```
function begruessung()
    print("hello")
end

cls()
begruessung()
print("world")
```

```
function _init()
    ball_x = 20
    cls()
end

function _update()
    ball_x = ball_x + 1
end

function _draw()
    cls()
    circfill(ball_x,64,2,10)
end
```

```
function _init()
    ball_x = 20
    ball_dx = 2
    ball_radius = 2
    cls()
end

function _update()
    ball_x = ball_x + ball_dx
    check_bounce()
end

function check_bounce()
    if ball_x+ball_radius > 127 then
        ball_dx = -ball_dx
    end
    if ball_x-ball_radius < 0 then
        ball_dx = -ball_dx
    end
end

function _draw()
    cls()
    circfill(ball_x,64, ball_radius,10)
end
```

```
function _init()
    ball_x = 20
    ball_y = 64
    ball_dx = 1
    ball_dy = -1
    ball_radius = 2
    cls()
end

function _update()
    ball_x = ball_x + ball_dx
    ball_y = ball_y + ball_dy

    check_bounce()
end

function check_bounce()
    -- x-achse
    if ball_x+ball_radius > 127 or ball_x-ball_radius < 0 then
        ball_dx = -ball_dx
    end

    -- y-achse
    if ball_y+ball_radius > 127 or ball_y-ball_radius < 0 then
        ball_dy = -ball_dy
    end
end

function _draw()
    cls()
    circfill(ball_x,ball_y,ball_radius,10)
end
```

```
function _init()
    -- ball
    ball_x = 20
    ball_dx = 2
    ball_y = 64
    ball_dy = -2
    ball_radius = 2

    -- paddle
    pad_x = 30
    pad_y = 120
    pad_w = 30
    pad_h = 4
    pad_speed = 2

    cls()
end

function _update()
    ball_x = ball_x + ball_dx
    ball_y = ball_y + ball_dy

    move_paddle()

    check_bounce()
end

function check_bounce()
    -- x-achse
    if ball_x+ball_radius > 127 or ball_x-ball_radius < 0 then
        ball_dx = -ball_dx
    end
    -- y-achse
    if ball_y+ball_radius > 127 or ball_y-ball_radius < 0 then
        ball_dy = -ball_dy
    end
end

function move_paddle()

    -- wenn links gedrueckt ist
    if btn(0) then
        pad_x = pad_x - pad_speed
    end

    -- wenn rechts gedrueckt ist
    if btn(1) then
        pad_x = pad_x + pad_speed
    end

    -- ist pad am linken rand?
    if pad_x < 0 then
        pad_x = 0
    end

    -- ist pad am rechten rand?
    if pad_x + pad_w > 127 then
        pad_x = 127 - pad_w
    end

end

function _draw()
    cls()
    circfill(ball_x,ball_y,ball_radius,10)

    rectfill(pad_x,pad_y,pad_x+pad_w,pad_y+pad_h,7)

end
```

```
function _init()
    -- ball
    ball_x = 20
    ball_dx = 2
    ball_y = 64
    ball_dy = -2
    ball_radius = 2
    -- paddle
    pad_x = 30
    pad_y = 120
    pad_w = 30
    pad_h = 4
    pad_speed = 2
    -- clear screen
    cls()
end

function _update()
    ball_x = ball_x + ball_dx
    ball_y = ball_y + ball_dy
    move_paddle()
    check_bounce()

    check_collision()

end

function check_collision()
    -- ist ball unter pad?
    if ball_y-ball_radius > pad_y+pad_h then
        return
    end
    -- ist ball ueber pad?
    if ball_y+ball_radius < pad_y then
        return
    end
    -- ist ball rechts von pad?
    if ball_x-ball_radius > pad_x+pad_w then
        return
    end
    -- ist ball links von pad?
    if ball_x+ball_radius < pad_x then
        return
    end
    -- wir haben eine kollision!
    -- ist die kollision vertikal?
    if ball_y < pad_y or ball_y > pad_y+pad_h then
    ball_dy = -ball_dy
    else
    ball_dx = -ball_dx
    end
end

function check_bounce()
    -- x-achse
    if ball_x+ball_radius > 127 or ball_x-ball_radius < 0 then
        ball_dx = -ball_dx
    end
    -- y-achse
    if ball_y+ball_radius > 127 or ball_y-ball_radius < 0 then
        ball_dy = -ball_dy
    end
end
```

```
function move_paddle()
    -- wenn links gedrueckt ist
    if btn(0) then
        pad_x = pad_x - pad_speed
    end
    -- wenn rechts gedrueckt ist
    if btn(1) then
        pad_x = pad_x + pad_speed
    end
    -- ist pad am linken rand?
    if pad_x < 0 then
        pad_x = 0
    end
    -- ist pad am rechten rand?
    if pad_x + pad_w > 127 then
        pad_x = 127 - pad_w
    end
end

function _draw()
    cls()
    circfill(ball_x,ball_y,ball_radius,10)
    rectfill(pad_x,pad_y,pad_x+pad_w,pad_y+pad_h,7)
end
```

```
function _init()
    -- ball
    ball_x = 20
    ball_dx = 2
    ball_y = 64
    ball_dy = -2
    ball_radius = 2
    ball_color=10
    -- paddle
    pad_x = 30
    pad_y = 120
    pad_w = 30
    pad_h = 4
    pad_speed = 2
    pad_color=7

    -- brick
    brick_x = 30
    brick_y = 30
    brick_w = 13
    brick_h = 4
    brick_color=8
    brick_exists=true

    cls()
end

function _update()
    ball_x = ball_x + ball_dx
    ball_y = ball_y + ball_dy
    move_paddle()
    check_bounce()
    check_collision(pad_x, pad_y, pad_w, pad_h)

    if brick_exists==true then
        collision = check_collision(brick_x, brick_y, brick_w, brick_h)
        if collision==true then
            brick_exists = false
        end
    end

end

function move_paddle()
    -- wenn links gedrueckt ist
    if btn(0) then
        pad_x = pad_x - pad_speed
    end
    -- wenn rechts gedrueckt ist
    if btn(1) then
        pad_x = pad_x + pad_speed
    end
    -- ist pad am linken rand?
    if pad_x < 0 then
        pad_x = 0
    end
    -- ist pad am rechten rand?
    if pad_x + pad_w > 127 then
        pad_x = 127 - pad_w
    end
end
```

```
function check_bounce()
    -- x-achse
    if ball_x+ball_radius > 127 or ball_x-ball_radius < 0 then
        ball_dx = -ball_dx
    end
    -- y-achse
    if ball_y+ball_radius > 127 or ball_y-ball_radius < 0 then
        ball_dy = -ball_dy
    end
end

function check_collision(box_x, box_y, box_w, box_h)
    -- ist ball unter pad?
    if ball_y-ball_radius > box_y+box_h then
        return false
    end
    -- ist ball ueber pad?
    if ball_y+ball_radius < box_y then
        return false
    end
    -- ist ball rechts von pad?
    if ball_x-ball_radius > box_x+box_w then
        return false
    end
    -- ist ball links von pad?
    if ball_x+ball_radius < box_x then
        return false
    end
    -- wir haben eine kollision!
    -- ist die kollision vertikal?
    if ball_y < box_y or ball_y > box_y+box_h then
        ball_dy = -ball_dy
    else
        ball_dx = -ball_dx
    end
    return true
end

function _draw()
    cls()
    circfill(ball_x,ball_y,ball_radius,ball_color)
    rectfill(pad_x,pad_y,pad_x+pad_w,pad_y+pad_h,pad_color)
    if brick_exists==true then
        rectfill(brick_x,brick_y,brick_x+brick_w,brick_y+brick_h,brick_color)
    end
end
```