Check collision

When one asset touches the other, the second one jumps forward.

The `isColliding()` function compares the corners of both shapes. The two shapes overlap if all the conditions are met. In this case the function will return `true`. If any of the comparisons fails the function returns `false`.

The `update(dt)` function:
– `if isColliding()` is true, moves the box 100 pixels horizontally.
– if the right arrow key is pressed, the star moves to the right at the given speed.

Check collision

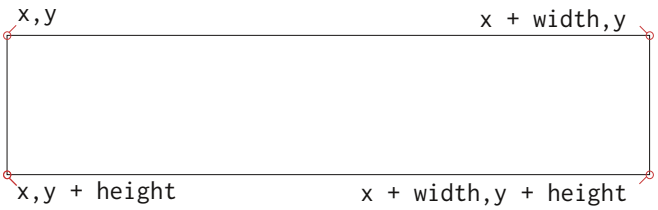
When one asset touches the other, the second one jumps forward.

The `isColliding()` function compares the corners of both shapes. The two shapes overlap if all the conditions are met. In this case the function will return `true`. If any of the comparisons fails the function returns `false`.

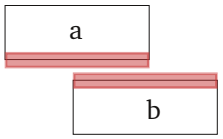
The `update(dt)` function:
– `if isColliding()` is true, moves the box 100 pixels horizontally.
– if the right arrow key is pressed, the star moves to the right at the given speed.

```
star = {x = 10, y = 200, speed = 150,
  char = '*', width = 10, height = 5}
box = {x = 100, y = 200,
  char = '[ ]', width = 10, height = 5}

function isColliding(a, b)
  return
    a.x < b.x + b.width and
    b.x < a.x + a.width and
    a.y < b.y + b.height and
    b.y < a.y + a.height
end

function love.load(arg)
  love.graphics.setFont(love.graphics.newFont(36))
end

function love.update(dt)
  if isColliding(star, box) then
    box.x = box.x + 100
  end

  if love.keyboard.isDown('right', 'd') then
    star.x = star.x + (star.speed * dt)
  end
end
```

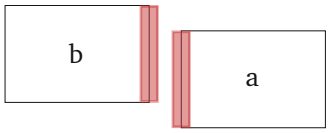
```
function love.draw()
  love.graphics.print(star.char, star.x, star.y)
  love.graphics.print(box.char, box.x, box.y)
end
```



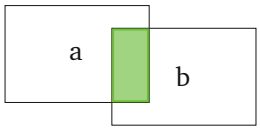
When all equations are met – shown in green – there is a collision. If even a single one is red, the shapes are not colliding.




```
a.x < b.x + b.width
b.x < a.x + a.width
a.y < b.y + b.height
b.y < a.y + a.height
```



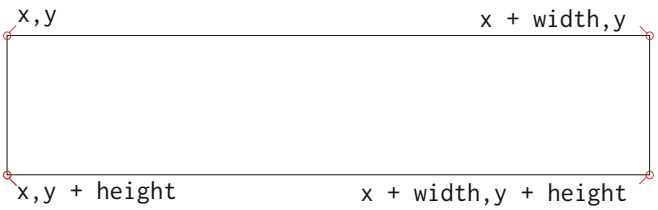

```
a.x < b.x + b.width
b.x < a.x + a.width
a.y < b.y + b.height
b.y < a.y + a.height
```



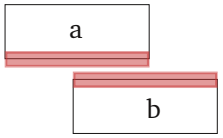

```
a.x < b.x + b.width
b.x < a.x + a.width
a.y < b.y + b.height
b.y < a.y + a.height
```


```
star = {x = 10, y = 200, speed = 150,
  char = '*', width = 10, height = 5}
box = {x = 100, y = 200,
  char = '[ ]', width = 10, height = 5}

function isColliding(a, b)
  return
    a.x < b.x + b.width and
    b.x < a.x + a.width and
    a.y < b.y + b.height and
    b.y < a.y + a.height
end

function love.load(arg)
  love.graphics.setFont(love.graphics.newFont(36))
end

function love.update(dt)
  if isColliding(star, box) then
    box.x = box.x + 100
  end

  if love.keyboard.isDown('right', 'd') then
    star.x = star.x + (star.speed * dt)
  end
end
```

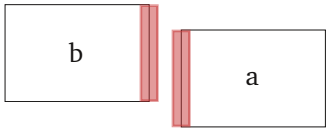
```
function love.draw()
  love.graphics.print(star.char, star.x, star.y)
  love.graphics.print(box.char, box.x, box.y)
end
```



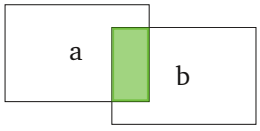
When all equations are met – shown in green – there is a collision. If even a single one is red, the shapes are not colliding.




```
a.x < b.x + b.width
b.x < a.x + a.width
a.y < b.y + b.height
b.y < a.y + a.height
```




```
a.x < b.x + b.width
b.x < a.x + a.width
a.y < b.y + b.height
b.y < a.y + a.height
```




```
a.x < b.x + b.width
b.x < a.x + a.width
a.y < b.y + b.height
b.y < a.y + a.height
```