workspace.Dynamite.Touched:Connect(onTouchBam)

```

### M1L7 – 20

**Script que puede ser usado para cambiar la hora del día:**

```

B=0
while true do
    game.Lighting.ClockTime=B
    B = B+0.1
    if B > 24 then B = 0 end
    wait()
end

```

---

**Script que hace que las partes brillen y parpadeen:**

```

while true do
    game.workspace.Part1.PointLight.Enabled = false
    game.workspace.Part2.SpotLight.Enabled = false
    game.workspace.Part3.SurfaceLight.Enabled = false
    wait(1)
    game.workspace.Part1.PointLight.Enabled = true
    game.workspace.Part2.SpotLight.Enabled = true
    game.workspace.Part3.SurfaceLight.Enabled = true
    wait(1)
end

```

---

### Script que enciende la linterna cuando empieza la oscuridad:

```
B=0
while true do game.Lighting.ClockTime=B
  B=B+0.1
  if B > 24 then B=0 end
  if B > 16 then game.workspace.Sphere.SpotLight.Enabled = true end
  if B < 6 then game.workspace.Sphere.SpotLight.Enabled = true end
  if B > 6 and B < 16 then game.workspace.Sphere.SpotLight.Enabled
= false end
  wait()
end
```

### M1L9 – 28

#### Un script donde el fuego se enciende en un tiempo específico:

```
while true do
  game.Workspace.Torch.Fire.Enabled = true
  wait(3)
  game.workspace.Torch.Fire.Enabled = false
  wait(2)
end
```

---

#### Un script donde el fuego se enciende cuando el jugador se aproxima:

```
function onTouch(who)
  game.Workspace.Torch.Fire.Enabled = true
end
game.workspace.Area.Touched:Connect(onTouch)
```

---

**Un script donde el fuego se enciende cuando el jugador se aproxima y se extingue cuando se aleja:**

```
function onTouch(who)
    game.Workspace.Torch.Fire.Enabled = true
end
function onTouchEnded(who)
    wait(3)
    game.Workspace.Torch.Fire.Enabled = false
end
game.workspace.Area.Touched:Connect(onTouch)
game.workspace.Area.TouchEnded:Connect(onTouchEnded)
```

---

**Un script donde el jugador explota cuando se toca una parte:**

```
function onTouchBam(who)
    who.Parent.Humanoid.Health=0
end
game.workspace.Torch.Touched:Connect(onTouchBam)
```

---

**Un script en el que cuando un jugador toca una parte, un modelo aparece:**

```
game.workspace.Model.Parent=game.ServerStorage
function onTouchPart(who)
    game.ServerStorage.Model.Parent=game.Workspace
end
game.workspace.Part.Touched:Connect(onTouchPart)
```

---

**Un script en el que cuando un jugador toca una parte, se destruye:**

```
function onTouchPartD(who)
    game.Workspace.Part:Destroy()
end
game.workspace.Part.Touched:Connect(onTouchPartD)
```

---

**Un script en el que cuando un jugador toca el modelo, se destruye:**

```
local model = game.Workspace.Model
local function onModelTouched(part)
    if part:IsDescendantOf(model) then return end
    print(model:GetFullName() .. " was touched by " .. part:GetFullName())
    model:Destroy()
end
for _, child in pairs(model:GetChildren()) do
    if child:IsA("BasePart") then
        child.Touched:Connect(onModelTouched)
    end
end
```

## **M1L10 – 31**

### **Script que hace disminuir la salud del jugador**

```
D = 100
function B(C)
    if C and C.Parent:FindFirstChild("Humanoid") then
        D = D - 5
        C.Parent.Humanoid.Health = D
        if C.Parent.Humanoid:GetState() == Enum.HumanoidStateType.Dead then
            D = 100
        end
    end
end
```

```

        wait(1)
    end
end
end
game.Workspace.Trap.Touched:Connect(B)

```

---

### Script para crear un teletransportador

```

local function onHit(hit)
    if hit and hit.Parent:FindFirstChild("Humanoide") then
        hit.Parent:MoveTo(script.Parent.Kuda.Position)
    end
end
game.workspace.Teleport.Touched:Connect(onHit)

```

---

### Script para crear una escalera

```

function Stairs()
    game.Workspace.Part1.CanCollide=false
    game.Workspace.Part1.Transparency=1
    wait(1)
    game.Workspace.Part2.CanCollide=false
    game.Workspace.Part2.Transparency=1
    wait(1)
    game.Workspace.Part3.CanCollide=false
    game.Workspace.Part3.Transparency=1
    wait(1)
    game.Workspace.Part4.CanCollide=false
    game.Workspace.Part4.Transparency=1
    wait(1)
    game.Workspace.Part1.CanCollide=true
    game.Workspace.Part1.Transparency=0
    wait(1)
    game.Workspace.Part2.CanCollide=true
    game.Workspace.Part2.Transparency=0
    wait(1)
    game.Workspace.Part3.CanCollide=true
    game.Workspace.Part3.Transparency=0
    wait(1)
    game.Workspace.Part4.CanCollide=true

```

```
game.Workspace.Part4.Transparency=0
end
game.Workspace.Part1.Touched:Connect(Stairs)
```

---

## Script que habilita el salto doble

```
local UserInputService = game:GetService("UserInputService")
local character = script.Parent

local humanoid = character:WaitForChild("Humanoid")
local doubleJumpEnabled = false
--función que comprueba lo que está haciendo el personaje: Saltando o Cayendo, luego permite o deniega el salto
function CheckState(oldState, newState)
    if newState == Enum.HumanoidStateType.Jumping then
        --Si está saltando y no hizo un salto doble, entonces
        if not doubleJumpEnabled then
            wait(.2)
            if humanoid:GetState() == Enum.HumanoidStateType.Freefall then
                doubleJumpEnabled = true
            end
        end
        --de lo contrario, si el jugador aterrizó, entonces puede hacer el salto doble de nuevo
    elseif newState == Enum.HumanoidStateType.Landed then
        doubleJumpEnabled = false
    end
end
--función que comprueba si la Barra espaciadora es presionada y organiza un salto doble.
function GetJump(inputObject)
    if inputObject.KeyCode == Enum.KeyCode.Space then
        if doubleJumpEnabled then
            --Si el humanoide no está en estado de salto, entonces cambiamos su estado a saltar
            if humanoid:GetState() ~= Enum.HumanoidStateType.Jumping then
                humanoid:ChangeState(Enum.HumanoidStateType.Jumping)
                spawn(function()
                    doubleJumpEnabled = false
                end)
            end
        end
    end
end
humanoid.StateChanged:Connect(CheckState)
UserInputService.InputBegan:Connect(GetJump)
```

---

## Script que hace que caigan pelotas del cielo.

```
debounce = true
```

```

function Balls(any_touch)
    local ch = any_touch.Parent:FindFirstChild("HumanoidRootPart")
    if ch and debounce == true then
        debounce = false
        for k = 0, 100, 1 do
            local X = math.random(-20, 20)
            local Z = math.random(-20, 20)
            local detal = Instance.new("Part", workspace)
            local human = any_touch.Parent.HumanoidRootPart
            local X0 = human.Position.X
            local Y0 = human.Position.Y
            local Z0 = human.Position.Z
            detal.Position = Vector3.new(X0 + X, Y0 + 100, Z0 + Z)
            detal.BrickColor = BrickColor.Random()
            detal.Size = detal.Size * math.random() * 1.5
            detal.Shape = "Ball"
            wait(0.1)
            debounce = true
        end
    end
end

game.Workspace.Button.Touched:Connect(Balls)

```