

# Contents

- [1 Learn the Names of Items Pressed](#)
- [2 Subscribe to Pressings on All Items](#)
- [3 Learn the Popup Status \(Shown or Hidden\)](#)
- [4 Assign Dynamic Images\(Camera\) to Items](#)

## Learn the Names of Items Pressed

**Syntax:**

```
IR.AddListener(Event, Item-owner, Function, Item-executor);
```

**Input parameters:**

**Event</b>**

the event name or number available for the graphic item;

**Item-owner</b>**

the link to the graphic item which event is used;

**Function</b>**

the named or unnamed function containing commands for execution;

**Item-executor</b>**

the link to the graphic item which will be controlled;

**Output parameters:**

**Subscription success</b> - true / false;**

**Operation description:**

As it is required to learn the name of the event owner, in this case Item-owner and Item-executor will be one and the same item.

**Example:**

Subscribe the function to the event of pressing on the graphic item named Item 1 on Page 1. The function will type the graphic item name

```
IR.AddListener(
```

```
IR.EVENT_ITEM_PRESS,  
IR.GetPage("Page 1").GetItem("Item 1"),  
Item_Press,  
IR.GetPage("Page 1").GetItem("Item 1")  
);  
  
function Item_Press(){  
    IR.Log(this.Name);  
}
```

### Unnamed function:

```
IR.AddListener(IR.EVENT_ITEM_PRESS, IR.GetPage("Page 1").GetItem("Item 1"),  
function(){  
    IR.Log(this.Name);  
}, IR.GetPage("Page 1").GetItem("Item 1"));
```

### Subscription by identifiers:

```
IR.AddListener(IR.EVENT_ITEM_PRESS, IR.GetPage(2).GetItem(0), Item_Press,  
IR.GetPage(2).GetItem(0));  
  
function Item_Press(){  
    IR.Log(this.Name);  
}
```

 Download the example 0,8 Mb

## Subscribe to Pressings on All Items

 Download the example 0,8 Mb

## Learn the Popup Status (Shown or Hidden)

```

// Announce the variable for storing the popup status
var Popup_1_is_Showing = false; // false - hidden, true - shown
// Receive the link to the popup
var Popup = IR.GetItem("Popup 1");

// Subscribe the unnamed function to the event of popup showing. Change the
// value of the status in the function to true
IR.AddListener (IR.EVENT_ITEM_SHOW, IR.GetPage("Popup 1"), function() {

    Popup_1_is_Showing = true;
});

// Subscribe the unnamed function to the event of popup hiding. Change the
// value of the status in the function to false
IR.AddListener (IR.EVENT_ITEM_HIDE, IR.GetPage("Popup 1"), function() {

    Popup_1_is_Showing = false;
});

// Announce the function to check the popup status
function Popup_1_Action_of_Status () {

    if (Popup_1_is_Showing) {

        // Actions if it is shown
    } else {

        // Actions if it is hidden
    }
};

```

## Assign Dynamic Images(Camera) to Items

### Variant 1:

There is a preset dynamic image in the tab Project Gallery / Dynamic Images. The image has the name "Camera".

```
// Assigning the dynamic image to the Image property by the name
IR.GetItem("Page 1").GetItem("Item 1").GetState(0).Image = "Camera";
```

### Variant 2:

Assigning of a dynamic image by the URL link (at that it does not matter if there a preset dynamic image in the project or not).

```
// Assigning a dynamic image (video stream) to the Image property by th URL
```

```
link
IR.GetItem("Page 1").GetItem("Item 1").GetState(0).Image =
"http://admin:12345@192.168.0.1:80/link";

// You can assign images (not video) to the Image property the same way:
IR.GetItem("Page 1").GetItem("Item 1").GetState(0).Image =
"https://images.google.com/images/nav_logo195.png";
```