

# findZones

`findZones(recursive, nameOrFilterFunction)`

## Description

Function to find zones in the specified layer. For example, `this.parent` defines the parent layer of the script module as the layer to be searched in. If `recursive` is set to `true`, sublayers will also be searched. The function returns an array with the [Zone](#) objects of the found zones. Particular zones can be searched by name or through a filter function. If searching by name, `findZones` accepts only the [Zone](#) objects that match the specified name. The filter function uses the [Zone](#) object of each zone as argument. Only those [Zone](#) objects that return `true` for the search criteria defined in the filter function will be accepted by `findZones`. Without a name or filter function the [Zone](#) objects of all zones in the searched layers will be returned.

**Available in:** Controller, Processor.

## Arguments

<b>recursive</b>	If set to <code>false</code> , only the current layer will be searched. If set to <code>true</code> , sublayers will also be searched. The default is <code>false</code> .	boolean
<b>nameOrFilterFunction</b>	The name of the zones searched for or a filter function. Only the <a href="#">Zone</a> objects that match the name or return <code>true</code> for the search criteria of the filter function will be accepted. Set this to <code>nil</code> to deactivate any name filter or search criteria.	string or function, optional

## Return Values

Returns an array with the [Zone](#) objects of the found zones. Returns an empty table if no zones are found.

## Example

```
-- find all zones and print their names
zones = this.program:findZones(true)

if zones[1] then
  for i, zone in ipairs(zones) do
    print(zone.name)
  end
else
  print("Could not find any zones!")
end
```

**See Also:** [findBusses](#), [findChildren](#), [findEffects](#), [findLayers](#), [findMidiModules](#), [findSlots](#), [Zone](#)